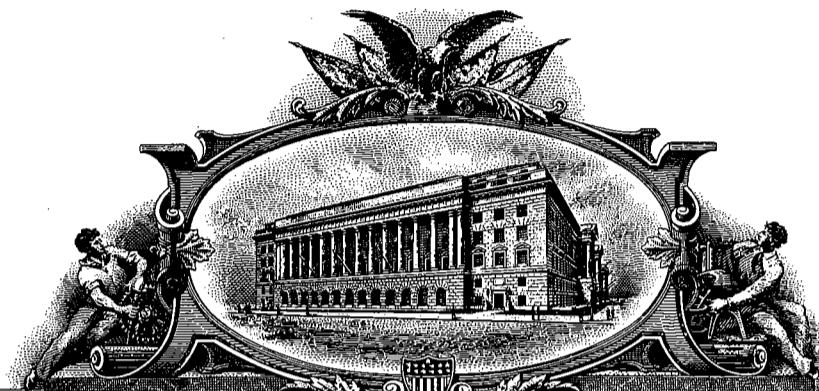


IW 907159



# 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 (1) OF (3) PART(S)

P. SWAIN  
Certifying Officer

The following two claims show that CMFS can accept a set of session if there is a feasible workahead-augmenting sequence. The order  $\pi$  is not important; for simplicity, we assume it is the identity permutation.

**CLAIM 4.** *If  $\Phi$  is workahead-augmenting and feasible, then there is a system state that is safe relative to  $\Phi$ .*

**PROOF.** Let  $W$  be the state in which all buffers are full. Then for all  $j$  we have

$$\begin{aligned} W_j(t) &= \frac{P(t) - C(t) - \bar{Y}}{R} \\ &= \frac{B_j - \bar{Y}}{R} \\ &> \frac{M_j A}{R} \\ &> L(\Phi) \\ &\geq \sum_{i=1}^j L(i), \end{aligned}$$

so  $W$  is safe relative to  $\Phi$  (the final three steps use eqs. (13), (12), and (11)).  $\square$

**CLAIM 5.** *Suppose that there is a feasible workahead-augmenting operation sequence  $\Phi$ , and assume that at time  $t_0$  the state  $W$  is safe relative to  $\Phi$ . Then the CMFS can satisfy the Read Session Axioms (eqs. (1) and (2)) for all  $i$  and  $t \geq t_0$ .*

**PROOF.** We prove this by defining a disk scheduling policy, called the *Static* policy, that satisfies the axioms. The policy is as follows: Repeatedly apply the schedule given by  $\Phi$ , with the exception that, if  $P(t) + A - G(t) > B_i$  at the point of starting a block read for  $S_i$ , then immediately skip to the next session (since reading the block could cause a buffer overflow). It is clear that this policy preserves eq. (1).

Consider a particular session  $S_k$  during one "cycle" of  $\Phi$ , starting at time  $t = 0$  (see Figure 5). Let  $t_D$  denote the time at which  $S_k$ 's operation ends. Equation (2) holds during  $[0, t_D]$  since

$$t_D \leq \sum_{i=1}^k L(i) \leq W_k(0)$$

and  $C(t)$  advances by at most  $R$  bytes/s. The operation for  $S_k$  either reads the full amount or is truncated; in either case,

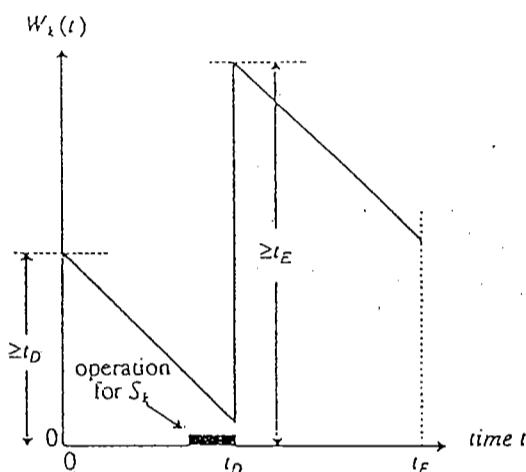


Fig. 5. Diagram for the proof of Claim 5, showing the workahead  $W_k$  of a session  $S_k$  during one cycle of the Static scheduling policy. At the start of the cycle,  $S_k$  has enough workahead to last until time  $t_D$ , when its operation is finished. The amount of data read suffices for at least  $L(\phi)$ , which exceeds the length  $t_E$  of the cycle. Therefore,  $W_k$  is always positive; that is,  $S_k$  never "starves."

since  $\Phi$  is workahead-augmenting. Let  $t_E$  denote the time at which the cycle ends. Then

$$t_E - t_D \leq \sum_{i=k+1}^n L(i). \quad (15)$$

Combining eq. (14), eq. (15), and the clock rate bound  $R$ , we see that  $W_k$  remains positive during  $[t_D, t_E]$  and, moreover,

$$W_k(t_E) \geq \sum_{i=1}^k L(i). \quad (16)$$

Therefore, no starvation occurs during the cycle, and by eq. (16), the state  $W$  at the end of the cycle remains safe relative to  $\Phi$ . Hence, the Static scheduling policy maintains the Read Session Axioms for all sessions.  $\square$

## 5.2 The Minimal Feasible WAS

Claim 5 shows the CMFS can satisfy the data rates of a set of sessions if there is a feasible WAS. We now describe an algorithm to compute the *minimal feasible WAS*  $\bar{\phi}$  (the feasible WAS for which  $L(\phi)$  is least). Clearly, a minimal feasible WAS exists iff a feasible WAS exists:

Suppose that sessions  $S_1 \dots S_n$  are given. Let  $D_i$  be the "duration" of one block of data for  $S_i$ , given by  $A/R_i$ . Let  $\{t_0 < t_1 < \dots\}$  be the set of numbers of the form:  $kD_i$  for  $k \geq 0$  and  $i \geq 0$  (see Figure 6). Let  $I_i$  denote the interval  $(t_i, t_{i+1}]$ . Let  $\phi_i$  denote the operation set  $\langle [R_1 t_i] \dots [R_n t_i] \rangle$ . Note that  $\phi_{i+1}$  differs from  $\phi_i$  by the addition of one block to all sessions whose data periods divide  $t_{i+1}$ ; hence, the sequence of  $\phi_i$  is easy to compute. Note also that  $L(\phi_i) < L(\phi_{i+1})$  for all  $i$ .

**CLAIM 6.** *If  $\phi$  is an operation set such that  $D(\phi) \in I_i$ , then  $L(\phi) \geq L(\phi_i)$ .*

**PROOF.** Any sequence  $\phi$  for which  $D(\phi) > t_i$  must read at least  $[t_i R_j]$  blocks for each session  $j$  and, hence,  $L(\phi) \geq L(\phi_i)$ .  $\square$

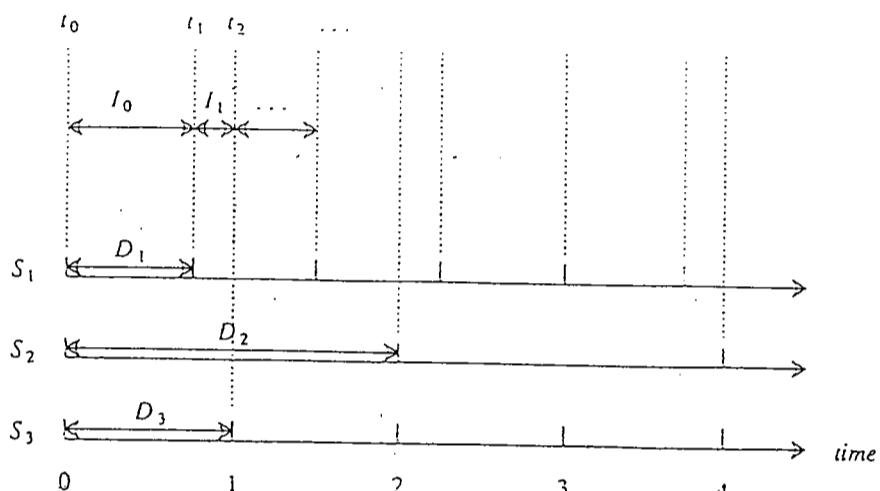


Fig. 6. A block of data for session  $S_i$  has a "duration"  $D_i$  that depends on the data rate of  $S_i$ . The set of all multiples of these periods defines a set of intervals  $I_i$ . Within each interval there is a unique minimal-length operation set  $\phi_i$ . By enumerating the  $\phi_i$ , we can find the minimal feasible WAS.

**CLAIM 7.** *If there is a feasible WAS  $\phi$  such that  $D(\phi) \in I_i$ , then  $\phi_i$  is feasible.*

**PROOF.**  $\phi$  must read at least as many blocks for each session as does  $\phi_i$ . Therefore, since  $\phi$  is feasible, so is  $\phi_i$ .  $\square$

**CLAIM 8.** *The following algorithm computes the minimal WAS:*

- (1) Let  $\phi_0 = \langle 1, \dots, 1 \rangle$ . (this is the minimal operation set for which  $D(\phi) \in I_0$ )
- (2) If  $\phi_i$  is infeasible (i.e., there is no allocation  $\langle B_1, \dots, B_n \rangle$  of buffer space to client FIFOs such that  $M_i A + Y_i \leq B_i$  for all  $i$ ), stop; there is no feasible WAS.
- (3) If  $L(\phi_i) \leq D(\phi_i)$ , stop;  $\phi_i$  is the minimal feasible WAS.
- (4) Compute  $\phi_{i+1}$ , and go to (2).

**PROOF.** Suppose the algorithm stops in step (3), returning a WAS in  $I_j$ . Let  $\phi$  be the minimal WAS, and let  $i$  be such that  $D(\phi) \in I_i$ . It is not possible that  $i < j$ , since then  $\phi_i$  is feasible (Claim 7) and workahead-augmenting, so the algorithm would have terminated at iteration  $i$ . It is also not possible that  $i > j$ , since then  $L(\phi) \geq L(\phi_i) > L(\phi_j)$ , contradicting the minimality of  $L(\phi)$ . Therefore,  $i = j$ , and (from Claim 6) we must have  $L(\phi_i) = L(\phi)$ ; so  $\phi_i$  is the minimal WAS.

Finally, suppose that the algorithm terminates in step (2) for some  $i$ . Suppose that a feasible WAS  $\phi$  exists, with  $D(\phi) \in I_i$ . By the above arguments,  $i \leq j$ . But then  $\phi$  reads at least as many blocks for each session as does  $\phi_i$ , so the buffer allocation feasible for  $\phi$  is feasible for  $\phi_i$ , which is a contradiction.  $\square$

### 5.3 Buffer Space Allotment

Suppose that a fixed amount  $B$  of buffer space is available for CMFS client FIFOs. How should this space be divided among the various clients? CMFS performs best when all sessions can "work ahead" by about the same time (see Section 7). In other words, the buffer space allocated to a session, beyond that needed for the client's cushion  $\bar{Y}$ , should be roughly proportional to the data rate  $R$ .

CMFS therefore uses the following policy: Let  $Y = \sum_{i=1}^n \bar{Y}_i$  and  $R = \sum_{i=1}^n R_i$ . Session  $S_j$  is allocated

$$\bar{Y}_j + \frac{(B - Y)R_j}{R} \quad (17)$$

bytes. This allocation is rounded up, if needed to a multiple of the memory-allocation block size.

## 6. DISK SCHEDULING POLICY

On completion of each disk block I/O, CMFS decides which disk block to read or write next, and issues the appropriate command (seek, read, or write) to the disk device driver. The algorithm for this decision constitutes a *disk scheduling policy*. Such a policy must prevent starvation of current sessions, and must delay the return of the `request_session()` call for a newly accepted session until it is safe to do so. It should also handle non-real-time work load efficiently. Policies for real-time CPU scheduling, such as earliest-deadline-first [7], are not immediately relevant because of seeks. In this section we describe several possible disk scheduling policies. Some of these policies are defined in terms of *slack time*, which we now define.

### 6.1 Slack Time

If, at a particular time, enough data are buffered for all sessions, CMFS is free to do non-real-time operations or workahead for real-time sessions. The amount of this "slack time," denoted  $H$ , is computed as follows: Suppose that the minimal WAS  $\bar{\phi}$  takes worst-case time  $L(1) + \dots + L(n)$ , where  $L(i) = U_{F_i}(M_i)$ . Let  $\pi$  be a permutation of  $1 \dots n$ , and let  $\Phi$  be the operation sequence  $(\pi, \bar{\phi})$ . If  $\Phi$  is performed immediately, the workahead of session  $j$  will not fall below  $H_j = W_j - \sum_{i=1}^j L(\pi(i))$  ( $H_j$  is called the *slack time* of session  $j$ ). CMFS can safely defer starting  $\Phi$  for a period of  $H = \min_{i=1}^n (H_i)$ .

**CLAIM 9.** *Let  $\bar{\pi}$  be the ordering of sessions by increasing value of workahead  $W_i$ . Then, among all permutations  $\pi$ ,  $\bar{\pi}$  gives the maximal value of  $H$ .*

**PROOF.** Consider a permutation  $\pi$  in which  $W_i$  is not increasing; in particular, suppose  $W_i > W_j$ , where  $\pi(i) + 1 = \pi(j)$ . Let  $B$  and  $C$  denote the slack times of the two sessions. If we reverse the order of the two sessions in  $\pi$ , then for the new slack times  $D$  and  $E$  we have  $C < D$  and  $C < E$  (see Figure 7). The slack times of other sessions remain unchanged. Hence, the minimum of the slack times is not decreased by reversing the order.  $\square$

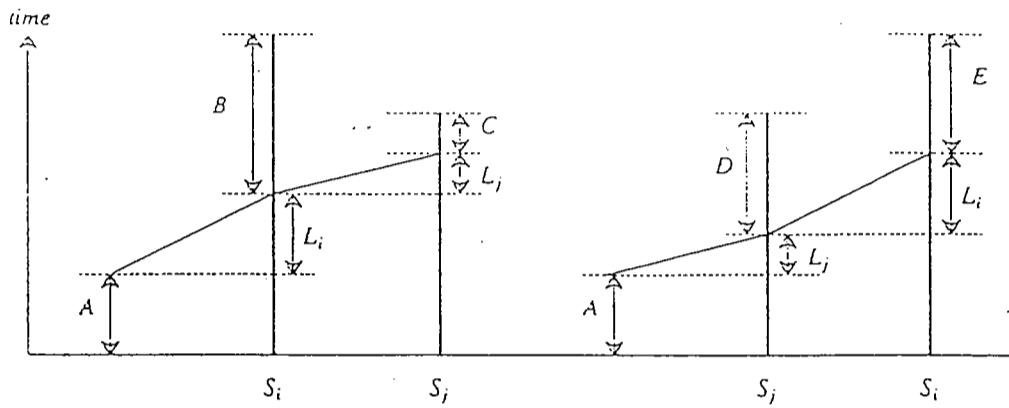


Fig. 7. Slack time is maximized by ordering sessions by increasing workahead. Suppose a sequence has two sessions  $S_i$  and  $S_j$  that are not in this order. The bold lines represent their workahead  $W$ , and their slack times are  $B$  and  $C$  as shown ( $A$  denotes the maximum time needed for operations preceding  $i$ ;  $L_i$  is the time bound for the operation of session  $S_i$ ). By reversing the order of the two sessions, the slack times are  $D$  and  $E$ . Simple algebra shows that  $C < D$  and  $C < E$ . Global slack time is the minimum of the session slack times, so the result follows.

We therefore consider only the increasing-workahead ordering  $\bar{\pi}$  of sessions. Let  $\bar{\Phi}$  denote  $(\bar{\pi}, \bar{\phi})$ , and let  $H_i$  and  $H$  denote the corresponding slack times at a particular moment. It is important to note that  $H < 0$  does not imply that starvation has occurred or will occur.  $H$  is based on the pessimistic assumption that an interfile seek is needed prior to every operation in the WAS. For example, if  $H < 0$  during a multiblock file operation, CMFS is compelled to finish the current operation; starting a new WAS would incur an interfile seek.

## 6.2 Real-Time Scheduling Policies

We now describe several possible disk scheduling policies. These policies all avoid starvation; their relative performance is discussed in Section 7.

- (1) The *Static/Minimal policy* (a special case of the Static policy described in Section 5) simply repeats the minimal WAS.
- (2) The *Greedy policy* does the longest possible read for each session. At each iteration, it computes the slack time  $H$ , finds the session  $S_i$  with smallest workahead, and reads blocks for  $S_i$  for a period of  $H \div L(i)$ ; in other words, it devotes the entire slack time to reading ahead on  $S_i$ .
- (3) The *Cyclical Plan policy* differs from the Greedy policy in that it tries to distribute current slack time among the sessions in a way that maximizes future slack time. It augments the minimal WAS  $\bar{\Phi}$  with  $H$  seconds of additional reads (the reads are done, for each session  $S_i$ , immediately after the read for  $S_i$  in  $\bar{\Phi}$ ). The policy distributes workahead by identifying the “bottleneck session” (that for which  $H_i$  is smallest) and schedules an extra block for it, updating  $H_i$  and  $H$ ; this is repeated until  $H$  is

for the least-workahead session; when this read completes, the procedure is repeated.

In both the Greedy and Cyclical Plan policies, the least-workahead session is serviced immediately. Therefore, the value of  $H$  used by these policies can be computed as the minimum of the slack times of all sessions *except* the least-workahead session, yielding *Aggressive* versions of each policy. All policies skip to the next session when a buffer size limit is reached. If at some point all buffers are full, no operation is done; when a client subsequently removes sufficient data from a FIFO, the policy is restarted.

The Greedy and Static/Minimal policies have low CPU overhead: In our prototype, on a 15-MIPS workstation and with three sessions, they use about 200  $\mu$ s per scheduling decision. Because the Cyclical Plan policy builds its schedule one block at a time, it uses CPU time proportional to buffer space on each transition between sessions. This limits its utility.

### 6.3 Non-Real-Time Operations

A non-real-time operation  $N$  with worst-case latency  $L$  can safely be started if  $L \leq H$ . However, the policy of servicing non-real-time operations whenever it is safe to do so may tend to keep  $H$  low. This forces the scheduler to do short real-time operations (close to the minimal WAS), causing the system to run inefficiently. It may be preferable to do non-real-time operations only when  $H$  exceeds some nonzero threshold.

To avoid the seek overhead of rapidly alternating between real-time and non-real-time operations, CMFS uses the following *slack time hysteresis* policy for non-real-time work load: An interactive operation can be started whenever  $H \geq H_{I2}$ . Initially, interactive operations can be started if  $H \geq H_{I1}$ ; however, if  $H$  falls below  $H_{I1}$  no further interactive operations are started until  $H$  exceeds  $H_{I2}$ . Similarly, background operations are done within a hysteresis interval  $[H_{B1}, H_{B2}]$ . No background operation is started if an interactive operation is eligible to start. In Section 7 we examine the effects of hysteresis, and of the hysteresis parameters, on system performance.

### 6.4 Session Start-Up

A newly accepted session is said to *start* when its `request_session()` call returns. This must occur only when the system state is safe with respect to the new WAS. A special mechanism is needed for handling this "start-up" phase.

Suppose that sessions  $S_1 \dots S_n$  are currently active and that session  $S_{n+1}$  has been accepted but not yet started. Let  $\phi_n$  and  $\phi_{n+1}$  denote the feasible WASs for the sets  $S_1 \dots S_n$  and  $S_1 \dots S_{n+1}$ , respectively.  $S_{n+1}$  is started as follows: CMFS adjusts FIFO buffer sizes according to the procedure described in Section 5.3. It can shrink a buffer by discarding data from the end of the FIFO if needed (it must later reread the data from disk). The scheduler then goes into "start-up mode," during which its policies are changed as follows:

- (1) Non-real-time operations are queued for later execution.

- (2) For scheduling purposes, slack time  $H$  is computed relative to  $\phi_n$ . However, in the Cyclical Plan policy the allocation of slack time for workahead is done relative to  $\phi_{n+1}$ , using a session ordering in which the new session appears first (however, no I/O for  $S_{n+1}$  is done during this phase).
- (3) When the system state is safe with respect to  $\phi_{n+1}$ , a read of  $\phi_{n+1}(n + 1)$  blocks for  $S_{n+1}$  is started. When this read is completed, the system state is "safe" for all  $n + 1$  sessions. The request\_session( ) call for  $S_{n+1}$  is allowed to return,  $\phi_{n+1}$  becomes the system's WAS, and the system leaves start-up mode.

Step (3) can be omitted for write sessions because the equivalent read session starts with a full buffer (Section 2.3).

## 7. PERFORMANCE

In this section we study the effects of disk scheduling policies and hardware parameters on CMFS performance. Our study uses simulation. We chose not to use the "real I/O" version of CMFS because of the scheduling vagaries of UNIX, the poor performance of its SCSI disk I/O, and the restriction to our available disk.

We wrote the CMFS prototype so that disk I/O operations can optionally be simulated rather than performed. The simulator keeps track of the disk head radial and rotational position, and models latencies realistically. Other actions (e.g., CPU execution) are modeled as instantaneous. Unless otherwise stated, the simulations use the Cyclical Plan policy, and assume a disk with 11.8-Mbps transfer rate and 39-ms worst-case seek time. Block size is 512 bytes.

### 7.1 Number of Concurrent Sessions

Figure 8 shows the maximum number of concurrent sessions accepted by CMFS as a function of total buffer space. This is shown for two different session data rates: 64 Kbps and 1.4 Mbps. In each graph, curves are given for three different disk types: 39-ms maximum seek time and 11.8-Mbps transfer rate (CDC Wren V), 35-ms maximum seek time and 8.6-Mbps transfer rate (CDC Wren III), and 180-ms maximum seek time and 5.2-Mbps transfer rate (Sony 5.25" optical disk).

The disk transfer rate imposes an upper bound on the number of concurrent sessions that can be accepted. Unbounded buffer space is needed as this limit is approached. To reach 90 percent of the limit with a Wren V disk requires 4 MB for 1.4-Mbps sessions and 85 MB for 64-Kbps sessions. The efficiency depends on the length of operations in the minimal WAS; 64-Kbps sessions require a proportionally longer WAS and, therefore, more buffer space. When the number of accepted sessions is fixed, a disk with higher seek time needs a longer minimal WAS and, therefore, more buffer space.

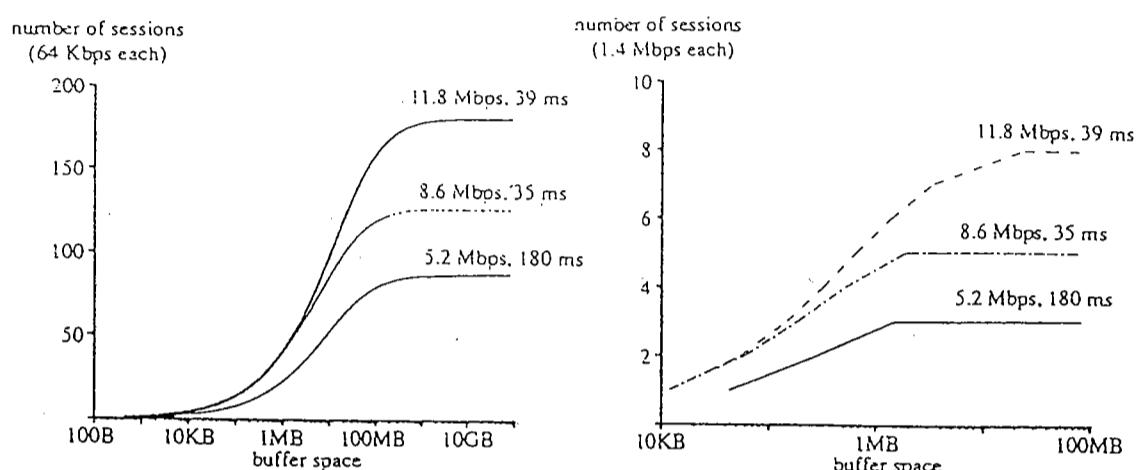


Fig. 8. The number of sessions that can be accepted by CMFS depends on the available buffer space. The disk transfer rate imposes an upper limit on the number of sessions; to reach 90 percent of the limit with the 11.8-Mbps disk requires 4 MB of buffer space for 1.4-Mbps sessions and 85 MB for 64-Kbps sessions.

## 7.2 Performance of Disk Scheduling Policies

Since non-real-time operations can be done only if there is enough slack time, an important criterion for disk scheduling policies is how quickly they increase slack time. To study this, we simulated CMFS with three concurrent 1.4-Mbps sessions, no non-real-time traffic, and 8-MB system buffer size. From the results (Figure 9) we see that the Cyclical Plan policy performs slightly better than the Greedy policy when slack is low, but Greedy quickly catches up. The Static/Minimal policy, because it cannot do long operations, performs much worse at higher slack levels. With appropriate hysteresis values, CMFS maintains moderate slack levels during steady-state operation; thus, the dynamics policies are preferable.

## 7.3 Response Time of Interactive Traffic

To study the effect of real-time traffic on interactive traffic, we simulated a fixed number of sessions together with interactive requests that read randomly positioned blocks from disk. The interactive request arrival is Poisson with mean arrival rate  $\lambda$ . We define the *response time* of an interactive request as the time from its arrival to the start of the disk operation; the delay of the operation itself, including the seek, is not included.

The effect of hysteresis parameters is most noticeable under heavy load (otherwise, slack remains high, and hysteresis is not exercised). Figure 10 plots mean interactive response time against  $H_{t2}$ , for different values of  $H_{t1}$ , under a heavy load. We observe the following:

- For fixed  $H_{t1}$  and increasing  $H_{t2}$ , interactive response time drops steeply and then rises gradually. When  $H_{t1}$  nearly equals  $H_{t2}$ , interactive response is poor because the system switches rapidly between real-time and

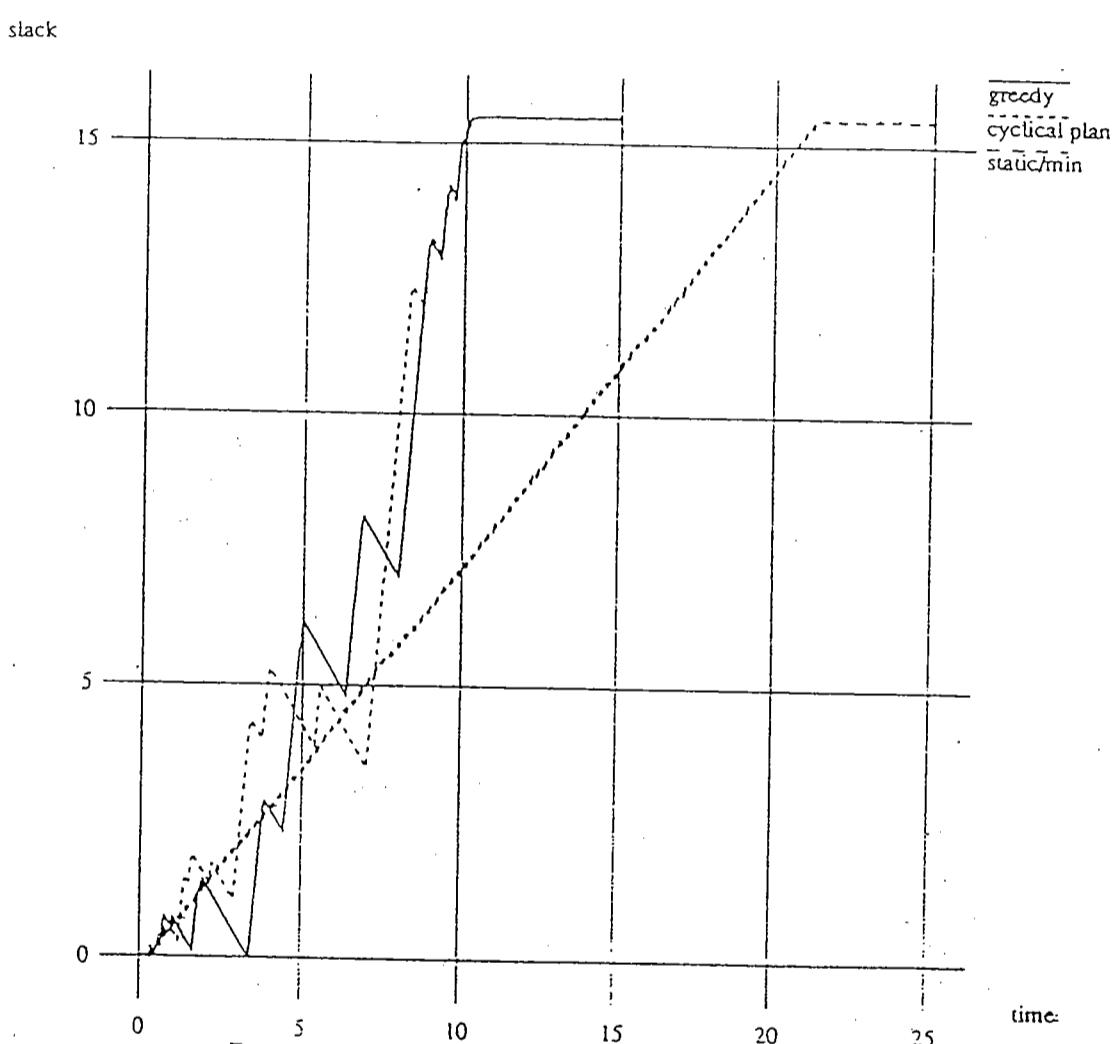


Fig. 9. Disk scheduling policies build up slack at different rates: The Aggressive Cyclical Plan (solid line), Aggressive Greedy (dotted), and Static/Minimal (dashed) policies are shown here (the nonaggressive versions, not shown, performed slightly worse).

non-real-time operations, causing high seek overhead. As  $H_{t2}$  increases, this oscillation becomes less frequent, and response improves. Since interactive requests are queued while slack builds up from  $H_{t1}$  to  $H_{t2}$ , response degrades if  $H_{t2}$  is increased past a certain point. For all  $H_{t1}$ , response is best when  $H_{t2} - H_{t1}$  is about 0.5 s.

When  $H_{t1}$  is very small (e.g., 0.1 s in Figure 10), slack builds up slowly from  $H_{t1}$  to  $H_{t2}$ , so response is poor. A similar effect is observed if  $H_{t2}$  exceeds about  $0.95H_{\max}$  ( $H_{\max}$  denotes the upper bound on  $H$  imposed by buffer space), since it is difficult to fill all buffers simultaneously.

Based on these observations, reasonable "rule of thumb" values are  $H_{t1} = H_{\max}/3$  and  $H_{t2} - H_{t1} = \min(H_{\max}/3, 0.5)$ .

Figure 11 plots mean interactive response time as a function of arrival rate, for different values of system buffer size. If interactive arrival rate is

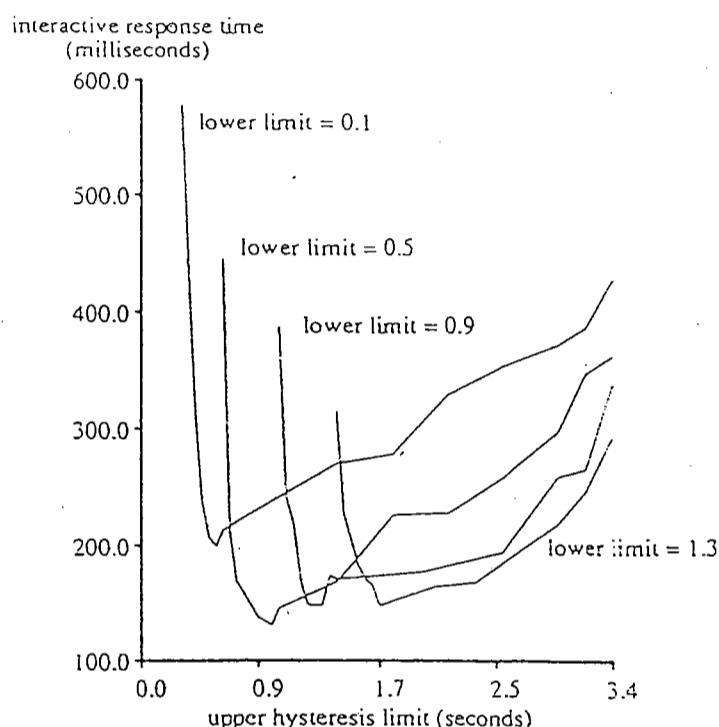


Fig. 10. Effect of the hysteresis limits  $H_{I1}$  and  $H_{I2}$  on the mean response time for non-real-time requests of the interactive class. This experiment was conducted with 2 MB total buffer ( $H_{\max}$  is 3.4 s), 20 interactive arrivals/s, and three 1.4-Mbps sessions.

low, system slack stays near  $H_{\max}$  (Figure 12), and most interactive requests are serviced without waiting for real-time traffic. At high arrival rates (in Figure 11, about 5 per second for the 500-KB case and 10 per second for 1 MB), interactive response degrades because slack sometimes reaches the lower hysteresis limit  $H_{I1}$ , and interactive requests then are blocked until slack reaches  $H_{I2}$ .

#### 7.4 Throughput of Background Traffic

To estimate the effect of real-time traffic on background traffic throughput, we simulated three 1.4-Mbps sessions and a single background task that sequentially reads a long, contiguously allocated file. We define the *background throughput fraction T* as the fraction of residual disk bandwidth (i.e., disk bandwidth not taken up by real-time sessions) used by the background task. For the same reasons as discussed in Section 7.3, T is low if  $H_{B1}$  is very small,  $H_{B2}$  is close to  $H_{\max}$ , or  $H_{B2} - H_{B1}$  is small. In this case (since throughput, rather than response time, is the goal) there is no penalty if  $H_{B2} - H_{B1}$  is large. We found that T was maximized for (roughly)  $H_{B1} = H_{\max}/4$  and  $H_{B2} = .9H_{\max}$ . Figure 13 plots T (with these hysteresis limits) against buffer space.

#### 7.5 Session Start-up Time

Write sessions can usually be started immediately; see Section 2.3. To study start-up time for read sessions, we ran a simulation in which requests for six

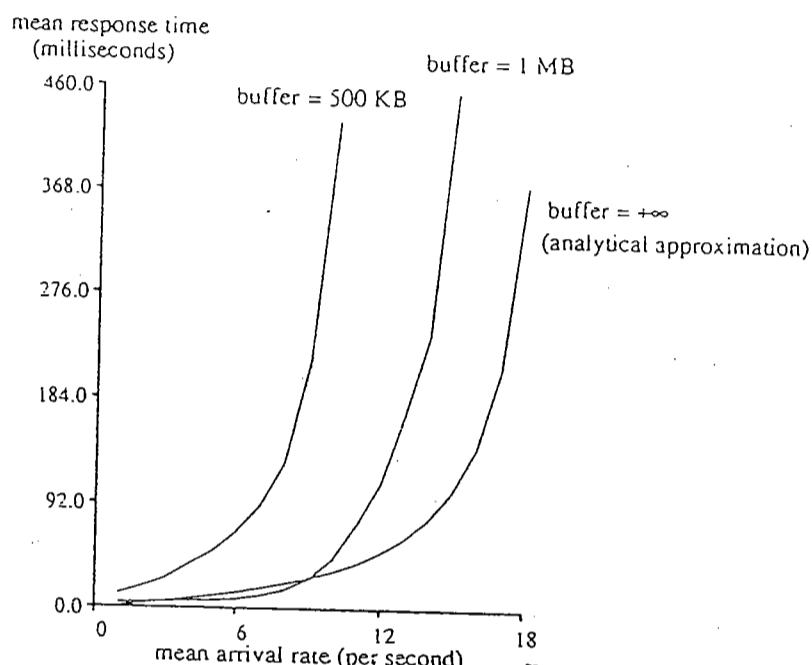


Fig. 11. Mean interactive response time as a function of arrival rate, for different values of total buffer space. In this experiment there were four concurrent 1.4-Mbps sessions. The hysteresis limits (0.04, 0.08) and (0.35, 0.65) were used for the 500-KB and 1-MB cases, respectively. The "infinite buffer" curve was obtained analytically, modeling the file system as an M/G/1 queue.

sessions arrive at time zero. Figure 14 shows the start times of the sessions. The differences between successive start times increase; this is because the workheads of existing sessions have to be increased to accommodate the new minimal WAS, which becomes longer as more sessions are added. Start-up times are on the order of 1 s, which is similar to the start-up time of a consumer VCR. However, it is too large for applications that require instantaneous response, such as interactive musical performance using sounds stored on disk. This problem can be solved by storing an initial segment of each sound file in memory.

## 8. RELATED WORK

Structural issues for multimedia files (sharing, parallel composition, annotations, etc.) have been addressed in the Xerox Etherphone system [14], the Sun Multimedia File System [13], and the Northwestern Network Sound System [12]. These projects do not concentrate on performance or scheduling issues, and the systems cannot make performance guarantees.

Other projects have addressed performance, but without hard guarantees. Abbott gives a qualitative discussion of disk scheduling for playback of multiple audio tracks [1]. He compares a "balanced" policy in which read-ahead is divided among sessions, to a shortest-seek-first policy. His analysis does not, however, provide an acceptance test or performance guarantees.

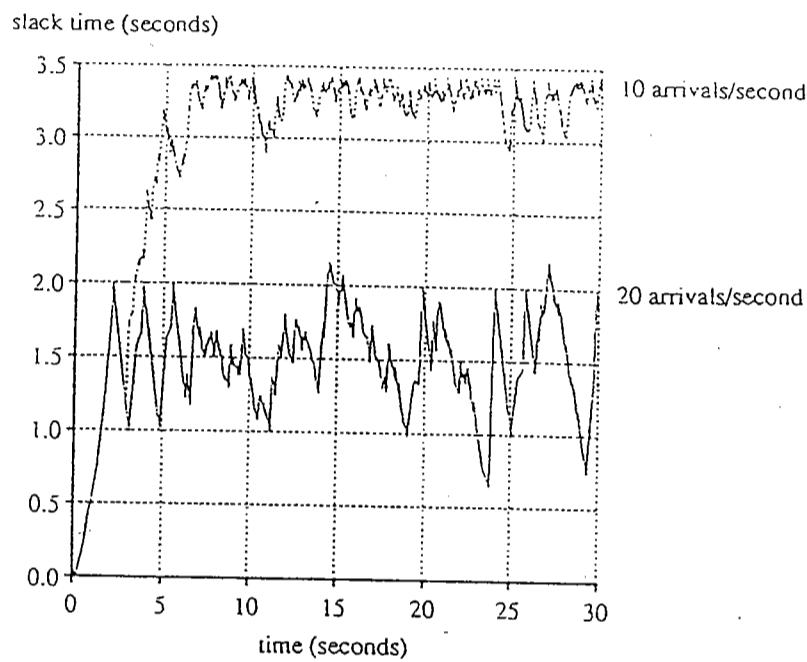


Fig. 12. Variation of slack time  $H$  in the presence of interactive non-real-time traffic. Three 1.4-Mbps sessions start at time zero, the system has 2 MB of buffer space ( $H_{\max}$  is 3.4/s), and hysteresis limits are  $H_{I1} = 1.0$  and  $H_{I2} = 2.0$ . If the interactive arrival rate is low (10 per second in this case),  $H$  stays near  $H_{\max}$ . For high arrival rates (20 per second),  $H$  oscillates between the hysteresis limits.

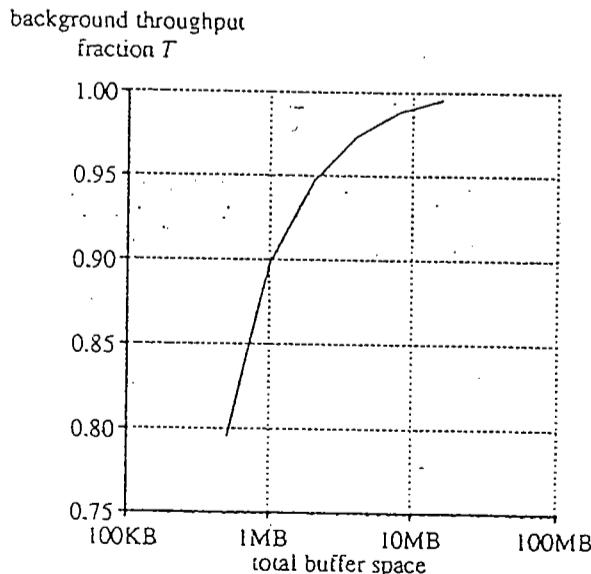


Fig. 13. Background throughput as a function of total buffer space, with a real-time work load of three 1.4-Mbps sessions. As buffer space increases, so does  $H_{B2} - H_{B1}$ ; this allows longer periods of background I/O and, hence, less seek overhead.

Park and English [9] describe a system supporting single-channel audio playback. Non-real-time traffic may concurrently access the disk, causing available disk bandwidth to change. As an alternative to disk bandwidth reservation for the audio channel, they propose changing the data rate of the

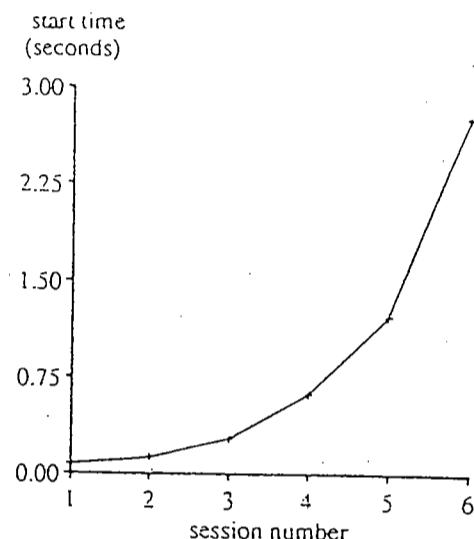


Fig. 14. When six 1.4-Mpbs session requests arrive simultaneously at time zero, their actual start times are staggered as shown.

channel dynamically, to accommodate non-real-time work load. The high data rate is chosen if the workahead on the stream is above a fixed threshold. This strategy does not guarantee a minimum data rate.

Yu et al. [15] discuss the layout of interleaved data streams with different data rates on a compact disk for guaranteed-performance playback. Their assumptions (single session, fixed rates, small buffers, no non-real-time traffic) are more restrictive than ours.

Gemmell and Christodoulakis [5] describe a file system supporting multiple audio-channel playback with concurrent non-real-time traffic. Like CMFS, this work provides a basis for hard performance guarantees. However, it differs from CMFS in several respects. The channels must have the same (constant) data rate and must start at the same time. The scheduling policy is static: The system repeatedly applies a single feasible WAS for the audio channels and reserves "free" time during each operation sequence to service non-real-time traffic. For non-real-time traffic, this static policy may perform worse than CMFS because (1) CMFS can "interrupt" a WAS, allowing non-real-time traffic to start immediately; and (2) CMFS can use accumulated system slack to handle long bursts of non-real-time traffic.

Rangan and Vin [11] describe a system that combines disk input and display-device output for multiple data streams. They give expressions for admission control under the assumption that streams have equal data rates. Their disk scheduling policy is similar to the Static/Minimal policy.

## 9. CONCLUSION

The Continuous Media File System (CMFS) provides guaranteed-performance read and write "sessions." Several such sessions can coexist with non-real-time work load on a single disk. The central ideas of CMFS include the following:

- Semantics.* The CMFS session interface supports a range of client requirements, including variable-rate data, starting and stopping, synchronization

of multiple streams, and client workahead. The semantics are defined rigorously (Section 2.1), but they include a “cushion” factor  $\bar{Y}$  that provides flexibility in client CPU scheduling.

- Layout.* CMFS requires that bounds functions  $U$  and  $V$  can be obtained (Section 4), but does not mandate a particular disk layout.
- Session acceptance.* To decide if a session request can be accepted, CMFS checks if a *feasible WAS* (Section 5.2) exists.
- Disk scheduling.* We found that dynamic policies (Greedy and Cyclical Plan) performed better than the Static/Minimal policy.
- Concurrent non-real-time access.* CMFS handles non-real-time as well as real-time file access; disk space can be dynamically used for either purpose. CMFS uses the *slack time hysteresis* policy for scheduling non-real-time access. With appropriate parameters, this policy allows long non-real-time operations to complete without interruption.

### 9.1 Refinements and Future Work

The following observations suggest possible improvements to CMFS: First, for a session  $S_i$ , the graph of workahead  $W_i$  as a function of time is roughly a “sawtooth” function. If we consider two sessions  $S_1$  and  $S_2$  that have opposite phases in the scheduling cycle, then  $\max(W_1 + W_2)$  is generally less than  $\max(W_1) + \max(W_2)$ .  $S_1$  and  $S_2$  can therefore share buffer space, possibly improving non-real-time performance or increasing the number of sessions that can be accepted. Second, the scheduling policy could take disk head position into account in various ways. For example, it could yield a session ordering that is more efficient than smallest-workahead-first (Section 6.1). Similarly, the use of a policy such as SCAN [4] for ordering non-real-time operations could improve their performance.

Although we have presented the CMFS algorithms in the context of a single-spindle disk drive, they are equally applicable to a disk array in which files are “striped” across multiple disks [10]. A client-level session could be composed of sessions on multiple disks, with each disk reserved and scheduled as described here. This could be used to provide sessions with data rates higher than those of the underlying disk drives. It could improve load-balancing and availability even for sessions with data rates lower than individual disks.

### APPENDIX

**PROOF OF CLAIM 2.** We show a client behavior that is consistent with the Read Session Axioms and that satisfies the above definition of reading a bounded-rate file in real time. Let the client behave as follows: At time  $i/R$  (for each  $i \geq 0$ ) the client removes the next byte  $n$  from the FIFO if  $T(n) < (i+1)/R$ . It then waits until time  $(i+1)/R$ . Hence, in each “time slot” of length  $1/R$  the client either reads a byte or skips to the next time slot. The clock  $C(t)$  advances during a read slot and pauses during a skip slot. The client “works ahead” by at most one byte, so (since the workahead parameter  $\bar{Y}$  of the session is at least one) the Read Session Axioms are obeyed.

To verify the claim, we must first show that each byte  $n$  is read no later than  $T(n) + E/R$ . Suppose otherwise. Then some byte  $n$  is read at time  $j/R$  with

$$\frac{j}{R} > T(n) + \frac{E}{R}. \quad (18)$$

If no skips occurred in slots  $0 \dots j$ , then let  $i = 0$ ; otherwise, let  $i$  be such that  $i - 1$  is the last skip slot before  $j$ . Let  $m$  be the index of the byte read in slot  $i$ . Then

$$T(m) > \frac{i}{R} \quad (19)$$

since otherwise  $m$  would have been read in slot  $i - 1$ . Now combining eqs. (18) and (19) we have

$$T(n) - T(m) < \frac{j - i - E}{R}. \quad (20)$$

Since the client reads in every slot from  $i$  to  $j$ , we have  $j - i = n - m$ . Therefore,

$$n - m > (T(n) - T(m))R + E, \quad (21)$$

which contradicts the assumption that  $F$  is a bounded-rate file.

Finally, we must show that, at any time  $t$ , no more than  $E$  bytes  $i$  such that  $T(i) > t - E/R$  have been read. Let  $i$  be such that  $T(i) > t - E/R$ . Then, given our specification of client behavior, byte  $i$  must have been read at time  $t - E/R$  or later. The client reads at most  $E$  bytes in time  $E/R$ , so the claim holds.  $\square$

#### ACKNOWLEDGMENTS

Discussions with Vassilios Polimenis, George Homsy, and Mark Moran contributed to the CMFS design.

#### REFERENCES

1. ABBOTT, C. Efficient editing of digital sound on disk, *J. Audio Eng. Soc.* 32, 6 (June 1984), 394.
2. ANDERSON D.P. Meta-scheduling for distributed continuous media, Tech. Rep. UCB/CSD 90/599, Electrical Engineering and Computer Science Dept., Univ. of California, Berkeley, Oct. 1990.
3. ANDERSON, D. P., AND HOMSY, G. A continuous media I/O server and its synchronization mechanism, *Computer* 24, 10 (Oct. 1991), 51-57.
4. DENNING, P. J. Effects of scheduling on file memory operations. In *Proceedings of the AFIPS Spring Joint Computer Conference*. AFIPS, Reston, Va., 1967, pp. 9-21.
5. GEMMELL, J., AND CHRISTODOULAKIS, S. Principles of delay-sensitive multimedia data storage and retrieval. *ACM Trans. Inf. Syst.* 10, 1 (Jan. 1992), 51-90.
6. GOVINDAN, R., AND ANDERSON, D. P. Scheduling and IPC mechanisms for continuous media. In *Proceedings of the 13th ACM Symposium on Operating System Principles* (Pacific Grove, Calif., Oct. 14-16, 1991). ACM, New York, pp. 68-80.

7. LIU, C. L., AND LAYLAND, J. W. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM* 20, 1 (Jan. 1973), 47-61.
8. MCKUSICK, M. K., JOY, W. N., LEFFLER, S. J., AND FABRY, R. S. A fast file system for UNIX. *ACM Trans. Comput. Syst.* 2, 3 (Aug. 1984), 181-197.
9. PARK, A., AND ENGLISH, P. A variable rate strategy for retrieving audio data from secondary storage. In *Proceedings of the International Conference on Multimedia Information Systems* (Singapore, Jan. 13-17, 1991), pp. 135-146.
10. PATTERSON, D., GIBSON, G., AND KATZ, R. A case for redundant arrays of inexpensive disks (RAID). In *ACM SIGMOD 88* (Chicago, Ill. June 1988). ACM, New York, pp. 109-116.
11. RANGAN, P. V., AND VIN, H. M. Designing file systems for digital audio and video. In *Proceedings of the 13th ACM Symposium on Operating System Principles*. (Pacific Grove, Calif., Oct. 14-16, 1991). ACM, New York, pp. 81-94.
12. ROTH, J. M., KENDALL, G. S., AND DECKER, S. L. A network sound system for UNIX. In *Proceedings of the 1985 International Computer Music Conference* (Burnaby, B.C., Canada, Aug. 19-22, 1985). pp. 61-67.
13. STEINBERG, D., AND LEARMONT, T. The multimedia file system. In *Proceedings of the 1989 International Computer Music Conference* (Columbus, Ohio, Nov. 2-3, 1989). pp. 307-311.
14. TERRY, D. B., AND SWINEHART, D. C. Managing stored voice in the etherphone system. *ACM Trans. Comput. Syst.* 6, 1 (Feb. 1988), 3-27.
15. YU, C., SUN, W., BITTON, D., BRUNO, R., AND TULLIS, J. Efficient placement of audio data on optical disks for real-time applications. *Commun. ACM* 32, 7 (July 1989), 862-871.

Received August 1991; revised February and June 1992; accepted July 1992

8283-5

23200

(5)



# OPERATING SYSTEMS REVIEW

A Quarterly Publication of the  
Special Interest Group on Operating Systems

Volume 25, Number 5

Special Issue

Proceedings of  
the Thirteenth ACM Symposium

on

Operating Systems Principles

October 13-16, 1991

Asilomar Conference Center

Pacific Grove, CA



XP 000288304

# Designing File Systems for Digital Video and Audio

P. Venkat Rangan and Harrick M. Vin

81-94  
Multimedia Laboratory  
Department of Computer Science and Engineering  
University of California, San Diego  
La Jolla, CA 92093-0114

## Abstract

We address the unique requirements of a multimedia file system such as continuous storage and retrieval of media, maintenance of synchronization between multiple media streams, and efficient manipulation of huge media objects. We present a model that relates disk and device characteristics to the recording rate, and derive storage *granularity* and *scattering* parameters that guarantee continuous access. In order for the file system to support multiple concurrent requests, we develop *admission control algorithms* for determining whether a new request can be accepted without violating the real-time constraints of any of the requests.

We define a *strand* as an immutable sequence of continuously recorded media samples, and then present a *multimedia rope* abstraction which is a collection of individual media strands tied together by synchronization information. We devise operations for efficient manipulation of multi-stranded ropes, and develop an algorithm for maintaining the scattering parameter during editing so as to guarantee continuous playback of edited ropes.

We have implemented a prototype multimedia file system, which serves as a testbed for experimenting with policies and algorithms for multimedia storage. We present our initial experiences with using the file system.

---

This work was supported by the IBM Corporation, the NSF Research Initiation Award No. NCR-9009757, Xerox Corp., UVC Corp., Compaq Corp., Starlight Networks Inc., NCR Corp., and the University of California MICRO program.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1991 ACM 0-89791-447-3/91/0009/0081...\$1.50

## 1 Introduction

### 1.1 Motivation

Future advances in networking and storage [2,3] will make it feasible for distributed systems to support multimedia services such as video and audio mail, news distribution, advertisement, and entertainment [11]. In this paper, we develop mechanisms for multimedia file storage and access, thereby taking us a step closer to multimedia computer systems.

Digital video and audio differ fundamentally from text in three important ways with regard to their storage requirements:

- *Multiple data streams:*

A multimedia object consists of three components: audio, video, and text. Generally, these three components of a multimedia object are separated at the input and they arrive at the file system as three different streams. Similarly during retrieval, these streams are routed to three different output devices. Storing these media together may entail additional processing for combining them during storage, and for separating them during retrieval. The complexity of such processing can be significant if different encodings are used for the three media. On the other hand, if the three media are stored separately, the file system must explicitly maintain temporal relationships among the media so as to ensure synchronization between them during retrieval.

- *Continuous recording and retrieval of data streams:*

Recording and playback of motion video and audio are continuous operations. The file system must organize multimedia data on disk so as to guarantee that their storage and retrieval proceed at their respective real-time rates.

- *Large file size:*

Video and audio data have very large storage space requirements. If the file system is to act as a basis for supporting media services such as document editing, mail, distribution of news and entertainment, etc., it must provide mechanisms for manipulating and sharing stored data. For these mechanisms to be efficient on large sizes of multimedia data, they must minimize copying of data on the disk.

The design and implementation of a file system that addresses the above requirements of multimedia data is the subject matter of this paper.

## 1.2 Relation to Previous Work

Most of the multimedia file systems that are being built or proposed have focused on storage of still images and/or audio. The Diamond system [13], the Muse system of Gibbs et al [4], and the optical disk-based system of Ooi et al [10] are targeted towards storage and exchange of documents containing images. The Sun Multimedia File System [8], which consists of a collection of library functions built on top of Unix, is a storage scheme in which audio samples are stored as Unix files and shared among workstations using Sun's Network File System (NFS). The VOX audio server [1] also supports audio storage. Work by Mackay and Davenport [7] supports video filing, but video is stored in an analog form on consumer electronic devices. There has not been much work on storage systems for digital motion video. The Cambridge Pandora project [5] and Matsushita's Real Time Storage System [9] have begun investigating low level storage mechanisms for digital video.

Terry and Swinehart of the Etherphone project [12] present a powerful voice file system, in which sequences of intervals of voice samples form a *voice rope*. They present mechanisms for copy-free manipulation of voice, and a sophisticated reference count mechanism called *interests* for performing garbage collection of unreferenced voice. Our experience with the Etherphone system has been the initial motivation for the work reported in this paper.

## 1.3 Research contributions of this paper

Digitization of motion video yields a sequence of frames, and that of audio yields a sequence of samples. We call a sequence of continuously recorded video frames or audio samples a *Strand*. Each strand is organized on the disk in terms of blocks. The questions that we first attempt to answer in this paper are (1) how many video frames and/or audio samples are stored in each block (i.e., the *storage granularity*), and (2) how are

the blocks constituting a media strand separated on the disk (i.e., the *scattering parameter*). In order to answer these questions, we relate disk and device characteristics (such as, disk read/write latency, and video capture/display times) to the recording rate, and derive the storage granularity and scattering parameter that result in continuous retrieval.

To enable the file system to support multiple concurrent storage/retrieval requests, we develop an *admission control algorithm* for determining whether a new request can be accepted without violating the real-time constraints of any of the requests.

A file system must not only store video and audio data, but also preserve temporal relationships among them. We present an abstraction called a *Multimedia Rope*, which is a collection of individual media strands tied together by synchronization information. (The term *rope* is derived from the Etherphone project). We devise operations for efficient manipulation of multi-stranded ropes, and develop an algorithm for maintaining the scattering parameter (so as to guarantee continuity of playback) without significant copying of data during insertion and deletion operations.

Using the above results, we have implemented a file system on an environment of SPARCstations and PC-ATs equipped with video compression hardware. We present our initial experiences with using the file system.

The rest of the paper is organized as follows: Section 2 defines the terminology used in the paper. In Section 3, we present methods to determine the granularity and scattering parameter of media storage. In Section 4, we define the structure of a multimedia rope, and discuss operations for its manipulation. Section 5 describes the software architecture of the multimedia file system that we have implemented. Finally, Section 6 summarizes the results and presents directions for future work.

## 2 Preliminary Definitions and Terminology

Frame is the basic unit of video.

Sample is the basic unit of audio.

Strand is an immutable sequence of continuously recorded audio samples or video frames. Immutability of strands is necessary to simplify the process of garbage collection.

Block is the basic unit of disk storage. There are two types of blocks: (1) Homogenous blocks, which contain data belonging to one medium, and (2) Heterogenous blocks, which contain data belonging to multiple media.

Rope is a collection of multiple strands (of same or different medium) tied together by synchronization information.

Table 1 defines the symbols used in this paper. Using these symbols, it can be seen that the duration of playback of a video block (which is the same as its recording duration) is given by  $\frac{n_s}{R_{sr}}$ , the total delay to read a video block from disk is given by  $l_{ds} + \frac{n_s \cdot s_{av}}{R_{sr}}$ , and the time to display a video block, which consists of the time for decompression and digital-to-analog conversion, is given by  $\frac{n_s \cdot s_{av}}{R_{sr}}$ . Note that the time to display a block must not exceed the duration of its playback.

### 3 Determining Granularity and Scattering of Media Strands

A file system must divide video and audio strands into blocks while storing them on a disk. Most existing storage server architectures employ random allocation of blocks on disk. In such storage servers, reserving computational cycles to meet real time requirements is not sufficient to support continuous retrieval of media strands. This is because, separations between blocks of a strand may not be constrained enough to guarantee bounds on access and latency times of successive blocks of the strand. Buffering can nullify the effects of unconstrained variation (i.e., jitter) in separations between blocks. The average seek time per block can be constrained by retrieving blocks in the order in which they are encountered while spanning all the cylinders of a disk (as opposed to the order in which they are to be played back), and then buffering the blocks until their playback. However, the number of blocks that need to be retrieved out of order and buffered can be as much as  $\frac{l_{ds} \cdot n_{cyl}}{l_{desired}}$ , where,  $l_{ds}$  is the maximum possible seek time between blocks on two adjacent cylinders on the disk,  $n_{cyl}$  is the total number of cylinders on the disk, and  $l_{desired}$  is the desired average seek time. Constrained block allocation, on the other hand, can yield the desired average seek time while minimizing the memory buffer requirements on media devices.

Partitioning a disk for multimedia and employing contiguous allocation of blocks within the partition can guarantee continuous access to blocks of a media strand, but it is fraught with inherent problems of fragmentation and can entail enormous copying overheads during insertions and deletions. Even the projected speeds of future fast disk configurations are not sufficient to ensure that unconstrained separation between blocks (i.e., the maximum possible access and latency times) lie within the requirements of high performance video applications. For example, with a block size of 4 Kbytes, future disk arrays with 100 parallel heads and projected

seek and latency times of the order of 10 ms will be able to support 0.32 Gigabits/s transfer rates in the absence of constrained block allocation. This is inadequate for the retrieval of even one HDTV-quality video strand which may require data transfer rates of up to 2.5 Gigabit/s. Hence, constrained block allocation for storing media strands is not an artifact of today's storage performance, but a fundamental problem that is not likely to be obviated by the availability of faster storage devices in the near future. A common file server can, however, integrate the functions of both a conventional text file server and a multimedia file server by employing constrained block allocation for (real-time) media strands, and using the gaps between successive blocks of a media strand to store text files.

There are two questions that need to be answered in constrained allocation of blocks of a media strand: (1) What should the size of the blocks (i.e. the granularity) be? and (2) What should the separation between successive blocks (i.e. the scattering parameter) of a strand be? The guiding factor in determining the block size and separation is the requirement of continuous recording and retrieval. In the remainder of this section, we present a model that attempts to answer these questions and obtain the associated buffering requirements for motion video which is the most demanding medium (with regard to performance); the analysis for audio can be carried out in a similar manner.

We make two simplifying assumptions, both of which are reasonable in our hardware environment: (1) the disk write and read times are approximately equal, and (2) the time to capture a video frame (which consists of digitization and compression) and the time to display it (which consists of decompression and digital-to-analog conversion) are approximately equal. Hence, the continuity requirements of retrieval and storage are similar to each other, and in the analysis that follows, we only consider continuous retrieval.

#### 3.1 Continuity Requirement

For continuous retrieval of media data, it is essential that media information be available at the display device at or before the time of its playback. We refer to this as the '*continuity requirement*'. Whereas the duration of playback of a media block stored on the disk is determined by the rate of recording; the time to access a disk block depends on the level of concurrency between disk access and video display. Disk access and video display can be purely sequential, or can be pipelined. Furthermore, if disks with multiple heads are used (such as RAIDs), multiple disk accesses can take place concurrently. We now analyze the sequential, pipelined, and concurrent architectures for continuity requirements.

Symbol	Explanation	Unit
$R_{ar}$	Audio recording rate	samples/sec
$R_{vr}$	Video recording rate	frames/sec
$R_{dr}$	Rate of data transfer from disk	bits/sec
$R_{vd}$	Rate of video display	bits/sec
$\eta_{vs}$	Granularity of video storage	frames/block
$\eta_{as}$	Granularity of audio storage	samples/block
$s_{vf}$	Size of a video frame	bits/frame
$s_{as}$	Size of audio sample	bits/sample
$l_{ds}$	Scattering parameter	sec

Table 1: Symbols used in this paper

Sequential architectures serialize read and display (similarly, capture and store) operations (see Figure 1). Each block is transferred from disk to a buffer in the video device, and then displayed before initiating the transfer of the next block.

The continuity requirement is met in this case if the sum of the time to read a block from disk and the time to display it does not exceed the duration of its playback. That is,

$$(l_{ds} + \frac{\eta_{vs} * s_{vf}}{R_{dr}}) + \frac{\eta_{vs} * s_{vf}}{R_{vd}} \leq \frac{\eta_{vs}}{R_{vr}} \quad (1)$$

Pipelined architectures perform read and display operations in parallel (see Figure 2).

If there are a minimum of two buffers on the video device, one holding the block being transferred and the other holding the block being displayed, the continuity requirement is met if the time to read a block does not exceed the duration of its playback. That is,

$$l_{ds} + \frac{\eta_{vs} * s_{vf}}{R_{dr}} \leq \frac{\eta_{vs}}{R_{vr}} \quad (2)$$

Concurrent architectures perform multiple disk read operations in parallel. Let  $p$  be the degree of concurrency, i.e., the number of concurrent disk accesses.

If there are  $p$  buffers in the video device to hold the  $p$  blocks being transferred simultaneously (see 3), continuity of playback will be maintained if the time to read a block does not exceed the duration for playback of  $(p - 1)$  blocks. Hence,

$$l_{ds} + \frac{\eta_{vs} * s_{vf}}{R_{dr}} \leq (p - 1) * \frac{\eta_{vs}}{R_{vr}} \quad (3)$$

### 3.2 Synchronous Playback Requirement

In addition to maintaining continuity of retrieval, it is essential that the playback of a strand proceed at exactly the same rate as it was recorded. This is referred to as synchronous playback, and can be accomplished by one of two possible techniques:

- *Forced synchronization:* Using a clocking device, the display process can be forced to wait for appropriate time before displaying each block. This scheme entails communication overhead between clocking and display devices, and can be performed at the frame or block boundaries.
- *Automatic synchronization:* The left hand sides of Equations (1), (2), and (3) represent the effective access time per media block. If this becomes equal to the playback duration of a block, synchronization becomes automatic.

### 3.3 Discussion

#### 3.3.1 Strict and Average Continuity Requirements

Recall that for continuous retrieval of media data, it is essential that media data be available at the display device at or before the time of its playback. Satisfying this condition deterministically for each block is referred to as '*strict continuity requirement*', and is difficult to achieve in the presence of scheduling and seek time variations. By introducing anti-jitter delay at the beginning of each request, we can relax the continuity requirements so as to satisfy it on an average. Anti-jitter delay can be introduced by performing read-ahead of media blocks.

#### 3.3.2 Buffering and Read-Ahead Requirements

When strict continuity requirements are satisfied, assuming buffers to be of the same size as disk blocks, the

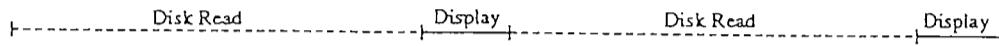


Figure 1: Sequential retrieval

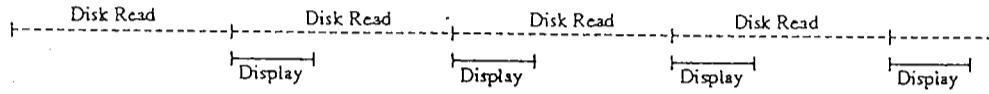


Figure 2: Pipelined retrieval

sequential, pipelined, and concurrent architectures require 1, 2, and  $p$  buffers, respectively. When continuity requirements are satisfied over an average of  $k$  successive blocks of a strand, in order to guarantee that the next group of  $k$  blocks can be retrieved from the disk within the time required to display a previous group of  $k$  blocks of a strand, the sequential and pipelined architectures require a read-ahead of  $k$  blocks, whereas the concurrent architecture requires a read-ahead of  $pk$  blocks ( $k$  for each of the  $p$  heads). In the case of sequential and concurrent transfers, the number of buffers required is same as the amount of read-ahead (i.e.,  $k$  and  $pk$ , respectively), whereas, in the case of pipelined transfer, the number of buffers required is twice that amount,  $2k$ : one set of  $k$  buffers to hold the blocks being displayed, and another set of  $k$  buffers to hold the blocks being transferred from the disk, both of which occur simultaneously.

Functions such as fast-forwarding can be supported by satisfying continuity requirements at the fastest required display rate. Whereas fast-forwarding without skipping frames increases both continuity and buffering requirements, fast-forwarding with skipping increases only the continuity requirement. However, when blocks are displayed slower than the fastest rate (e.g., in slow motion), continuity requirements become over-satisfied, and retrieval of media blocks proceeds faster than their display, leading to accumulation of media blocks in buffers. In order to prevent unbounded accumulation, the disk can switch to some other task after all the buffers allocated to the retrieval of a media strand are filled, and switch back when sufficient buffers become empty. In order to compute the buffering needs in such a situation, note that after the disk switches to some other task, the disk head may have moved to a random location, and hence may have to incur maximum seek (and latency) time,  $l_{seek}^{max}$  before being able to resume the transfer of blocks of the earlier media strand. Thus, in order to guarantee that the display does not run out of media blocks during a switch to another task, the disk must read ahead an additional  $h$  blocks before the switch, given by,

$$h = l_{seek}^{max} \times \frac{R_{vr}}{\eta_{vs}}$$

where,  $\frac{R_{vr}}{\eta_{vs}}$  is the rate at which blocks are played back. Thus, the number of buffers would also be increased by  $h$ .

### 3.3.3 Storing Multiple Media Strands

The analysis presented so far has considered only one medium. There are two approaches for storing multiple media on a disk.

- *Heterogeneous Blocks:* Multiple media being recorded are stored within the same block, which may entail additional processing for combining these media during storage, and for separating them during retrieval. The advantage of this scheme is that it provides implicit inter-media synchronization.

- *Homogeneous Blocks:* Each block contains exactly one medium. This scheme permits the file system to exploit the properties of each medium to independently optimize its storage. However, the file system must maintain explicit temporal relationships among the media so as to ensure synchronization between them during retrieval.

We illustrate the analysis for deriving continuity equations for the pipelined architecture when there is one audio and one video component in the media source. For homogeneous blocks, the number of blocks to be retrieved increases with the number of media. Hence, if the duration of playback of audio block is  $n$  times that of a video block, an audio block is retrieved from disk for every  $n$  video blocks. Hence, the continuity requirement becomes

$$\frac{1}{n} \underbrace{(l_{ds}^a + \frac{\eta_{as} * s_{as}}{R_{dr}})}_{\text{Audio}} + \sum_{i=1}^n l_{ds}^v + \underbrace{\frac{\eta_{vs} * s_{vf}}{R_{dr}}}_{\text{Video}} \leq \frac{\eta_{vs}}{R_{vr}} \quad (4)$$

On the other hand, if the duration of audio blocks is identical to that of video blocks (i.e.,  $n = 1$ ), then the continuity requirement reduces to

$$\underbrace{l_{ds}^a + \frac{\eta_{as} * s_{as}}{R_{dr}}}_{\text{Audio}} + \underbrace{l_{ds}^v + \frac{\eta_{vs} * s_{vf}}{R_{dr}}}_{\text{Video}} \leq \frac{\eta_{vs}}{R_{vr}} \quad (5)$$

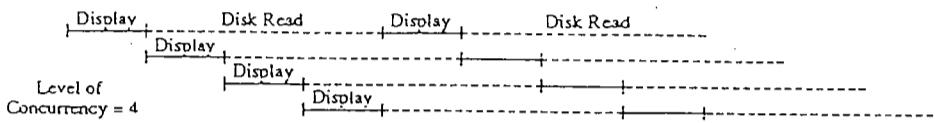


Figure 3: Concurrent retrieval

If the audio and video blocks are scattered on the disk such that  $l_{ds}^a = 0$ , then the continuity requirement reduces to that of the heterogeneous block case:

$$l_{ds}^v + \frac{\eta_{vs} * s_{vf}}{R_{dr}} + \frac{\eta_{as} * s_{as}}{R_{dr}} \leq \frac{\eta_{vs}}{R_{vr}} \quad (6)$$

Video+Audio

### 3.3.4 Determining Granularity and Scattering

Having derived the continuity equations relating granularity of storage ( $\eta_{vs}$ ) with the scattering parameter ( $l_{ds}$ ), we discuss the process of determining each of these parameters for a given target environment.

Media blocks may be transferred from disk to a display device either *directly* to the internal buffers of the display device, or *through memory* (from disk to main memory, and then from main memory to the internal buffers of the display device). Whereas direct transfer is usually preferable, transfer through memory is more suitable for heterogeneous blocks. However, transfer through memory requires double the internal bus bandwidth. Hence, we will only consider the direct transfer approach below.

When direct transfer is used, the sizes of internal buffers available on the display devices can be used to determine the granularity of storage. For instance, if the video display device contains an internal buffer of the size of one video frame, the size of disk block can match this size, yielding  $\eta_{vs} = 1$ . On the other hand, if the internal buffers can store multiple frames (say  $f$ ), then pipelined retrieval can be used by dividing the buffer into two parts, each of size  $f/2$ , and  $\eta_{vs}$  can be chosen anywhere in the range  $1, \dots, f/2$ . If the disk permits  $p$  concurrent accesses, and the size of internal buffers is  $f$  frames, then  $\eta_{vs}$  can be chosen anywhere in the range  $1, \dots, f/p$ .

Having determined the value of granularity  $\eta_{vs}$ , the upper bound of the scattering parameter  $l_{ds}$ , can be obtained by direct substitution in the continuity equations.

## 3.4 Servicing Multiple Requests

In practice, a file server has to process requests from several clients simultaneously. Given a maximum rate of disk data transfer, the file system can only accept a limited number of requests without violating the continuity requirements of any of the requests. In this section, we

formulate this resource allocation problem, and present an *admission control algorithm* for determining whether to accept a new request, given an existing set of requests being serviced.

Consider a scenario in which a file server is servicing  $n$  active media storage/retrieval requests. In order to service multiple requests simultaneously, the file system proceeds in rounds. In each round, it multiplexes among the media block transfers of the  $n$  requests. Let  $k_i$ , for  $i \in [1, n]$ , be the number of consecutive blocks retrieved for  $i$ th request before switching to the next request. Let  $\eta_{vs}^1, \eta_{vs}^2, \dots, \eta_{vs}^n$ , and  $R_{vr}^1, R_{vr}^2, \dots, R_{vr}^n$  be the granularities, and recording rates, respectively, of the strands corresponding to the  $n$  requests.

When the file server switches from one request to another, it may entail an overhead of up to the maximum disk seek time to move from a block in the first strand to a block of the second strand (since there is no guarantee on the relative positions of two strands belonging to two requests). The total time spent servicing  $i$ th request in each round can be divided into two parts:

1.  $\theta_i^1$ : The overhead of switching from the previous request to the  $i$ th request, and then transferring the first block of  $i$ th request.

$$\Rightarrow \theta_i^1 = l_{seek}^{max} + \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}} \quad (7)$$

2.  $\theta_i^2$ : The time to transfer remaining  $(k_i - 1)$  blocks of this request in this round.

$$\Rightarrow \theta_i^2 = \sum_{j=1}^{k_i} (l_{ds}^{ij} + \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}}) \quad (8)$$

Hence, the total time spent servicing  $i$ th request in a round is

$$\theta_i = \theta_i^1 + \theta_i^2 \quad (9)$$

The total time spent servicing one round of all the  $n$  requests is

$$\Theta = \sum_{i=1}^n \theta_i = n * l_{seek}^{max} + \sum_{i=1}^n (\frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}}) + \sum_{i=1}^n \sum_{j=1}^{k_i-1} (l_{ds}^{ij} + \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}}) \quad (10)$$

The continuity requirement for each of the requests can be satisfied if and only if the service time per round does

not exceed the minimum of the playback durations of all the requests. That is,

$$n * l_{ds}^{max} + \sum_{i=1}^n \left( \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}} \right) + \sum_{i=1}^{k_i-1} \sum_{j=1}^{k_i-1} \left( l_{ds}^{ij} + \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}} \right) \leq \min_{i \in [1, n]} (k_i * \frac{\eta_{vs}^i}{R_{vr}}) \quad (11)$$

Thus, the file system can service all the  $n$  requests simultaneously if and only if  $k_1, k_2, \dots, k_n$  can be determined such that Equation (11) is satisfied. Determination of  $k_1, k_2, \dots, k_n$  in this most general formulation is beyond the scope of this paper. We make the following simplifying assumptions and develop an algorithm for admission control:

- The values of all  $k_i$ 's are identical. That is,  $k_1 = k_2 = \dots = k_n = k$ .
- We assume that

$$\sum_{i=1}^n \left( \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}} \right) \approx n * \left( \frac{\eta_{vs}^{avg} * s_{vf}^{avg}}{R_{dr}} \right)$$

and

$$\sum_{i=1}^{k_i-1} \sum_{j=1}^{k_i-1} \left( l_{ds}^{ij} + \frac{\eta_{vs}^i * s_{vf}^i}{R_{dr}} \right) \approx n * (k-1) * \left( l_{ds}^{avg} + \frac{\eta_{vs}^{avg} * s_{vf}^{avg}}{R_{dr}} \right)$$

where, individual values of the granularity,  $\eta_{vs}$ , the size of a video frame,  $s_{vf}$ , and the scattering parameter,  $l_{ds}$  are replaced by their respective averages in the summation.

We define

$$\alpha = l_{ds}^{max} + \frac{\eta_{vs}^{avg} * s_{vf}^{avg}}{R_{dr}} \quad (12)$$

$$\beta = l_{ds}^{avg} + \frac{\eta_{vs}^{avg} * s_{vf}^{avg}}{R_{dr}} \quad (13)$$

$$\gamma = \min_{i \in [1, n]} \frac{\eta_{vs}^i}{R_{vr}^i} \quad (14)$$

where, for a block of average granularity,  $\alpha$  defines the maximum scattering, and  $\beta$  defines the average scattering. Note that since  $l_{ds}^{max} \geq l_{ds}^{avg}$ , it is guaranteed that  $\alpha \geq \beta$ . Under these assumptions, the continuity requirement of Equation (11) reduces to

$$n * \alpha + n * (k-1) * \beta \leq k * \gamma \quad (15)$$

$$\begin{aligned} k &\geq \frac{n(\alpha-\beta)}{(\gamma-n\beta)} & \text{if } \gamma > n\beta \\ \Rightarrow k &\leq \frac{n(\alpha-\beta)}{(n\beta-\gamma)} & \text{if } \gamma < n\beta \\ k &= \infty & \text{if } \gamma = n\beta \end{aligned} \quad (16)$$

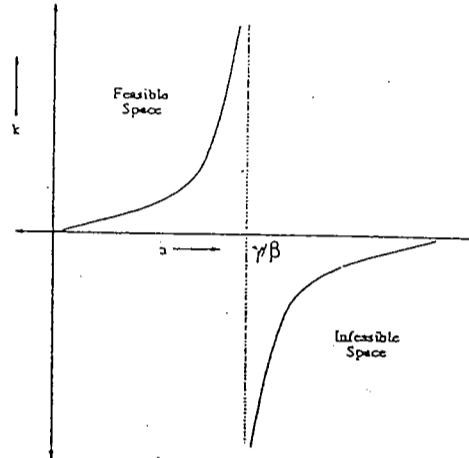


Figure 4: Variation of the number of blocks ( $k$ ) with respect to the number of requests ( $n$ )

Figure 4 shows the variation of  $k$  with respect to  $n$ . Since  $k$  must be non-negative, the value of  $k$  obtained from Equation (16) is meaningful if and only if  $\gamma > n\beta$ . Thus, the maximum number of simultaneous requests that a file system can service is

$$n_{max} = \lceil \frac{\gamma}{\beta} - 1 \rceil \quad (17)$$

It should be noted that, since the right hand side of Equation (15) represents the playback duration of the request with the fastest display rate, transferring  $k$  blocks of all other requests at that rate may lead to accumulation of data in the display subsystems of these other requests. Such an accumulation can be eliminated by regulating the number of data blocks transferred for each request during each service round, so as not to overflow the buffering available in the display subsystem of that request. Furthermore, larger the value of  $k$ , larger is the startup time for a new request. Thus, it is desirable to use the minimum possible value of  $k$ .

While servicing  $n$  requests, if the file server receives  $(n+1)$ th request, it must now decide whether or not to admit the new request or not. If  $n+1 \leq n_{max}$  derived from Equation (17), it can determine the new values of  $\alpha$ ,  $\beta$ , and  $\gamma$ , and compute  $k_{new}$  (from Equation (16)) necessary for satisfying  $(n+1)$  requests.

If  $k_{new} = k_{old}$  (where,  $k_{old}$  is the value of  $k$  when the file system was servicing  $n$  requests), then it can immediately admit the  $(n+1)$ th request. However, if  $k_{new} \neq k_{old}$ , then  $k_{new} > k_{old}$  (see Figure 4), and the file system has to begin transferring  $k_{new}$  blocks of each of the earlier  $n$  requests, and of the new  $(n+1)$ th request. During this round, the number of blocks being transferred is  $k_{new}$ , whereas, the number of blocks available for display are those of the previous round, which is  $k_{old}$ . Since,  $k_{new} > k_{old}$ , the time spent to transfer  $k_{new}$

blocks may exceed the playback duration of  $k_{old}$  blocks of some of the requests, and a discontinuity may result in their playback. In other words, Equation (15) guarantees continuity only in steady state, and not during transitions.

In order to guarantee a smooth and transparent transition, we propose the following modification to Equation (15). Suppose the file system makes a transition from  $k_{old}$  to  $k_{new}$  in steps of 1 before beginning to service the  $(n+1)$ th request. When it performs a transition from  $k_{old}$  to  $(k_{old}+1)$ , the time to transfer  $(k_{old}+1)$  blocks must not exceed the minimum playback duration of  $k_{old}$  blocks. Thus, if we use the time to transfer  $(k+1)$  blocks instead of  $k$  in the left hand side of Equation (15) but use  $k$  in the right hand side, and then solve for  $k$ , a transparent transition from  $k_{old}$  to  $(k_{old}+1)$  is guaranteed. Thus, Equation (15) changes to

$$n\alpha + n * k * \beta \leq k * \gamma \quad (18)$$

Furthermore, since  $\gamma \geq n\beta$ ,

$$n\alpha + nk\beta \leq k\gamma \Rightarrow n\alpha + n(k+1)\beta \leq (k+1)\gamma$$

Hence, a transition from  $k_{old}+1$  to  $k_{old}+2$ ,  $k_{old}+2$  to  $k_{old}+3$ , ...,  $k_{new}-1$  to  $k_{new}$  are also guaranteed. Thus, using Equation (18) to determine  $k$ , and increasing it in steps of 1, yields an admission control algorithm that guarantees both transient and steady state continuity.

### 3.5 Layout of Blocks in a Strand

A media strand consists of a sequence of *Media Blocks* (*MB*), whose size and separation are determined using the techniques described in the previous sections. Each media block contains either video frames, audio samples, or both. A 3-level index structure permits large strand sizes, and random as well as concurrent access to strands.

For each strand, the file system maintains primary indices in a sequence of *Primary Blocks* (*PB*), each of which contains mapping from media block numbers to their raw disk addresses. Secondary indices, which are pointers to Primary Blocks, are maintained in a sequence of *Secondary Blocks* (*SB*). Pointers to all Secondary Blocks of a strand are stored in the *Header Block* (*HB*). A pictorial representation and the exact data structures of these blocks are shown in Figures 5 and 6, respectively.

## 4 From Media Strands to Multimedia Ropes

Multimedia data includes information in various forms: audio, video, textual, olfactory, thermal, tactile, etc.

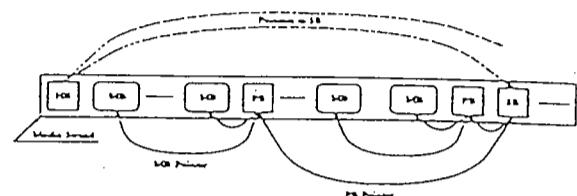


Figure 5: Organization of blocks constituting a media strand on disk

<b>Primary Block [</b>	
sector,	- position of MB on disk
sectorCount	- length of the MB in sectors
]	
<b>Secondary Block [</b>	
startBlock,	- start Block number
BlockCount,	- number of Blocks in PB
sector,	- position of PB on disk
sectorCount	- length of PB in sectors
]	
<b>Header Block [</b>	
frameRate,	- Rate of recording
secondaryCount,	- Number of secondary blocks
frameCount,	- Total number of frames
secondaryArray	- Array of pointers to SB
]	

Figure 6: Structure of 3-level indices of a media strand

All the media strands constituting a piece of information are tied together by inter-media synchronization to form a multimedia rope (see Figure 7). A rope contains the name of its creator, its length, access rights, and for each of its component media strands, the strand's unique ID (a NULL ID indicates the absence of that media in the rope), rate of recording, granularity of storage, and block-level correspondence (see Figure 8). The block-level correspondence information is used to synchronize the start of playback of all the media at strand interval boundaries. Within each strand interval, playing back at the strand's recording rate automatically guarantees simultaneity of playback between the media. Thus, the block-level correspondence and the recording rate information together maintain inter-media synchronization in multimedia ropes.

Maintenance of media synchronization information is complicated by silence detection and elimination of audio data. In silence elimination, if the average energy level over a block falls below a threshold, no audio data

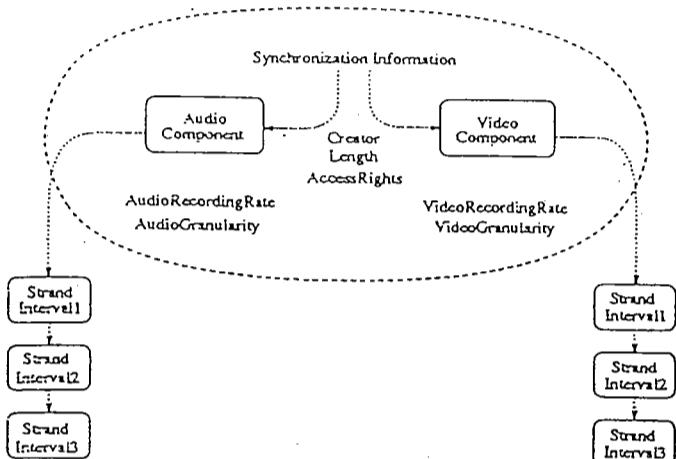


Figure 7: A multimedia rope containing video and audio

is stored for that duration. However, after undergoing silence elimination, audio strands no longer have lengths proportional to their duration. Hence, explicit delay holders have to be placed in audio strands to represent silences. We use NULL pointers in the primary blocks of a strand to indicate silence for the duration of a block.

In order to guarantee continuous retrieval, editing operations on ropes such as insert and delete may require substantial copying of their component strands. The strands can be very large in size and hence copying can consume significant amount of time and space. In order to minimize the amount of copying involved in editing, the multimedia file system regards strands as immutable objects, and all editing operations on ropes manipulate pointers to strands. Thus, an edited rope contains a list of pointers to intervals of strands. Many different ropes may share intervals of the same media strand. A media strand, no part of which is referred to by any rope, can be deleted to reclaim its storage space. A garbage collection algorithm such as the one presented by Terry and Swinehart in the Etherphone system [12], which uses a reference count mechanism called *interests*, can be used for this purpose. To simplify the process of garbage collection of media strands and ropes, synchronization information (which is typically very small in size) is copied from a rope to another when they share strands.

#### 4.1 Operations on Multimedia Ropes

The file system provides facilities for creating, editing, and retrieving multimedia ropes. The exact interfaces are as follows:

RECORD [media] — [requestID, mmRequestID]

Assuming the user has the required access permissions, the file system begins recording a new mul-

timedia rope (represented by `mmRopeID`) consisting of new media (audio, video or both) strands. `RECORD` invokes the continuity criteria obtained in the previous section to allocate free blocks for storing the media strands. If the media being recorded includes audio, then the file system performs silence detection and elimination. Recording continues until a subsequent `STOP` operation is issued.

PLAY [mmRopeID, interval, media] → requestID

Using the above interface, a user can retrieve any interval of any of the media of a previously recorded multimedia rope.

STOP [requestID]

When a user issues a STOP on an earlier PLAY or RECORD request, the retrieval or storage of the corresponding multimedia rope is halted.

Both RECORD and PLAY operations are non-blocking. Hence, a user may send multiple requests to the file system. The file system assigns a unique requestID to each request, and the clients use it to refer to the request subsequently. The file system accepts RECORD or PLAY requests using the admission control algorithm described in Section 3.4. Since RECORD and PLAY are continuous operations, it is desirable to allow a user to PAUSE (and later RESUME) a RECORD or a PLAY request, the file system provides the user the flexibility to specify either a *destructive* PAUSE, which causes resources to be deallocated during the PAUSE, or a *non-destructive* PAUSE, in which resources remain allocated. If a destructive PAUSE is specified, a subsequent RESUME will cause the file system to perform admission control.

In addition to RECORD, PLAY, PAUSE and RESUME operations, the file system also supports the following utilities:

```
INSERT[baseRope, position, media, withRope, withInterval]
REPLACE[baseRope, media, baseInterval,withRope,
        withInterval]
SUBSTRING[baseRope, media, interval]
CONCAT[mmRope|D1, mmRope|D2]
DELETE[baseRope, media, interval]
```

The functionality of these operations are similar to those on voice ropes in the Etherphone system. Figure 9 illustrates the `withInterval` of media strands of `withRope`,  $Rope_2$ , being `INSERTED` at position of `baseRope`,  $Rope_1$ . To guarantee real-time performance and continuity in retrieval operation, a small amount of copying of a strand may be necessary. We shall present an algorithm to bound this copying in the next section.

Any of the editing operations may be performed on any subset of media constituting a tape. An interesting

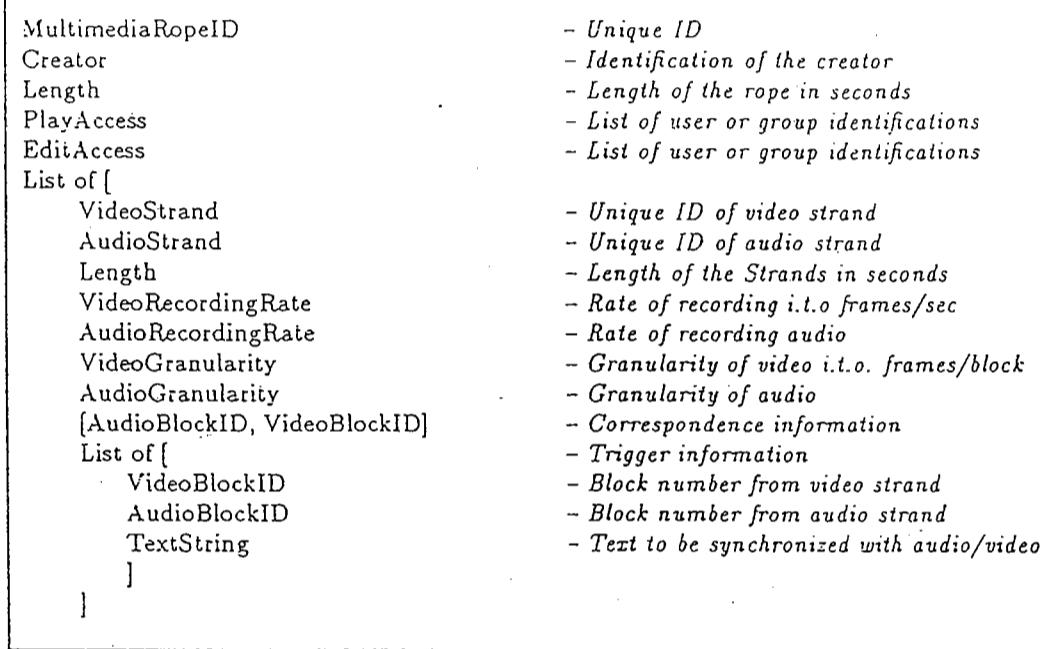


Figure 8: Data structure representing a multimedia rope

application is to merge video and audio strands recorded separately to form a multimedia rope. For example, if  $Rope_4$  contains only an audio strand, and  $Rope_5$  contains only a video strand, then the operation

```
REPLACE[baseRope: Rope4, media: video,
        baseInterval: [start:0, length: L3],
        withRope: Rope5, withInterval:
                  [start:0, length:L3]]
```

replaces the non-existent video component of  $Rope_4$  with the video component of  $Rope_5$ . The synchronization information for the resulting rope is generated by creating a correspondence between the blocks of the two strands.

#### 4.2 Maintenance of Scattering while Editing

Editing operations such as insertion and deletion may cause a multimedia rope to consist of a sequence of intervals of media strands. While immutability of a media strand guarantees that their scattering parameter is bounded and hence the continuity requirement is satisfied within each of its intervals, the scattering parameter may not be bounded while moving from the last block of one interval (of a strand) to the first block of the next interval (which may belong to the same or another

strand). Thus, discontinuities may be felt at interval boundaries during retrievals. These discontinuities can be eliminated by copying a small number of blocks of strands to which the intervals belong. We now present an algorithm to bound the number of blocks that need to be copied to guarantee continuity of retrieval.

Let us suppose that the result of an editing operation is a rope, one of whose components consists of intervals  $[a_j, a_l]$  of strand  $S_a$  and interval  $[b_j, b_l]$  of strand  $S_b$ . Let the maximum possible separation between two blocks on a disk be  $l_{sec}^{max}$ . Suppose that the scattering parameters of strands  $S_a$  and  $S_b$  are not only bounded above (from continuity requirement) but also bounded below. Let the lower bounds on the scattering parameters of  $S_a$  and  $S_b$  be  $l_{dsLower}^a$  and  $l_{dsLower}^b$  respectively. Similarly, let the upper bounds on the scattering parameters be  $l_{dsUpper}^a$  and  $l_{dsUpper}^b$  respectively. Let

$$m = \frac{l_{sec}^{max}}{l_{dsLower}^b}$$

Note that the maximum separation between the last block of  $S_a$  (which is  $a_l$ ) and block  $b_{j+m}$  of  $S_b$  is given by

$$\Delta_1 = l_{sec}^{max}$$

Also, the minimum separation between blocks  $b_{j+m/2}$  and  $b_{j+m}$  of  $S_b$  is

$$\Delta_2 = \frac{m}{2} * l_{dsLower}^b = \frac{l_{sec}^{max}}{2}$$

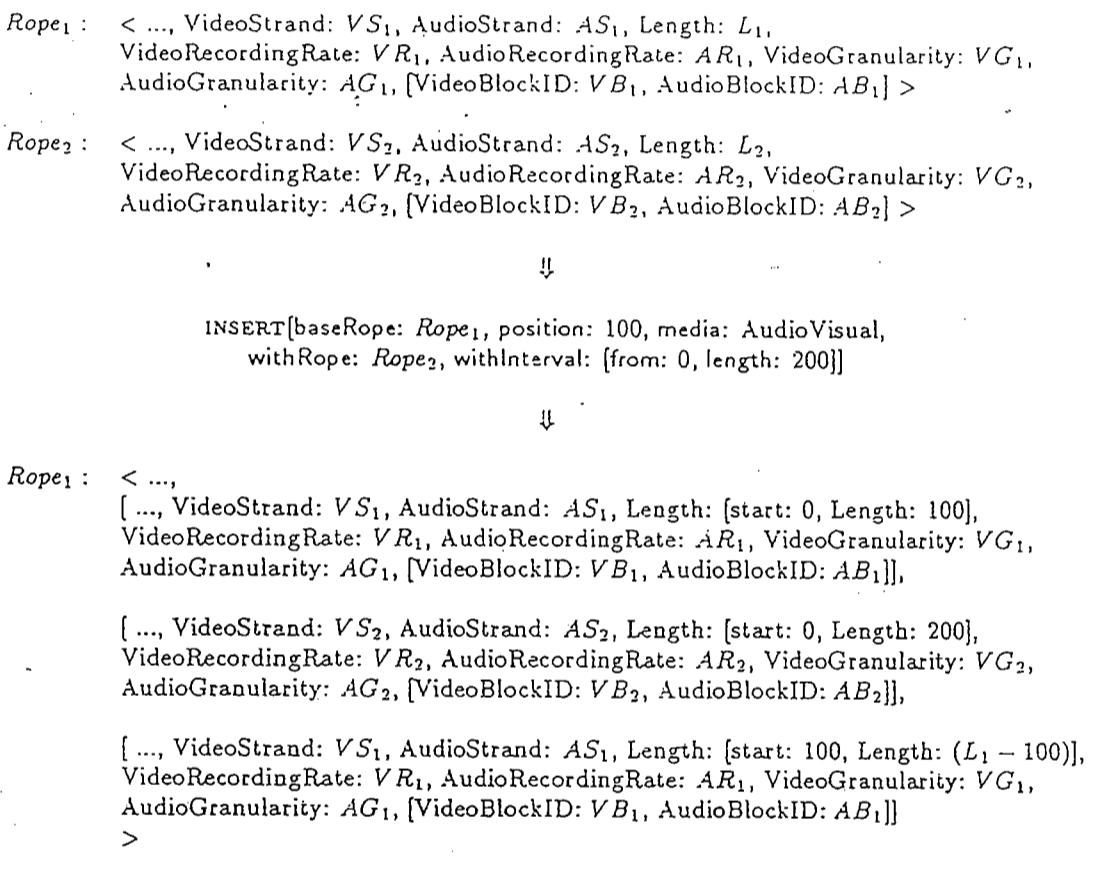


Figure 9: INSERT operation

Hence, the maximum separation between block  $a_l$  of  $S_a$  and block  $b_{f+m/2}$  of  $S_b$  is given by

$$\Delta = \Delta_1 - \Delta_2 = \frac{l_{\text{seek}}^{\max}}{2} = \frac{m}{2} * l_{\text{dsLower}}^b$$

in the best case (when the disk is sparsely occupied), and by

$$\Delta_1 = l_{\text{seek}}^{\max} = m * l_{\text{dsLower}}^b$$

in the worst case (when the disk is densely occupied). Thus, when the disk is sparsely occupied, by redistributing  $b_f, b_{f+1}, b_{f+2}, \dots, b_{f+m/2-1}$  blocks equally in the region between block  $a_l$  of  $S_a$  and block  $b_{f+m/2}$  of  $S_b$ , we can guarantee that the separation between  $a_l$  and  $b_f$  satisfies the bounds on the scattering parameter. Similar results hold when the disk is densely occupied, with block  $b_{f+m/2-1}$  replaced by block  $b_{f+m-1}$ . Thus, the maximum number of blocks of  $S_b$  required to be copied is given by

$$C_b = \frac{m}{2} = \lceil \frac{l_{\text{seek}}^{\max}}{2 * l_{\text{dsLower}}^b} \rceil \quad (19)$$

when the disk is sparsely occupied, which degrades to:

$$C_b = m = \lceil \frac{l_{\text{seek}}^{\max}}{l_{\text{dsLower}}^b} \rceil \quad (20)$$

when the disk is densely occupied (i.e., nearly full).

Alternatively, instead of the first  $C_b$  blocks of  $S_b$ , we can redistribute the last  $C_a$  blocks of  $S_a$ , where  $C_a$  is computed in a similar manner. In practice, the actual number of blocks that needs to be copied is the minimum of  $C_a$  and  $C_b$ .

It should be noted that copying creates a new strand containing only the copied blocks because (1) strands are immutable, and (2) creating a separate strand aids the process of garbage collection. A unique ID is associated with this newly generated strand, and is used in the description of the multimedia rope created as a result of the editing operation.

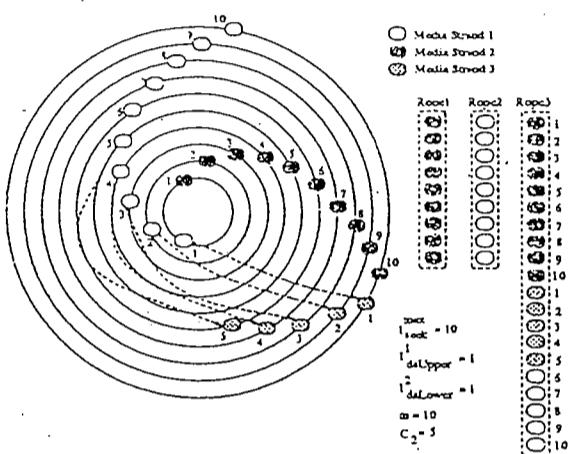


Figure 10: Copying of strands in editing operations ( $\text{Rope3} \leftarrow \text{CONCAT}[\text{Rope1}, \text{Rope2}]\right)$ )

## 5 Experience with Multimedia File Storage and Management

We have implemented a prototype testbed multimedia file system to serve as a vehicle for experimenting with policies and algorithms outlined in this paper. The hardware environment and the software architecture of our testbed system are described in the following sections:

### 5.1 Hardware Environment

Our Multimedia Laboratory is equipped with a number of multimedia stations, each consisting of a Sun SPARCstation, a PC-AT, a video camera, and a TV monitor (see Figure 11). The SPARCstations and PC-ATs are connected via Ethernets. The PC-ATs are equipped with digital video and audio processing hardware produced by UVC Corporation [6]. The audio hardware digitizes audio signals at 8 KBytes/sec. The video hardware can digitize and compress motion video at real-time rate up to NTSC broadcast with a resolution of 480x200 pixels and 12 bits of color information per pixel. Video data is stored on the local disk attached to the PC-AT, and displayed on a monitor attached to it. The operation of the PC-ATs is controlled via SPARCstations. Communication between the SPARCstations and PCs is accomplished using TCP/IP socket library.

Note that the separation of the media-processing functionality from the workstation provides reliability, performance, and flexibility: audio and video processing peripherals provide reliable media processing without compromising workstation performance on other tasks, and users of different workstations can use the media processing features without making any modifications to their workstation hardware.

### 5.2 Software Architecture

The software architecture of the prototype file system was designed to serve as a testbed for experimenting with various policies described in this paper. There are two main functional layers: the *Multimedia Storage Manager* (MSM) and the *Multimedia Rope Server* (MRS).

- *Multimedia storage manager*: This layer is responsible for physical storage of media strands on the disk. The functionality of the MSM include: determination of granularity and scattering of strands, enforcing admission control to service multiple requests simultaneously, and maintenance of scattering while editing.
- *Multimedia Rope Server*: This layer is responsible for creating and maintaining the multimedia ropes. It supports all the rope manipulation operations.

The rationale for the above layering is that:

- A decoupled design of the MRS and the MSM permits their execution on different hardware. In addition, it facilitates easy experimentation with various policies in one layer without effecting the other.
- The MRS implements the device-independent multimedia rope abstraction. The MSM implements storage device-specific algorithms, and hence, is hardware dependent.

The MRS of our testbed system is implemented on a SPARCstation, whereas the MSM is implemented on a PC-AT. Applications are compiled with a rope stub library which uses remote procedure calls to contact the MRS. The first application we implemented that uses the file system is a window-based editor to manipulate multimedia ropes. Figure 12 shows a typical editing session with the editor.

## 6 Concluding Remarks

### 6.1 Summary

We have analyzed the unique requirements of a multimedia file system such as continuous storage and retrieval of media, maintenance of synchronization between multiple media streams, and efficient manipulation of huge media objects. We have presented a model that relates disk and device characteristics (such as, disk read/write latency, and video capture/display times) to the recording rate, and derived storage granularity and scattering parameters that guarantee continuous access. The continuity requirements define an upper bound on the scattering parameter. The algorithm that bounds

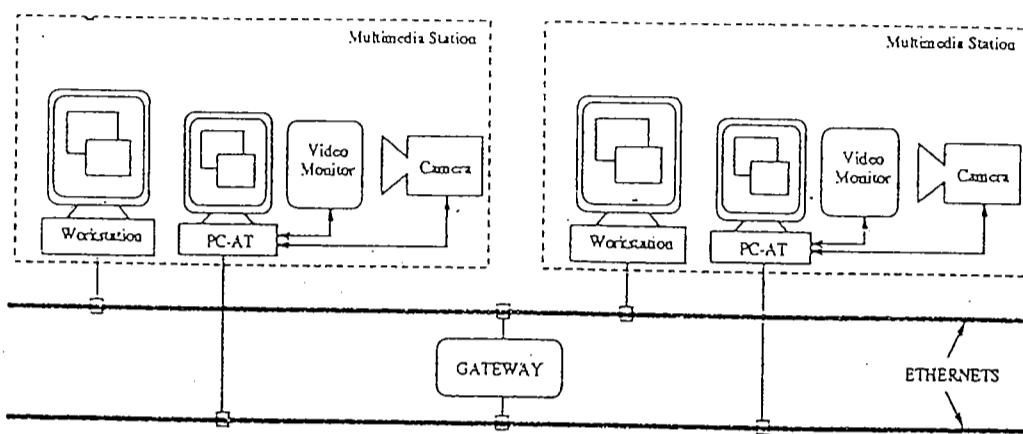


Figure 11: System configuration

the amount of copying necessary during editing operations define the lower bound of the scattering parameter. Thus, the separation between consecutive blocks of a strand must be chosen within these bounds.

In order to support multiple concurrent requests, we have presented an admission control algorithm that determines whether a new request can be accepted without violating the real-time constraints of any of the requests. The algorithm guarantees both transient and steady state continuity.

We have defined strand and rope abstractions, and have outlined an approach to maintain synchronization information among strands. We have described editing operations for multi-stranded ropes.

## 6.2 Future Work

In the storage model presented in this paper, we have assumed that video frames and disk blocks are of fixed size. However, variable rate compression of video (analogous to silence elimination in audio), such as differencing between frames, can result in varying but smaller sizes of video frames, thereby yielding better bounds for granularity and scattering. We are extending the continuity equations to incorporate such effects of compression algorithms.

Constrained scattering of blocks of a media strand can be difficult to achieve when the disk is densely utilized. When it becomes impossible to place new media strands in such a way that their scattering bounds are satisfied, the storage of existing media strands on the disk may have to be reorganized. Towards this end, we are investigating mechanisms for merging multiple media strands so as to optimize storage utilization, and we are studying techniques by which a small number of anomalies in scattering can be smoothed out.

The admission control algorithm that we have developed uses a round-robin servicing of requests in the

order in which they are received, and assumes maximum separation between blocks while switching between requests. As a result, the estimates of the maximum number of requests that can be simultaneously serviced are pessimistic. We are investigating algorithms for servicing requests in the order that minimizes (possibly, in a statistical sense) the separations between blocks, thereby minimizing the overhead of switching between requests, and optimizing the maximum number of requests that can be serviced simultaneously.

We have implemented a prototype multimedia file system, which serves as a testbed for experimentation. We are enhancing the prototype to (1) permit access over a network, and (2) provide conversational interface so that it can be accessed from within multimedia conferences.

## Acknowledgement

The initial motivation for this work comes from our involvement in the Etherphone project, which was made possible by Dan Swinehart, Doug Terry, and Polle Zelleweger. We are thankful to Walt Burkhard, Robert Bowdidge, Kashun Chan, Ingvar Aaberg, John Lindwall, Ljubisa Radivojevic, Linda Yamamoto, and Ian Harris for their contributions towards implementing the prototype multimedia file system, to Srinivas Ramanathan for helping in preparing the final version of the paper, and to Bill Walton of Compaq, Thomas Kaeppner, David Reed, and anonymous referees for their insightful comments on the paper.

## References

- [1] S. Angebranndt, R.L. Hyde, D.H. Luong, N. Siravar, and C.Schmandt. Integrating Audio and

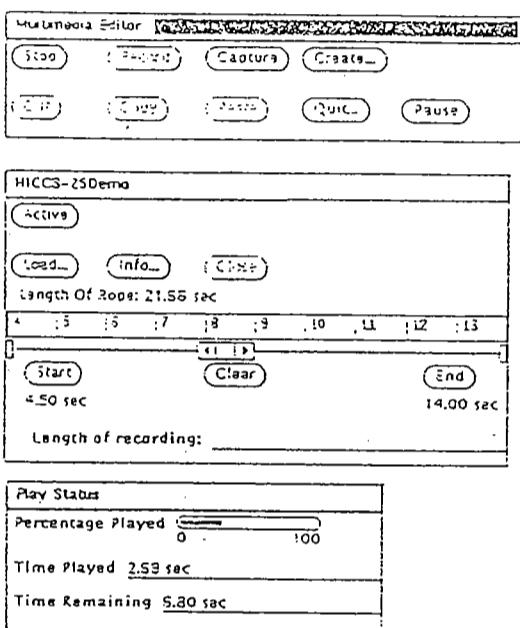


Figure 12: Window-based Multimedia Editor

- [1] Telephony in a Distributed Workstation Environment. In *Proceedings of Summer 1991 Usenix Conference*, pages 419–436, June 1991.
- [2] P. Cochrane and M. Brain. Future Optical Fiber Transmission Tech. and Networks. *IEEE Communications Magazine*, 26(11):45–60, November 1988.
- [3] J. Gait. The Optical File Cabinet: A Random-Access File System for Write-Once Optical Disks. *IEEE Computer*, 21(6):11–22, June 1988.
- [4] S. Gibbs, D. Tsichritzis, A. Fitas, D. Konstantas, and Y. Yeotgaroudakis. Muse: A Multi-Media Filing System. *IEEE Software*, 4(2):4–15, March 1987.
- [5] A. Hopper. Pandora – an experimental system for multimedia applications. *ACM Operating Systems Review*, 24(2):19–34, April 1990.
- [6] M. Leonard. Compression Chip Handles Real-Time Video and Audio. *Electronic Design*, 38(23):43–48, December 1990.
- [7] W.E. Mackay and G. Davenport. Virtual Video Editing in Interactive Multimedia Applications. *Communications of the ACM*, 32(7):302–310, July 1989.
- [8] Sun Microsystems. Multimedia File System. *Software Release*, August 1989.
- [9] Y. Mori. Multimedia Real-Time File System. In *Matsushita Electric Industrial Co.*, February 1990. private communication.
- [10] B.C. Ooi, A.D. Narasimhalu, K.Y. Wang, and I.F. Chang. Design of a Multi-Media File Server using Optical Disks for Office Applications. *IEEE Computer Society Office Automation Symposium, Gaithersburg, MD*, pages 157–163, April 1987.
- [11] C.S. Skrzypczak. The Intelligent Home of 2010. *IEEE Communications Magazine*, 25(12):31–34, December 1987.
- [12] D.B. Terry and D.C. Swinehart. Managing Stored Voice in the Etherphone System. *ACM Transactions on Computer Systems*, pages 3–27, February 1988.
- [13] R.H. Thomas, H.C. Forsdick, T.R. Crowley, R.W. Schaaf, R.S. Tomlinsin, V.M. Travers, and G.G. Robertson. Diamond: A Multimedia Message System Built on a Distributed Architecture. *IEEE Computer*, 18(12):65–78, Dec. 1985.



EOT

90 - 216 - 2317

PATENT

RECEIVED

MAR 22 1996

GROUP 2500

#3  
LDS  
3-22-96

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Inventor(s): Kulas, Charles J.  
Serial No.: 08/252,460  
Filed: May 31, 1994  
Art Unit: 2317  
Examiner: Huang, Po  
For: SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED  
INTERACTIVE PRODUCTIONS

ASSISTANT COMMISSIONER FOR PATENTS  
Washington, D. C. 20231

**REQUEST FOR EXTENSION OF TIME**

Sir:

Applicant(s) hereby petitions for an automatic extension of time in the second month to respond to the office action dated October 5, 1995.

Check number 1243 in the amount of \$190.00 is included, made payable to the Commisioner of Patents and Trademarks.

Respectfully submitted,

  
Charles J. Kulas  
Registration No. 35,809

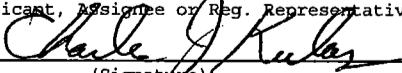
Please direct correspondence to:  
Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107

3/5/96

**CERTIFICATE OF MAILING**

I hereby certify that this document and any document referenced herein is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: 3-5-96

**CHARLES J. KULAS 35809**  
Reg. No.  
(Applicant, Assignee or Reg. Representative)

  
(Signature)

#3

ID	MCH	TPE	NAME OR ACCOUNT	C-NBR	MLEDTE	CURDTE	F-C	\$ AMOUNT
C	140	1	08252460	00006	940701	940701	101	392.00
C	270	1	08252460	00061	960311	960315	202	39.00
C	270	1	08252460	00062	960311	960315	216	190.00
C	330	1	08252460	00184	960325	960329	202	39.00
C	330	1	08252460	00185	960325	960329	217	260.00

NO MORE TRANSACTIONS

END OF YOUR QUERY



PATENT

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Case Docket No. \_\_\_\_\_

Re application of:

Inventor(s): Charles J. Kulas  
Serial No.: 08/252,460  
Filed: May 31, 1994  
Art Unit: 2317  
Examiner: Huang, Po  
For: SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED  
INTERACTIVE PRODUCTIONS

ASSISTANT COMMISSIONER FOR PATENTS

Washington, D. C. 20231

RECEIVED

MAR 22 1996

GROUP 2000

Sir:

This is an amendment in the above patent application and includes an attachment which is incorporated by reference. The signature and Certificate of Mailing below serves as a signature and Certificate of Mailing for the attachment(s) in the absence of any signature(s) thereon. The following items are enclosed.

- [X] A Certificate of Mailing.  
[X] A receipt post card.  
[X] Amendment or other response to Office Action.  
[X] Request for Extension of Time.

[ ] \_\_\_\_\_  
[ ] \_\_\_\_\_

The fee for this response has been calculated as shown below:

	CLAIMS REMAINING	HIGHEST PAID FOR	NO. EXTRA	RATE (SMALL ENT.)	FEE
TOTAL	13	20	0	x \$11	0
INDEPENDENT	5	4	1	x \$39	39
MULTIPLE DEPENDENT CLAIMS PRESENTED				\$250	0
				TOTAL \$ 39	

- [X] Enclosed is check number 1244 in the amount of the above total.  
A duplicate of this letter is enclosed.  
[X] Please date stamp and return the enclosed receipt post card.

Respectfully submitted,

Charles J. Kulas  
Registration No. 35,809

CERTIFICATE OF MAILING

I hereby certify that this document and any document referenced herein is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: 3-5-96

CHARLES J. KULAS Reg. No. 35809

(Applicant, Assignee or Reg. Representative)

(Signature)



39-202-2317

#4/A

PATENT 105  
4-4-96  
entered

RECEIVED

MAR 22 1996

GROUP 2300

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In Re Patent Application of:

Inventor(s): Kulas, Charles J.

Filed: May 31, 1994

Serial No.: 08/252,460

Art Unit: 2317

Examiner: Huang, Po

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

Assistant Commissioner for Patents

Washington, DC 20231

**AMENDMENT**

Sir:

Responsive to the Office Action dated October 5, 1995, please amend the subject application as follows:

**IN THE DRAWINGS:**

Fig. 8 has been amended to indicate "interval 314" which was not previously shown in the Figure but which was referenced at page 31, line 10 of the specification. Upon receipt of a Notice of Allowability, Applicant will provide formal drawings incorporating the amendment as indicated in red ink. The formal drawings will also comply with the Draftsperson's objections regarding the improper margins and the character of lines, numbers and reference characters.

**IN THE SPECIFICATION:**

At page 8, line 25, insert a comma after "with".

Also at page 8, line 26, insert a comma after "reading".

At page 12, line 11 delete "alternately".

A

Also at page 12, line 13 replace "displayed, or 'dropped.'" with --displayed. I.e., they are "dropped."--

At page 12, line 28, change "dropping" to --drop--.

At page 13, line 24, insert a comma after "example".

Also at page 14, line 1, replace "shows" with --includes--.

Also at page 14, line 4, change "and" to --while--.

At page 14, line 21, change "as small computer systems interface" to -- as a small computer system interface--.

#### IN THE CLAIMS:

Please insert claims 8-13 as shown below.

Please amend claims 1, 2, 4, 5 and 7 as follows:

1. (Amended) 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 for selective display in place of the first animation sequence, wherein the second animation sequence is a selectable path in the interactive production; 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. (Amended) 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] and 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;

~~A~~  
~~nd~~

displaying only the frames of the first animation sequence on the display screen to play back the first animation;  
and  
accepting signals from the user input device selecting the second animation sequence;  
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.

---

4. (Amended) 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 the CD-ROM further includes identification information associated with data describing the 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] memory including a buffer, and CD-ROM drive, the method further comprising the following steps performed under the control of the processor:

continuously reading the interleaved frames and identification information from the CD-ROM;

storing the frames of the first animation sequence into the buffer;  
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[,] performing the following steps:

using the identification information to identify the data in the buffer associated with the frames of the first animation sequence;  
discarding the identified data in the buffer associated with the frames of the first animation sequence; and

displaying [only] the frames of the second animation sequence on the display screen to play back the second animation.

5. (Amended) 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;

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 in place of displaying one or more frames corresponding to the first animation sequence.

7. (Amended) An apparatus for creating a CD-ROM disc with interleaved animation sequences[,] comprising:

A CD-ROM disc:

[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, wherein the frames of the first animation sequence are designed for playback to produce a first visual sequence and wherein the frames of the second animation sequence are designed for playback to produce a second visual sequence to be shown in place of the first visual sequence, 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.~~

*3* 8. (New) The method of claim ~~7~~, further comprising the steps of:

writing a tag, wherein the tag indicates that data representing one or more frames belongs to a specific sequence; and

using the tag during the play back of the interactive production to determine whether to display the associated data representing one or more frames.

*4* 9. (New) The method of claim ~~8~~, wherein the computer system includes memory comprising a buffer, wherein frames are stored into the buffer prior to display on the display device, the method further comprising the step of:

using the tag to prevent loading of the associated data representing one or more frames into the buffer.

*5* 10. (New) The method of claim ~~4~~, wherein the identification information comprises: a tag for indicating that one or more frames are from a particular sequence.

*5 6 7 8 C 2* 11. (New) 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 CD-ROM

further including tags associated with one or more frames on the CD-ROM, wherein the tags indicate which sequence the one or more frames associated with a given tag belongs to, the apparatus comprising:

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

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

means for reading the tags from the CD-ROM;

means for using the tags to determine which of the retrieved frames are stored into the buffer;

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 using the tags to display on the display screen only frames corresponding to the second animation sequence.

12. (New) A programmed CD-ROM disk having a data track read by a CD-ROM read head, the CD-ROM disk comprising:

a first portion of frames of a first animation sequence encoded onto the data track;

a first portion of frames of a second animation sequence encoded onto the data track after the first portion of frames of the first animation sequence;

a second portion of frames of the first animation sequence encoded onto the data track after the first portion of frames of the second animation sequence; and

a second portion of frames of the second animation sequence encoded onto the data track after the second portion of frames of the first animation sequence so that, upon playback, either the first or second animation sequence is selectively displayable on a display device by reading and displaying the frames from the desired animation sequence without the need to reposition the CD-ROM read head.

*A4*  
*end*

13. (New) The programmed CD-ROM disk of claim 12, further comprising:  
identification information stored onto the CD-ROM disk, wherein the identification  
information identifies one or more portions of frames as belonging to a sequence.

#### REMARKS

This application has been carefully reviewed in view of the above office action in which claims 1-7 were rejected under 35 U.S.C. § 103 as being unpatentable over Nguyen in view of Shusuke. The disclosure and drawings were objected to.

#### **THE 35 U.S.C. § 103 REJECTION**

Claims 1, 4, 5, 7, 11 and 12 are independent claims. Each of these independent claims includes a limitation not disclosed by, nor made obvious in view of, the prior art.

Applicant's invention is a method and apparatus for producing and playing back an interactive production on a CD-ROM. Claims 1, 5, 7 and 12 each recite first and second animation sequences. Either the first or second animation sequence is selected for display when the CD-ROM is used to play back an interactive production. Note that only one of the sequences is displayed. For example, in claim 1, the second animation sequence is displayed "in place of the first animation sequence" and is "a selectable path in the interactive production." Similarly, in claims 5 and 7 the second sequence is displayed "in place of" the first animation sequence. Claim 12 recites a programmed CD-ROM where "either the first or second animation sequence is selectively displayable".

Nguyen discloses a system that mixes animation sequences and computer graphics. Nguyen's purpose is to allow a single presentation to use animation sequences and computer graphics to aid in presenting information, such as where graphics are "overlaid" onto animations. Thus, while the present invention is directed to interactive productions where one or the other visual sequence is chosen depending upon a user's input, Nguyen's application is the more traditional non-interactive presentation where different images are mixed onto a single image display to provide a single coherent presentation. This is illustrated by Nguyen's Fig. 2 which shows digital mixer 32 providing a single output from the inputs of lines 18 and 38.

Further, Nguyen's "mixing" is merely overlaying of a graphics object over a portion of an animation. Nguyen at col. 7, lines 35-44. Not only does Nguyen not relate to two animation sequences but neither the graphics object nor the animation are shown "in place of" an animation sequence as recited in Applicant's claims 1, 5, 7 and 12 (equivalently reciting displaying "either" the first or second sequences).

Shusuke uses different interleaving rules to "obtain a natural reproduced picture" (emphasis added). Shusuke and Nguyen are similar in that they only disclose systems that are designed to present visual information in a traditional way. It is neither desirable nor obvious to combine Shusuke and Nguyen to obtain an interleaved sequence of first and second visual sequences of digital frames where the second visual sequence is "to be shown in place of the first visual sequence" as recited variously by Applicant's claims. To do so in Shusuke defeats Shusuke's goal of obtaining a "naturally reproduced picture" while to do so in Nguyen would produce a meaningless animation showing rapid cutting between two visual sequences.

Nguyen's goal of "simultaneously" presenting animations and computer graphics (Nguyen at col. 1, line 54) is incompatible with the present invention's goal -- that of creating an efficient interactive production by using interleaved animation sequences where only one of the sequences is to be displayed. Shusuke's goal, or purpose, that of using interleaving to achieve efficient video data compression and decompression while still preserving the original quality of the entire same animation sequence, or "picture," is completely incompatible with the present invention as claimed which interleaves frames from different animation sequences so that they are more quickly accessed on a CD-ROM. Therefore, Nguyen and Shusuke, alone or in combination, do not render the present invention obvious.

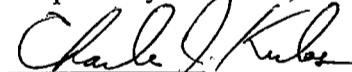
None of the cited references relates to other specific limitations in the claims regarding the problems of using CD-ROMs in interactive productions. Each of the independent claims recites an "interactive production" on a "CD-ROM" both in the preamble of the claims and in the elements or steps in the body of the claim. Claim 1 specifically recites "[a] method for creating an interactive production on a CD-ROM". The "writing" step of claim 1 is directed to writing frames to the "CD-ROM" to create the "interactive production." Neither of the Nguyen or Shusuke references mentions an interactive production much less a way to write frame data on a CD-ROM to allow efficient

selection of animation sequences. While Nguyen illustrates a CD-ROM device in Fig. 1, it is merely a player and not a recorder. There is no mention in the cited references of any writing of information onto a CD-ROM data track. As discussed in Applicant's specification, the CD-ROM has the property of being a highly linear device with an access time sacrifice whenever data is not accessed linearly. This is the problem that the present invention overcomes. Without addressing CD-ROMs, much less the problem of CD-ROM access times, it is not surprising to find that the references do not relate to the invention as claimed.

Independent claim 4 recites "identification information" which can be, as recited in dependent claim 10, "a tag for indicating that one or more frames are from a particular sequence." Independent claim 11 also recites the use of tags to aid in the selection of an animation sequence. Other dependent claims also recite the use of tags. Tags are described in the specification as filed at, for example, page 8 lines 27-34; and at page 18, lines 11-23. The prior art does not address writing information to a CD-ROM disk, much less the writing of specific "identification information" or "tags" to a CD-ROM.

In view of this communication, all claims are now believed to be in condition for allowance and such is respectfully requested at an early date. The Examiner is invited to contact the undersigned attorney at 408-955-5485, if necessary.

Respectfully submitted,

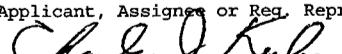


Charles J. Kulas

Registration No. 35,809

Dated: March 5, 1996

Please Send Correspondence to:  
Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107  
Phone: (408)955-5485  
Fax: (408)955-5490

<b>CERTIFICATE OF MAILING</b>	
I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: <u>March 5, 1996</u>	
CHARLES J. KULAS Reg. No. 35,809	
(Applicant, Assignee or Reg. Representative)	
 (Signature)	

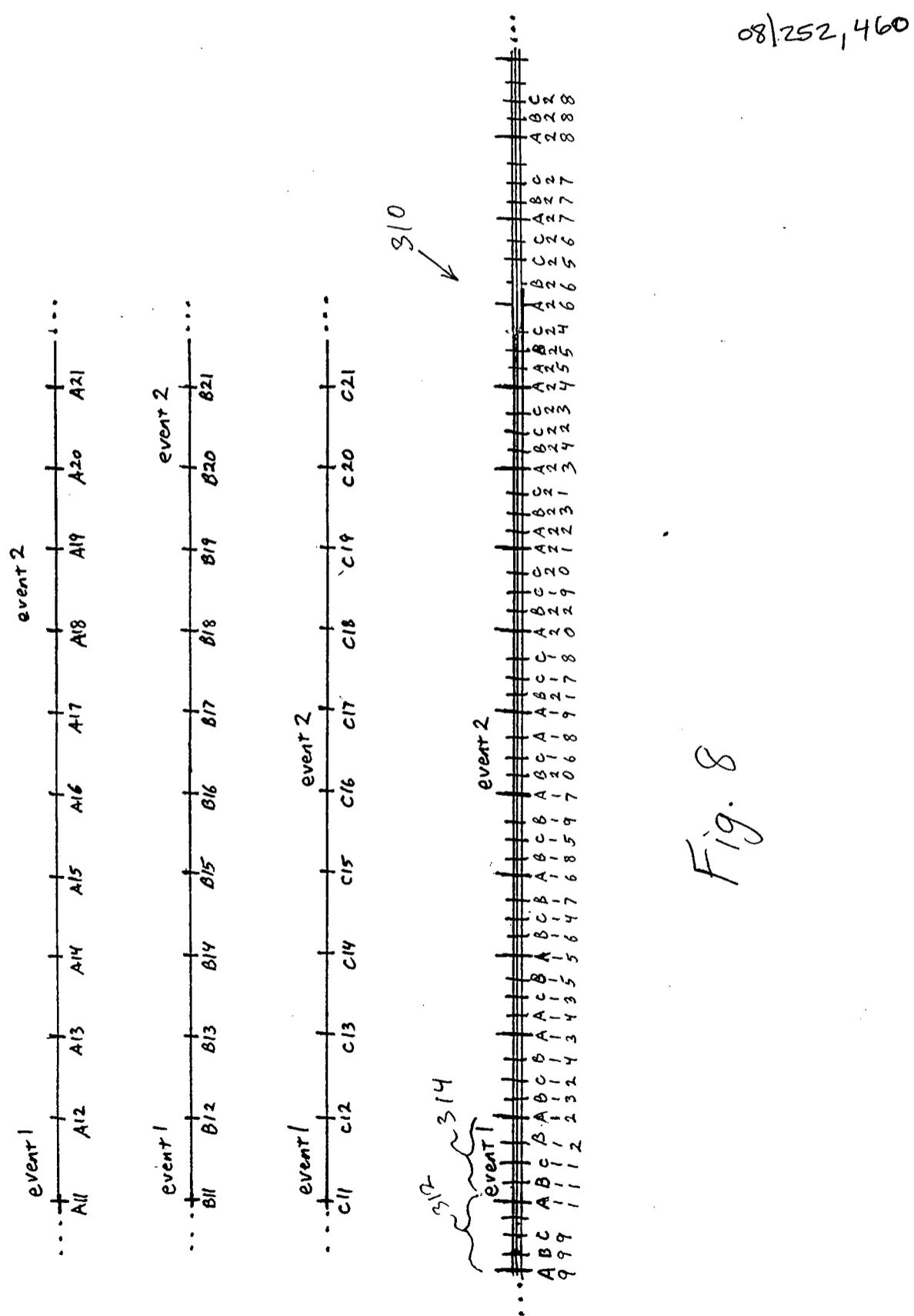


Fig. 8



34-202 (6 231)

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED

In Re Patent Application of:

Inventor(s): Kulas, Charles J.

Filed: May 31, 1994

Serial No.: 08/252,460

Art Unit: 2317

Examiner: Huang, Po

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

APR 02 1996

GROUP 2300

Assistant Commissioner for Patents

Washington, DC 20231

Copy of #  
4/A

AMENDMENT

Sir:

Responsive to the Office Action dated October 5, 1995, please amend the subject application as follows:

IN THE DRAWINGS:

Fig. 8 has been amended to indicate "interval 314" which was not previously shown in the Figure but which was referenced at page 31, line 10 of the specification. Upon receipt of a Notice of Allowability, Applicant will provide formal drawings incorporating the amendment as indicated in red ink. The formal drawings will also comply with the Draftsperson's objections regarding the improper margins and the character of lines, numbers and reference characters.

IN THE SPECIFICATION:

At page 8, line 25, insert a comma after "with".

Also at page 8, line 26, insert a comma after "reading".

At page 12, line 11 delete "alternately".

330 DS 03/29/96 08252460  
1 202 39.00 CK

Also at page 12, line 13 replace “displayed, or ‘dropped.’” with --displayed. I.e., they are “dropped.”--

At page 12, line 28, change “dropping” to --drop--.

At page 13, line 24, insert a comma after “example”.

Also at page 14, line 1, replace “shows” with --includes--.

Also at page 14, line 4, change “and” to --while--.

At page 14, line 21, change “as small computer systems interface” to -- as a small computer system interface--.

#### **IN THE CLAIMS:**

Please insert claims 8-13 as shown below.

Please amend claims 1, 2, 4, 5 and 7 as follows:

1. (Amended) 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 for selective display in place of the first animation sequence, wherein the second animation sequence is a selectable path in the interactive production; 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. (Amended) 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] and 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.

4. (Amended) 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 the CD-ROM further includes identification information associated with data describing the 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] memory including a buffer, and CD-ROM drive, the method further comprising the following steps performed under the control of the processor:

continuously reading the interleaved frames and identification information from the CD-ROM;

storing the frames of the first animation sequence into the buffer;

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[,] performing the following steps:

using the identification information to identify the data in the buffer associated with the frames of the first animation sequence;

discarding the identified data in the buffer associated with the frames of the first animation sequence; and

displaying [only] the frames of the second animation sequence on the display screen to play back the second animation.

5. (Amended) 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;

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 in place of displaying one or more frames corresponding to the first animation sequence.

7. (Amended) An apparatus for creating a CD-ROM disc with interleaved animation sequences[,] comprising:

A CD-ROM disc:

[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, wherein the frames of the first animation sequence are designed for playback to produce a first visual sequence and wherein the frames of the second animation sequence are designed for playback to produce a second visual sequence to be shown in place of the first visual sequence, 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.

8. (New) The method of claim 2, further comprising the steps of:

writing a tag, wherein the tag indicates that data representing one or more frames belongs to a specific sequence; and

using the tag during the play back of the interactive production to determine whether to display the associated data representing one or more frames.

9. (New) The method of claim 8, wherein the computer system includes memory comprising a buffer, wherein frames are stored into the buffer prior to display on the display device, the method further comprising the step of:

using the tag to prevent loading of the associated data representing one or more frames into the buffer.

10. (New) The method of claim 4, wherein the identification information comprises: a tag for indicating that one or more frames are from a particular sequence.

11. (New) 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 CD-ROM

further including tags associated with one or more frames on the CD-ROM, wherein the tags indicate which sequence the one or more frames associated with a given tag belongs to, the apparatus comprising:

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

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

means for reading the tags from the CD-ROM;

means for using the tags to determine which of the retrieved frames are stored into the buffer;

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 using the tags to display on the display screen only frames corresponding to the second animation sequence.

12. (New) A programmed CD-ROM disk having a data track read by a CD-ROM read head, the CD-ROM disk comprising:

a first portion of frames of a first animation sequence encoded onto the data track;

a first portion of frames of a second animation sequence encoded onto the data track after the first portion of frames of the first animation sequence;

a second portion of frames of the first animation sequence encoded onto the data track after the first portion of frames of the second animation sequence; and

a second portion of frames of the second animation sequence encoded onto the data track after the second portion of frames of the first animation sequence so that, upon playback, either the first or second animation sequence is selectively displayable on a display device by reading and displaying the frames from the desired animation sequence without the need to reposition the CD-ROM read head.

13. (New) The programmed CD-ROM disk of claim 12, further comprising:  
identification information stored onto the CD-ROM disk, wherein the identification  
information identifies one or more portions of frames as belonging to a sequence.

#### REMARKS

This application has been carefully reviewed in view of the above office action in which claims 1-7 were rejected under 35 U.S.C. § 103 as being unpatentable over Nguyen in view of Shusuke. The disclosure and drawings were objected to.

#### **THE 35 U.S.C. § 103 REJECTION**

Claims 1, 4, 5, 7, 11 and 12 are independent claims. Each of these independent claims includes a limitation not disclosed by, nor made obvious in view of, the prior art.

Applicant's invention is a method and apparatus for producing and playing back an interactive production on a CD-ROM. Claims 1, 5, 7 and 12 each recite first and second animation sequences. Either the first or second animation sequence is selected for display when the CD-ROM is used to play back an interactive production. Note that only one of the sequences is displayed. For example, in claim 1, the second animation sequence is displayed "in place of the first animation sequence" and is "a selectable path in the interactive production." Similarly, in claims 5 and 7 the second sequence is displayed "in place of" the first animation sequence. Claim 12 recites a programmed CD-ROM where "either the first or second animation sequence is selectively displayable".

Nguyen discloses a system that mixes animation sequences and computer graphics. Nguyen's purpose is to allow a single presentation to use animation sequences and computer graphics to aid in presenting information, such as where graphics are "overlaid" onto animations. Thus, while the present invention is directed to interactive productions where one or the other visual sequence is chosen depending upon a user's input, Nguyen's application is the more traditional non-interactive presentation where different images are mixed onto a single image display to provide a single coherent presentation. This is illustrated by Nguyen's Fig. 2 which shows digital mixer 32 providing a single output from the inputs of lines 18 and 38.

Further, Nguyen's "mixing" is merely overlaying of a graphics object over a portion of an animation. Nguyen at col. 7, lines 35-44. Not only does Nguyen not relate to two animation sequences but neither the graphics object nor the animation are shown "in place of" an animation sequence as recited in Applicant's claims 1, 5, 7 and 12 (equivalently reciting displaying "either" the first or second sequences).

Shusuke uses different interleaving rules to "obtain a natural reproduced picture" (emphasis added). Shusuke and Nguyen are similar in that they only disclose systems that are designed to present visual information in a traditional way. It is neither desirable nor obvious to combine Shusuke and Nguyen to obtain an interleaved sequence of first and second visual sequences of digital frames where the second visual sequence is "to be shown in place of the first visual sequence" as recited variously by Applicant's claims. To do so in Shusuke defeats Shusuke's goal of obtaining a "naturally reproduced picture" while to do so in Nguyen would produce a meaningless animation showing rapid cutting between two visual sequences.

Nguyen's goal of "simultaneously" presenting animations and computer graphics (Nguyen at col. 1, line 54) is incompatible with the present invention's goal -- that of creating an efficient interactive production by using interleaved animation sequences where only one of the sequences is to be displayed. Shusuke's goal, or purpose, that of using interleaving to achieve efficient video data compression and decompression while still preserving the original quality of the entire same animation sequence, or "picture," is completely incompatible with the present invention as claimed which interleaves frames from different animation sequences so that they are more quickly accessed on a CD-ROM. Therefore, Nguyen and Shusuke, alone or in combination, do not render the present invention obvious.

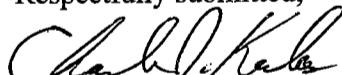
None of the cited references relates to other specific limitations in the claims regarding the problems of using CD-ROMs in interactive productions. Each of the independent claims recites an "interactive production" on a "CD-ROM" both in the preamble of the claims and in the elements or steps in the body of the claim. Claim 1 specifically recites "[a] method for creating an interactive production on a CD-ROM". The "writing" step of claim 1 is directed to writing frames to the "CD-ROM" to create the "interactive production." Neither of the Nguyen or Shusuke references mentions an interactive production much less a way to write frame data on a CD-ROM to allow efficient

selection of animation sequences. While Nguyen illustrates a CD-ROM device in Fig. 1, it is merely a player and not a recorder. There is no mention in the cited references of any writing of information onto a CD-ROM data track. As discussed in Applicant's specification, the CD-ROM has the property of being a highly linear device with an access time sacrifice whenever data is not accessed linearly. This is the problem that the present invention overcomes. Without addressing CD-ROMs, much less the problem of CD-ROM access times, it is not surprising to find that the references do not relate to the invention as claimed.

Independent claim 4 recites "identification information" which can be, as recited in dependent claim 10, "a tag for indicating that one or more frames are from a particular sequence." Independent claim 11 also recites the use of tags to aid in the selection of an animation sequence. Other dependent claims also recite the use of tags. Tags are described in the specification as filed at, for example, page 8 lines 27-34; and at page 18, lines 11-23. The prior art does not address writing information to a CD-ROM disk, much less the writing of specific "identification information" or "tags" to a CD-ROM.

In view of this communication, all claims are now believed to be in condition for allowance and such is respectfully requested at an early date. The Examiner is invited to contact the undersigned attorney at 408-955-5485, if necessary.

Respectfully submitted,

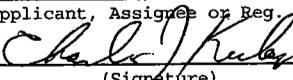


Charles J. Kulas

Registration No. 35,809

Dated: March 22, 1996

Please Send Correspondence to:  
Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107  
Phone: (408)955-5485  
Fax: (408)955-5490

<b>CERTIFICATE OF MAILING</b>	
I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: <u>3-22-96</u>	
CHARLES J. KULAS Reg. No. 35,809	
(Applicant, Assignee or Reg. Representative)	
 (Signature)	

## PATENT

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Case Docket No. CJK-1

In re application of:

Inventor(s): Charles J. Kulas

Serial No.: 08/252,460

Filed: May 31, 1994

Art. Unit: 2317

Examiner: Huang, Po

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

ASSISTANT COMMISSIONER FOR PATENTS

Washington, D. C. 20231

RECEIVED

APR 02 1996

GROUP 2500

## Patent and Trademark Office:

This is an additional fee paid in the above patent application and includes an attachment which is incorporated by reference. Lavinia Johnson of Group 230 informs the undersigned that the previously sent amendment is deficient in the amount of \$39 as the number of independent claims was miscalculated. The signature and Certificate of Mailing below serves as a signature and Certificate of Mailing for the attachment(s) in the absence of any signature(s) thereon. The following items are enclosed.

- A Certificate of Mailing.  
 A receipt post card.  
 Amendment or other response to Office Action.  
 Request for Extension of Time.
- [ ] \_\_\_\_\_  
[ ] \_\_\_\_\_

The fee for this response has been calculated as shown below:

	CLAIMS REMAINING	HIGHEST PAID FOR	NO. EXTRA	RATE (SMALL ENT.)	FEE
TOTAL	13	20	0	x \$11	0
INDEPENDENT	6	5	1	x \$39	39
MULTIPLE DEPENDENT CLAIMS PRESENTED				\$250	0
					TOTAL \$ 39

Enclosed is check number 1258 in the amount of the above total.

A duplicate of this letter is enclosed.

Please date stamp and return the enclosed receipt post card.

Respectfully submitted,

Charles J. Kulas  
Registration No. 35,809

## CERTIFICATE OF MAILING

I hereby certify that this document and any document referenced herein is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: 3-22-96

CHARLES J. KULAS Reg. No 35,809  
(Applicant, Assignee or Reg. Representative)

(Signature)



260-217 (P 231)  
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED #5  
APR 6 2 1996 BH  
4-5-96  
GROUP 2300

In re application of:

Inventor(s): Kulas, Charles J.  
Serial No.: 08/252,460  
Filed: May 31, 1994  
Art Unit: 2317  
Examiner: Huang, Po  
For: SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED  
INTERACTIVE PRODUCTIONS

ASSISTANT COMMISSIONER FOR PATENTS  
Washington, D. C. 20231

**REQUEST FOR EXTENSION OF TIME**

Sir:

Applicant(s) hereby petitions for an automatic extension of time in the third month to respond to the office action dated October 5, 1995, thereby extending the time for response to April 5, 1996.

Check number 1257 in the amount of \$260.00 is included, made payable to the Commissioner of Patents and Trademarks. This amount is the additional amount needed to meet the \$450.00 fee for a third month extension since, previously, the two month extension fee of \$190.00 has been paid regarding the present Response.

Respectfully submitted,

Charles J. Kulas  
Registration No. 35,809

Please direct correspondence to:  
Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107

3/22/96

<b>CERTIFICATE OF MAILING</b>	
I hereby certify that this document and any document referenced herein is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: <u>3-22-96</u>	
CHARLES J. KULAS (Applicant, Assignee or Reg. Representative)	35,809 Reg. No.  (Signature)

330 DS 03/29/96 08252460  
1 217 260.00 CK



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

HUANG, P

ART UNIT	PAPER NUMBER
----------	--------------

B3M1/0607

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

2317

6

DATE MAILED:

06/07/96

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 3/25/96  This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), 0 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 - 13 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 - 13 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 3/11/96, 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 appears 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

Serial Number: 08/252,460

-1-

Art Unit: 2317

1. Claims 1 - 13 are presented for examination.
2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office Action.
3. Applicant's arguments with respect to claims 1 - 7 have been considered but are deemed to be moot in view of the new grounds of rejection. Claims 8 - 13 have been added. The following rejections now apply.
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 negated 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.

Serial Number: 08/252,460

-2-

Art Unit: 2317

5. Claims 1 - 13 are rejected under 35 U.S.C. § 103 as being unpatentable over Beachy, patent no. 5,502,807 in view of Cooper, patent no. 4,789,894.

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

creating a first animation sequence [col. 1 lines 36 - 39];  
and

creating a second animation sequence [obvious in an interactive production].

Beachy does not teach interleaving the frames of the animation sequences and the writing it in a CD-ROM. Cooper teaches another data processing system which perform interleaving [abstract and col. 8 lines 35 - 68]. Although Cooper does not explicitly teach writing the frames to the CD-ROM. However, one of ordinary skill in the art would have recognized that CD-ROM interactive production is well known in the art.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Beachy and Cooper because they both directed the problem of data processing system. Cooper's teaching and CD-ROM recording, which is well known in the art, would provide flexibility to Beachy's teaching by allowing one to create an interactive production on CD-ROM.

Serial Number: 08/252,460

-5-

Art Unit: 2317

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 claim 4, Cooper teaches the use of identification information [col. 7 lines 11 - 15];  
storing frames into the buffer [col. 8 lines 38 - 42];  
displaying only the frames of the first animation sequence [col. 8 lines 63 - 68]; and  
in response to the signals, using the identification information to display the selected path [col. 8 lines 35 - 68].

10. As to claim 5, a computer system, a CD-ROM drive, means responsive to signals from the user input device, and displaying means are well known in the art.

11. As to claim 6, selection means and decompression means are well known in the art.

Serial Number: 08/252,460

-6-

Art Unit: 2317

12. As to claim 7, CD-ROM disc, storage means, frame selection means, combining means, and control means are well known in the art.

13. As to claims 8 and 10, writing and using a tag is well known in the art.

14. As to claim 9, using the tag to prevent loading of the associated data is a matter of design choice.

15. As to claim 11, it is similar to claim 4, therefore, it is rejected under the same rational.

16. As to claim 12, it is similar to claim 1, therefore, it is rejected under the same rational.

17. As to claim 13, identification information stored onto the CD-ROM disk is well known in the art.

18. 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.

Serial Number: 08/252,460

-7-

Art Unit: 2317

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) 308-5359.

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

June 3, 1996

  
THOMAS C. LEE  
SUPERVISORY PATENT EXAMINER  
230

<b><i>Notice of References Cited</i></b>			Application No.	Applicant(s)			
			<b>08/252,460</b>	<b>Charles J. Kulas</b>			
			Examiner	Group Art Unit			
			<b>Po Huang</b>	<b>2317</b>	<b>Page 1 of 1</b>		
<b>U.S. PATENT DOCUMENTS</b>							
	DOCUMENT NO.	DATE	NAME			CLASS	SUBCLASS
A	3,743,087	07/17/73	Harrison, III et al.			345	22
B	4,789,894	12/06/88	Cooper			348	155
C	5,359,468	10/25/94	Rhodes et al.			360	48
D	5,502,807	03/26/96	Beachy			395	152
E	5,519,825	05/21/96	Naughton et al.			395	152
F							
G							
H							
I							
J							
K							
L							
M							
<b>FOREIGN PATENT DOCUMENTS</b>							
	DOCUMENT NO.	DATE	COUNTRY	NAME		CLASS	SUBCLASS
N							
O							
P							
Q							
R							
S							
T							
<b>NON-PATENT DOCUMENTS</b>							
	DOCUMENT (Including Author, Title, Source, and Pertinent Pages)					DATE	
U							
V							
W							
X							

**United States Patent** [19]  
**Harrison, III et al.**

[11] **3,747,087**

[45] **July 17, 1973**

[54] **DIGITALLY CONTROLLED COMPUTER ANIMATION GENERATING SYSTEM**

[75] Inventors: Lee Harrison, III, Camarillo, Calif.; Francis J. Honey; Edwin J. Tajchman, both of Denver, Colo.; Marshall M. Parker, Lakewood, Colo.

[73] Assignee: Computer Image corporation, Denver, Colo.

[22] Filed: June 25, 1971

[21] Appl. No.: 156,762

[52] U.S. Cl. .... 340/324 AD, 178/6.8, 178/DIG. 6, 315/19

[51] Int. Cl. .... G06f 3/14

[58] Field of Search ..... 340/324 A, 324 AD; 315/19, 24; 178/6.8, DIG. 6, DIG. 35

[56] **References Cited**

UNITED STATES PATENTS

3,537,096	10/1970	Hatfield	..... 340/324 A
3,422,419	1/1969	Mathews et al.	..... 178/6.8 X
3,379,833	4/1968	Hecker et al.	..... 315/19 X
3,539,860	11/1970	Max et al.	..... 315/22

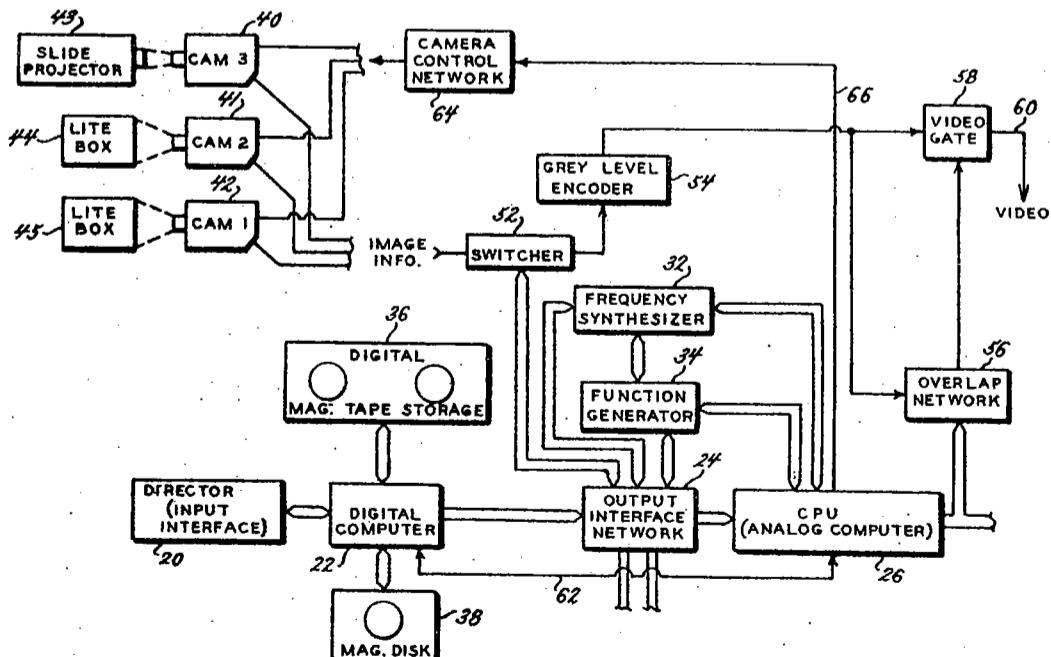
3,376,465	4/1968	Corpew	..... 315/10
3,517,122	6/1960	Owen	..... 178/DIG. 35
2,784,246	3/1957	Hurford	..... 178/6.8
3,180,932	4/1965	Oppenheimer	..... 178/6.8
3,191,169	6/1965	Shulman et al.	..... 340/324 A X
3,497,614	2/1970	Petrocelli et al.	..... 178/6.8 X
3,641,260	2/1972	Herndon	..... 315/24

Primary Examiner—David L. Trafton  
Attorney—Edmund C. Rogers et al.

[57] **ABSTRACT**

This invention relates to a system for automatically producing an animation sequence and includes an analog portion for generating output signals representing one or more sections of a raster on which images viewed by a video camera can be produced. Analog inputs to the analog portion define the parameters of the raster sections to effectively define the shape of each part of the viewed image produced thereon. The analog inputs to the analog portion are digitally controlled by signals from a digital computer portion which establishes these digital control signals from information fed to it from a director or a recording means.

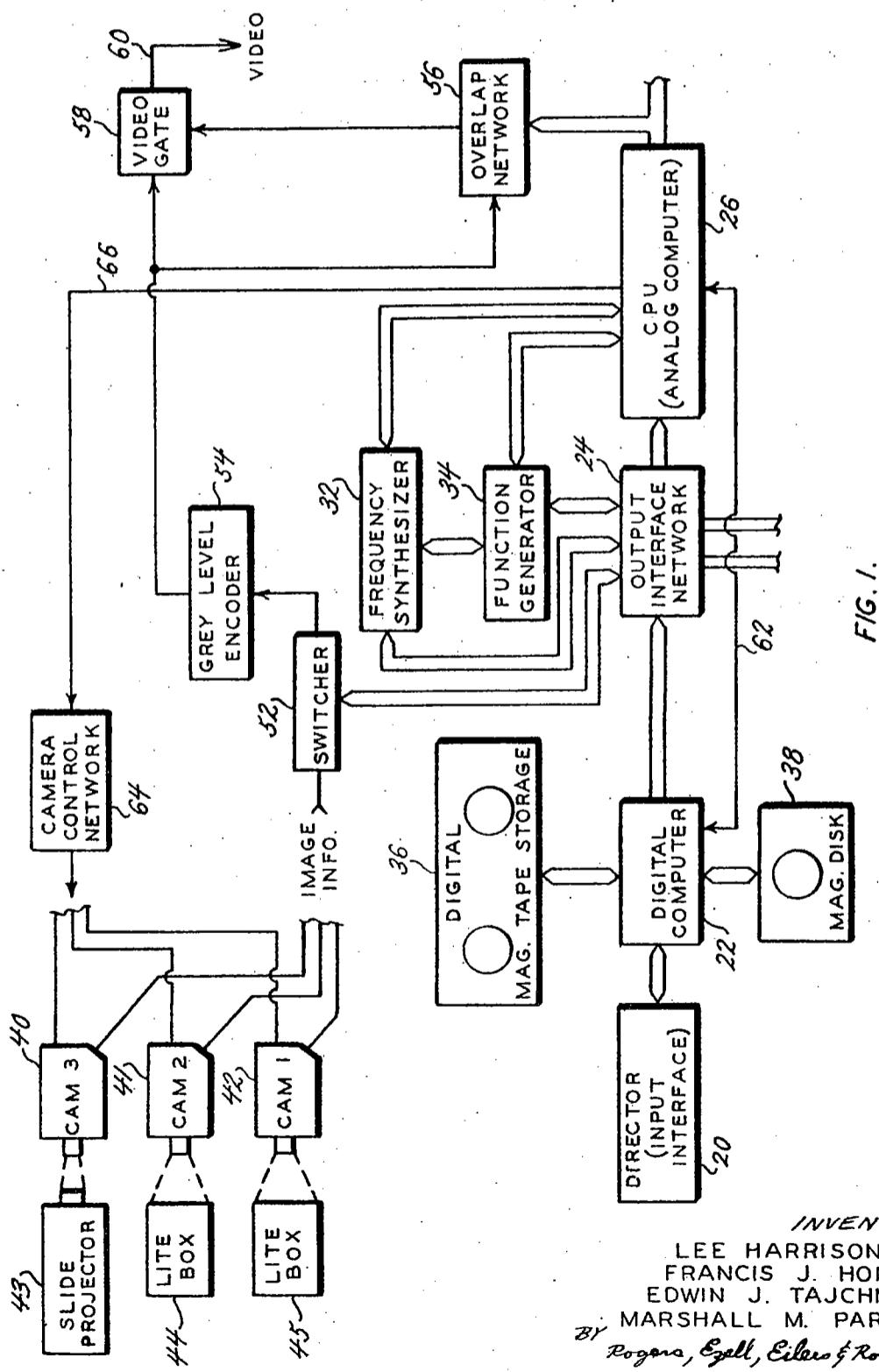
41 Claims, 13 Drawing Figures



PATENTED JUL 17 1973

3,747,087

SHEET 1 OF 7



F/G. 1.

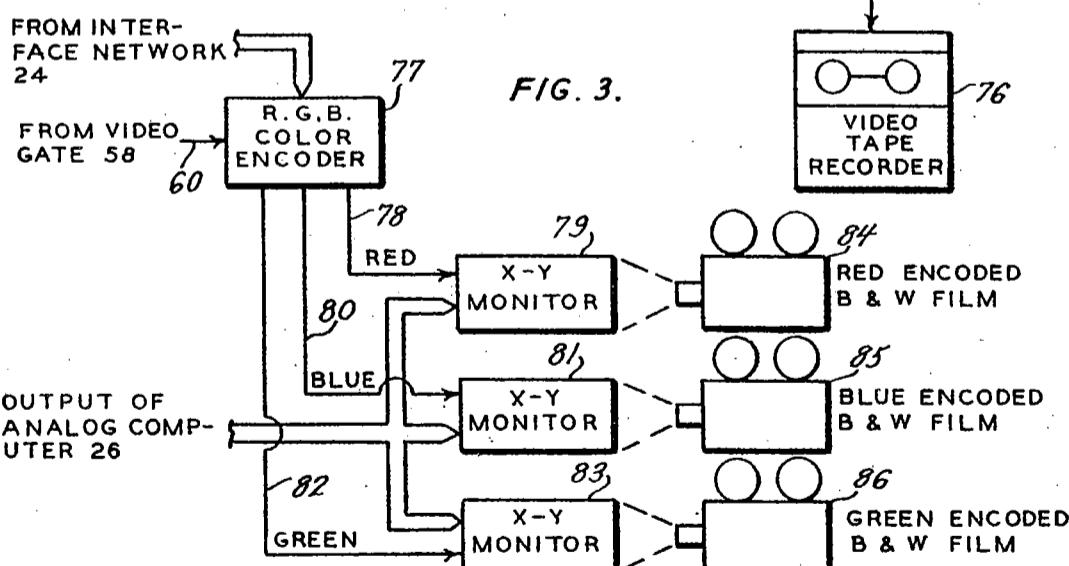
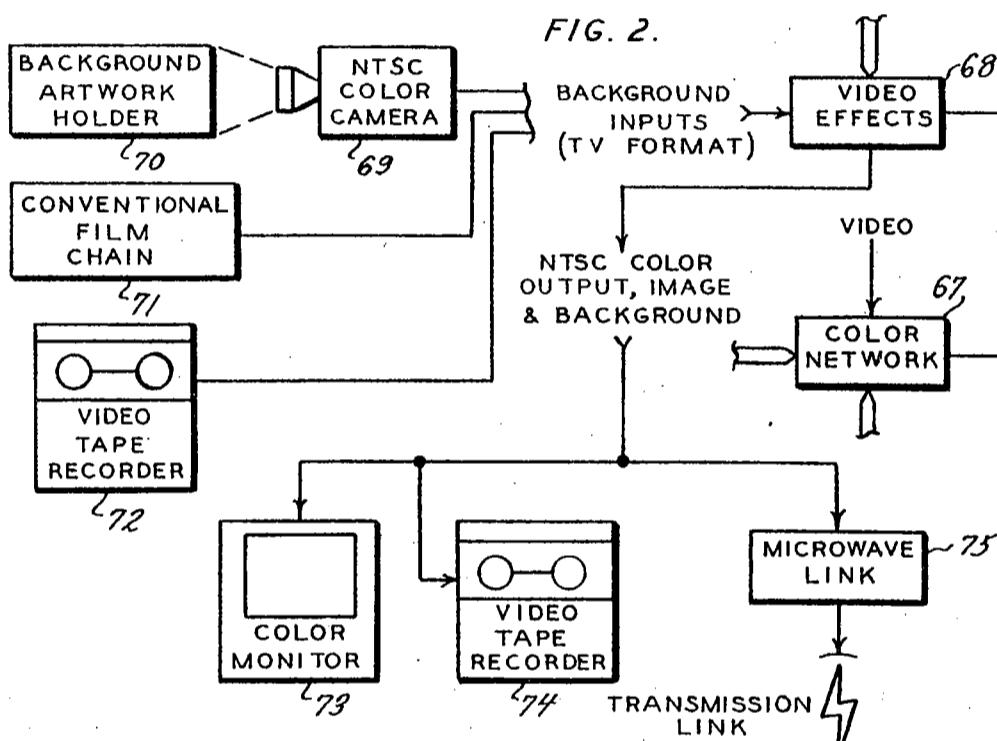
INVENTORS  
LEE HARRISON III  
FRANCIS J. HONEY  
EDWIN J. TAICHMAN  
MARSHALL M. PARKER  
BY *Rogers, Egell, Eiler & Robbins*

THEIR ATTORNEYS

PATENTED JUL 17 1973

3,747,087

SHEET 2 OF 7



INVENTORS  
LEE HARRISON III  
FRANCIS J. HONEY  
EDWIN J. TAJCHMAN  
MARSHALL M. PARKER  
By Rogers, Ezell, Ciles & Robbins  
THEIR ATTORNEYS

PATENTED JUL 17 1973

3,747,087

SHEET 3 OF 7

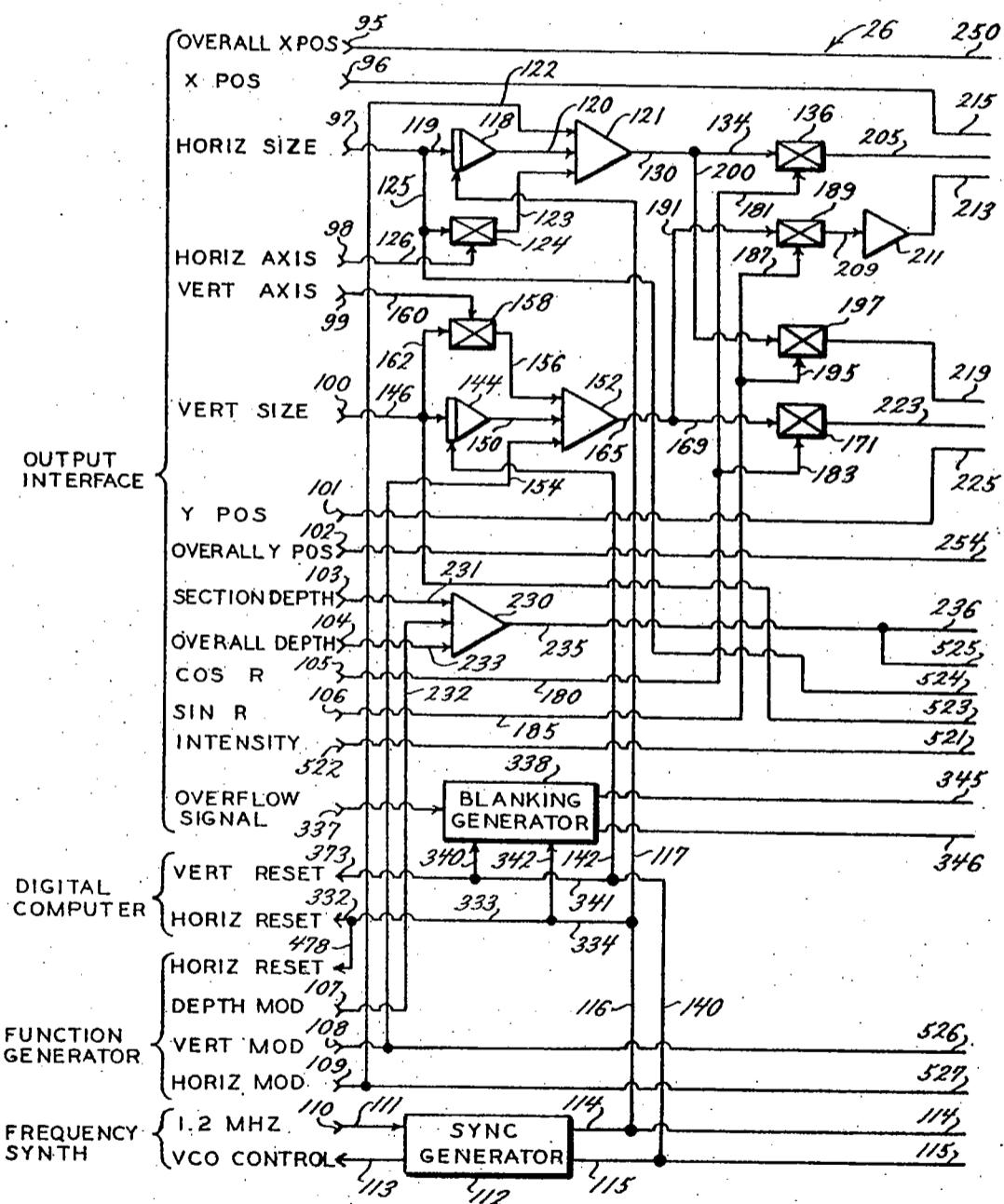


FIG. 4.

INVENTORS  
LEE HARRISON III  
FRANCIS J. HONEY  
EDWIN J. TAJCHMAN  
BY MARSHALL M. PARKER  
*Rogers, Gell, Cilera & Robbins*  
THEIR ATTORNEYS

PATENTED JUL 17 1973

3,747,087

SHEET 4 OF 7

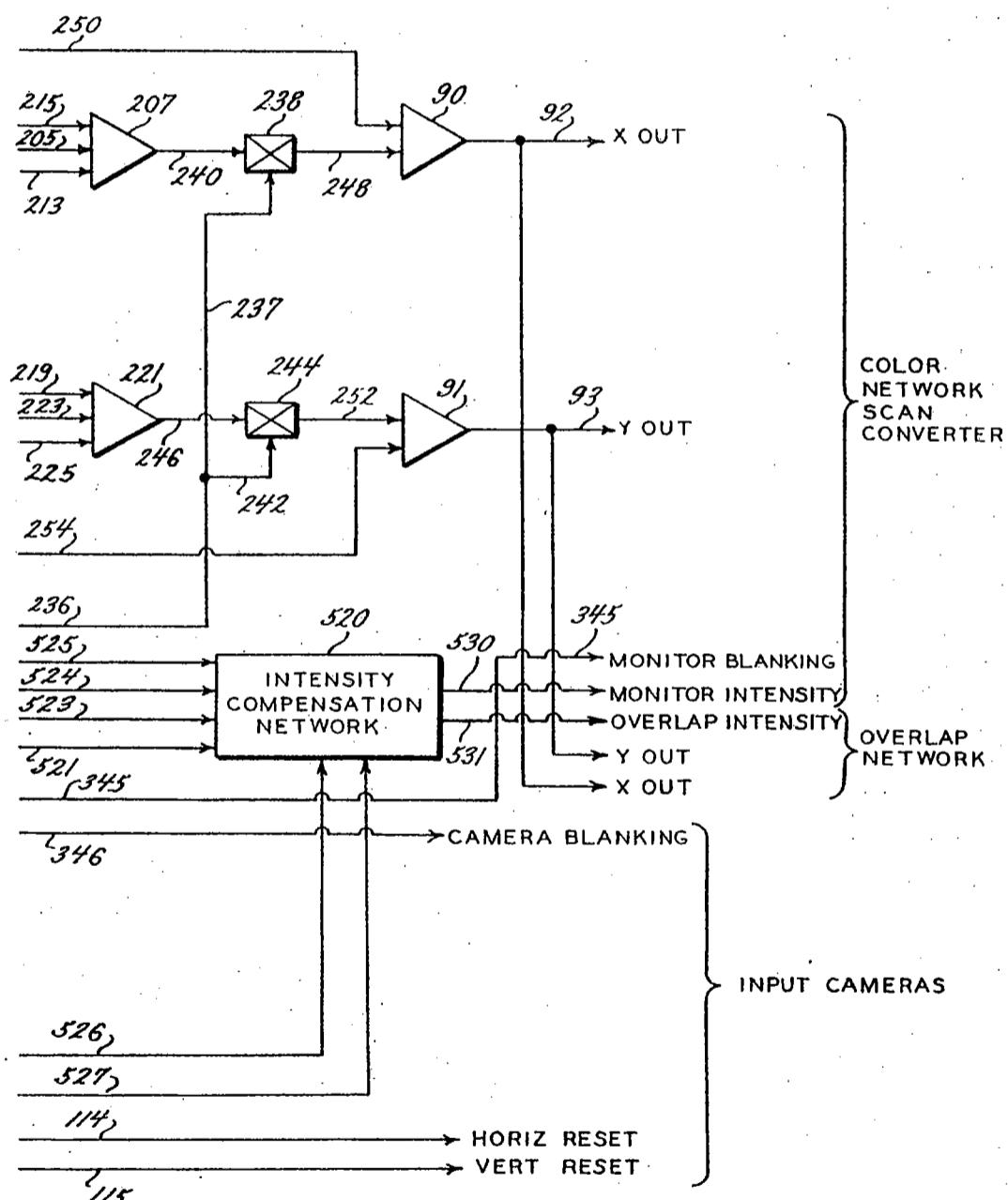


FIG. 4A.

INVENTORS

LEE HARRISON III

FRANCIS J. HONEY

EDWIN J. TAICHMAN

MARSHALL M. PARKER

BY Roger, Egel, Eiles & Robbins

THEIR ATTORNEYS

PATENTED JUL 17 1973

3,747,087

SHEET 5 OF 7

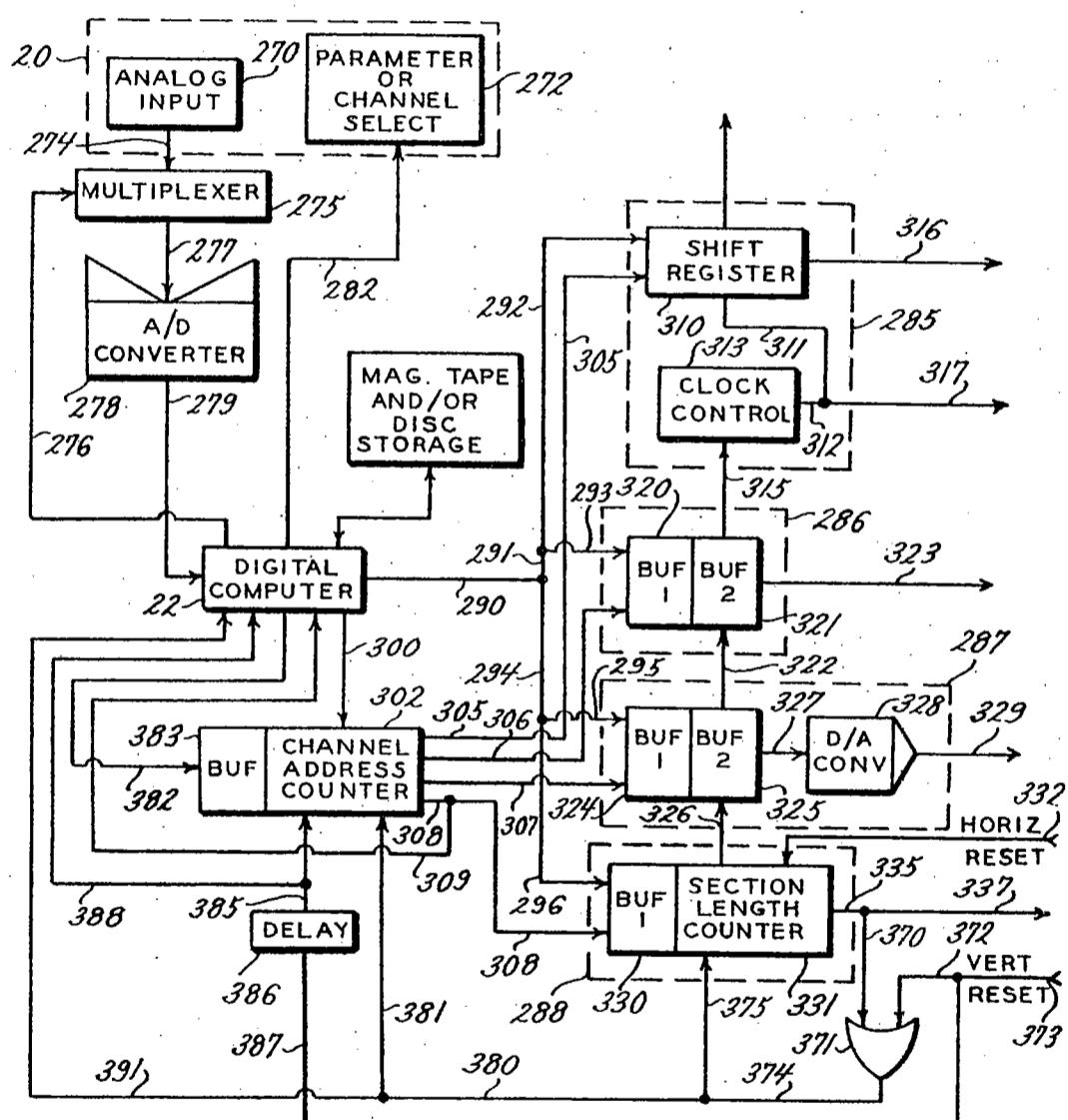


FIG. 5.

INVENTORS

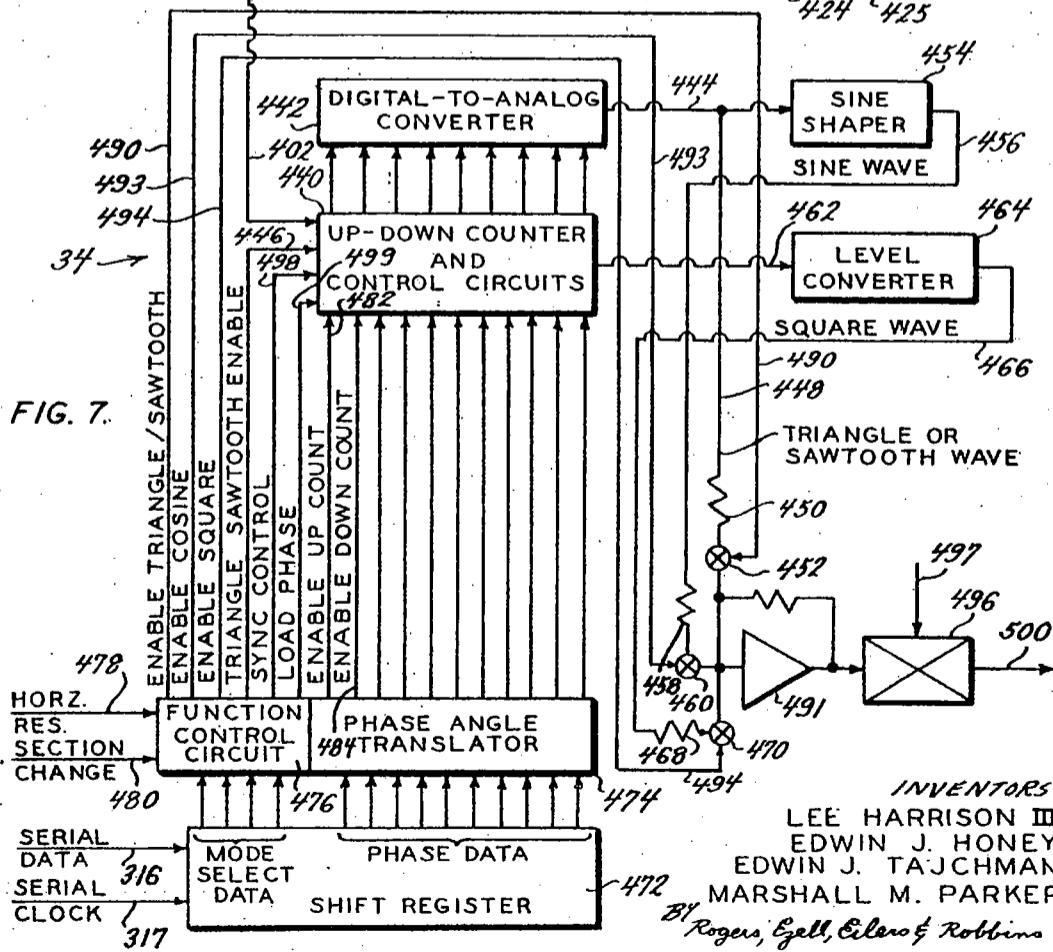
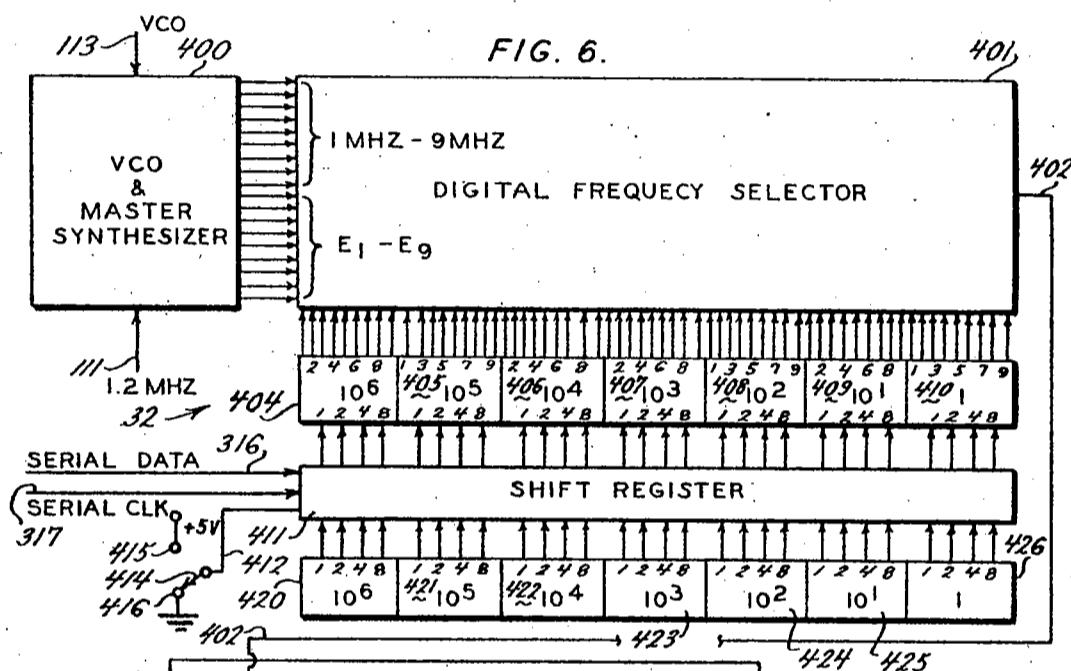
LEE HARRISON III  
FRANCIS J. HONEY  
EDWIN J. TAJCHMAN  
MARSHALL M. PARKER  
BY Roger, Egell, Eilera & Robbins

THEIR ATTORNEYS

PATENTED JUL 17 1973

3,747,087

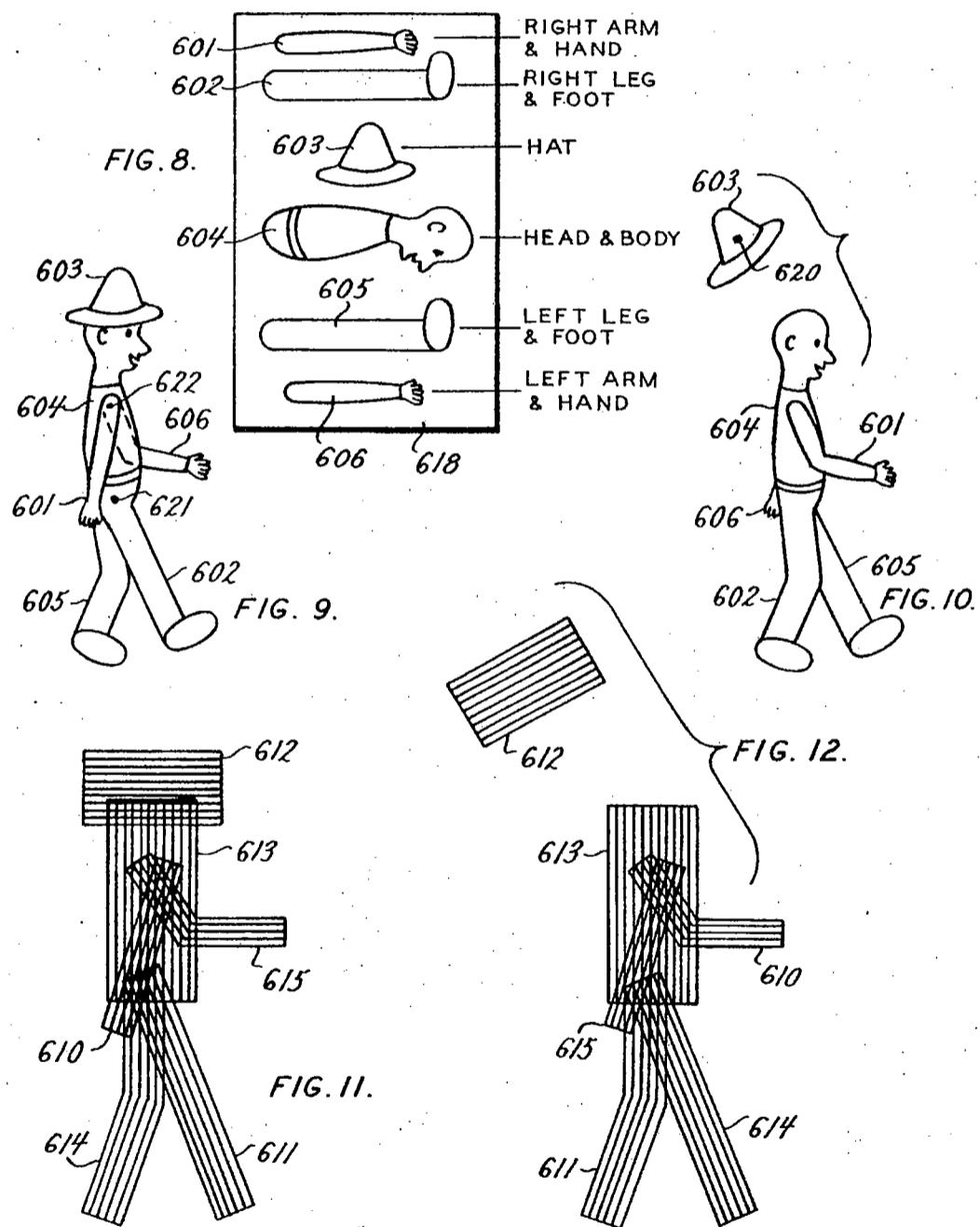
SHEET 6 OF 7



PATENTED JULY 17 1973

3,747,087

SHEET 7 OF 7



INVENTORS  
LEE HARRISON III  
FRANCIS J. HONEY  
EDWIN J. TAJCHMAN  
BY MARSHALL M. PARKER  
*Roger, Zell, Eilers & Robbins*  
THEIR ATTORNEYS

### DIGITALLY CONTROLLED COMPUTER ANIMATION GENERATING SYSTEM

#### CROSS REFERENCES TO COPENDING RELATED APPLICATIONS

This application relates to the disclosures of application Ser. No. 95,096 filed Dec. 4, 1970, now U.S. Pat. No. 3,710,011, and application Ser. No. 72,642, filed Sept. 16, 1970, now U.S. Pat. No. 3,689,917.

#### SUMMARY OF THE INVENTION

The invention of this application produces animation sequences of scenes in a way different from any other system, to create a new and novel computer animation system that provides greatly increased figure animation capability.

The system basically includes an analog portion for generating output signals representing one or more sections of a raster on which images viewed by a video camera can be produced. Analog inputs to the analog portion define the structure, size, shape, location, orientation and other parameters of the raster section to effectively define the shape of each part of the viewed image produced thereon. By varying the analog inputs, the raster section parameters can be made to vary, thus imparting motion to the image. In this system the values and variations in values of the analog inputs are controlled by digital signals from a digital computer which establishes these digital control signals from information fed to it from a director.

More specifically, the analog computer portion generates X and Y coordinate signals representing each section of an animated image for each frame of an animated sequence, each section of the image comprising a raster section. Sweep generators within the analog portion generate the basic horizontal and vertical sweep signals which are combined with, or modulated by, other input signals to define the structure, shape, size, position and other parameters of each raster section for each sequence frame. The X and Y coordinate signals at the output of the analog portion can be applied to a cathode ray tube to produce a display of the animated sequence or can be used to record the sequence on video tape or film.

The input signals to the analog portion define, for each raster section, parameters such as X and Y position, X and Y size, horizontal and vertical axes of rotation which determine the radius of rotation of each raster section, section depth, cosine and sine of the angle of rotation, intensity, and horizontal, vertical and depth modulation. Other parameter inputs define overall X and Y position and depth for the entire image. All of these parameter input signals act in conjunction with the sweep generators to ultimately produce the X and Y coordinate output signals from the analog portion for use in producing an animation sequence. By varying these input signals, each raster section can be made to vary in height or width, move anywhere in relationship to any other raster sections, rotate about any point located inside or outside the raster section, and can be modulated with the variety of modulation signals to produce bending or distortion of the raster section, which bending or distortion can be made to vary by varying modulation signal parameters such as frequency, phase and amplitude.

The parameter signals of the analog portion and the modulation signal parameters are established for each

section of the image and for each frame of the sequence by a digital computer portion which automatically calculates these parameters from information it receives at its input from a director. Parameter data for each raster section for initial and final frames of a sequence of a selected number of frames are selected on the director, which information is fed to the digital computer. The digital computer is programmed to automatically calculate, upon command, the parameter data for each section and for each frame between initial and final frames in accordance with a selected fairing function and to store this information in a digital memory such as magnetic tape or disc. Depending on the fairing function selected, these digital computations may be linear or based on some other mathematical function to define the patterns of parameter change throughout the sequence. For example, in this way an arm of a figure can be made to move at a constant rate from a first to a second position, or at a varying rate depending on the fairing function selected. The digital information recorded on the magnetic tape or disc can then be played back through the digital computer to the analog portion to produce the animated sequence.

Where a figure is displayed on the animated raster sections a video camera is trained to scan the individual parts of the figure to be animated. Timing pulses are generated to time the scan of the video camera in synchronization with the production of the X and Y coordinate output signals from the analog portion to reproduce each part of the scanned figure on a raster section as defined by the X and Y coordinate output signals. The result is that the video signals from the video camera determine the detail surface characteristics and shape of each part of the image as displayed on a raster section, while the analog portion of the computer defines the structure, shape, size, position and other parameters of each raster section as determined by the values of its parameter inputs. Digital signals from the digital computer vary the parameter inputs in a controlled manner between initial and final frames to animate the raster sections and hence the parts of the figure produced thereon to create the animation sequence.

With this system the editing is actually done by the digital computer and not after the scenes are recorded on video tape or film. By dividing a sequence to be filmed into segments or scenes having initial and final frames, the information for creating each frame of the entire sequence can be recorded on digital magnetic tape as the scenes are produced, which information can then be played back through the system for ultimate display or recording of the animation sequence on film or video tape. In this way, a sequence of any length can be produced without the need to edit the film or video tape.

Means are also provided for adding background information to the sequence and for varying the background and image foreground information by switching artwork video inputs during the sequence. The system further includes means for producing the entire sequence in colors that can be selectively varied.

From the foregoing, it is apparent that this system allows an artist to produce animated sequences in far less time than would be possible using conventional techniques (requiring 24 separate drawings for each second of animation) and yet provides him with wide control for producing many different types of animation. The

result is that the artist animates with the computer by operating the director controls, giving his full attention to creativity and results rather than the tedium of repetition.

#### DESCRIPTION OF THE DRAWINGS

FIG. 1 is a general block diagram of the system of this invention for generating an animation sequence;

FIG. 2 is a block diagram of a network for producing a color display or video tape of an animation sequence generated by the network of FIG. 1;

FIG. 3 is a block diagram of a network for producing a color recording on film of an animation sequence generated by the network of FIG. 1;

FIG. 4 and 4A combined are a schematic drawing of the analog computer portion of the system;

FIG. 5 is a block diagram of the control network of the system;

FIG. 6 is a block diagram of the frequency synthesizer of the system;

FIG. 7 is a block diagram of the function generator of the system; and

FIGS. 8, 9, 10, 11 and 12 are illustrations used in explaining the operation of the system.

#### DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Before describing the system components in detail, the system will be generally described by referring to the block diagrams of FIGS. 1, 2 and 3. An input interface unit 20, also called a director, generates analog and digital signals defining each parameter of each raster section for a given frame of the sequence to be generated. The values of these signals are set with analog controls, such as for example, potentiometers, and digital parameter or channel select controls, such as for example, multiposition switches, which determine the parameter or parameters being defined at a given instance by the analog control signals. The analog signals from the director 20 are converted to digital signals and fed to a digital computer 22, such as a Honeywell 316 or equivalent, which is programmed to store on appropriate command the digital information relating to each parameter of the given frame and to feed this information in a prescribed manner to an interface network 24 containing a plurality of interface units. Some of these units convert the digital parameter data from the digital computer 22 to analog parameter signals and transfer these signals to parameter inputs of an analog computer portion 26. Digital data from other interface units of the interface network 24 is fed to a frequency synthesizer 32 to define the frequencies of modulation signals used for producing certain animation effects. Digital data from still other interface units of the interface network 24 is fed to a function generator 34 for defining the phase, wave form, amplitude, and synchronization mode of the modulation signals, which signals are fed from the output of the function generator 34 to modulation parameter inputs of the analog computer portion 26. From these inputs and those from the interface units of the interface network 24, the analog computer portion 26 generates signals at its output defining raster sections. In this way such parameters as the structure, shape, size and position of each raster section generated at the output of the analog portion 26 are defined by the control settings of the director 20.

The digital computer 22 is also programmed upon receipt of parameter information from the director 20 defining initial and final frames of an animation sequence of a given number of frames to automatically calculate the parameter information defining each frame therebetween in accordance with an appropriate fairing function selector on the director, which function defines the pattern of variation for each parameter throughout the sequence, and to record the digital parameter information defining the entire sequence on a digital recording device such as a magnetic tape unit 36 or magnetic disc unit 38. Upon appropriate command from the director 20, the digital information stored on the digital tape 36 or disc 38 is played back through the digital computer 22, interface network 24, and analog computer portion 26, to produce signals at the output of the analog portion 26 representing the entire animation sequence.

The system also includes a plurality of video cameras such as 40, 41 and 42, each of which can be made to scan an art-work subject for display on the raster sections generated by the analog portion 26. The artwork subjects could be from most any source such as a slide projector 43, or light boxes 44 and 45, on which the artwork is mounted. The video information from each of the cameras 40 through 42 is sent to a switcher unit 52 which is controlled by signals from the outputs of still other interface units of the interface network 24 to switch the video of a selected one of the cameras 40 through 42 to the input of a gray level encoder 54. The encoder 54 is fully described in copending U.S. patent application Ser. No. 95,096, filed Dec. 4, 1970 now Pat. No. 3,710,011, entitled A System For Automatically Producing A Color Display Of A Scene From A Black And White Representation Of The Scene. For the purposes here, it is sufficient to know that the encoder 54 produces video signals at its output representing the artwork in discrete shades of gray, which signals are fed to an overlap network 56 together with the output signals from the analog portion 26. The overlap network 56 may be of the type disclosed in FIG. 8 of U.S. Pat. No. 3,364,382 for blanking portions of the generated image appearing behind other portions of the image. To accomplish this, an output signal from the overlap network 56 is fed to a video gate 58 to gate the video signal from the output of the gray level encoder 54, through the gate 58, to an output conductor 60 only when image information is generated over an area not previously covered with other image information. Hence, the output conductor 60 carries a video signal representing the artwork in discrete shades of gray and which has been compensated for overlap prevention. This video signal can be used to reproduce the various parts of the artwork on the raster sections generated by the analog computer portion 26.

The analog computer portion 26 generates horizontal and vertical reset pulses from signals received by it from the frequency synthesizer 32. These horizontal and vertical reset pulses are fed from the analog computer portion 26 to the digital computer portion 22, as shown by conductors 62, and to a camera control network 64, as shown by conductors 66, to synchronize the generation of the coordinate output signals from the analog portion 26 defining the raster sections with the generation of the video signals on the conductor 60 representing the artwork.

The coordinate output signals from the analog portion 26 and video signals on the conductor 60 may be typically used to produce either a black and white or color display, video tape, or film of the animation sequence with the networks of FIGS. 2 and 3. Referring to FIG. 2, the output signals from the analog portion 26 and video signals on the conductor 60 are fed into a color network 67, together with signals from still other interface units of the interface network 24 defining the red, blue and green color components for each discrete gray shade. The color network 67 may be of any one of the embodiments fully disclosed in the above-referenced application Ser. No. 95,096 for assigning colors to each gray shade. The output from the network 67 may be used directly to produce a color display or video tape of the animation sequence, or fed to a video effects unit 68 and combined with signals representing background information in TV format, such as from a color video camera 69 photographing background artwork 70, a conventional film chain 71, or a video tape recorder 72, any one of which may be used to supply background information for the sequence.

The output from the video effects network 68 is fed to a color monitor 73 for producing a color display of the animation sequence, directly to a video tape recorder 74 to produce a video tape of the sequence, or to a microwave link 75 for transmission to a video tape recorder 76.

In FIG. 3, there is shown a network for producing a color film of the animation sequence. The video signal on the conductor 60 and the output signals from the interface units of the interface network 24 defining the red, blue and green color components of the color to be assigned to each gray shade, are fed to an RGB color color encoder 77, such as the type shown in FIG. 6 of the above-referenced application Ser. No. 95,096. The encoder 77 generates a video signal at an output 78 representing the red color component of each assigned color which is fed to an X-Y monitor 79, a video signal at an output 80 representing the blue color component of each assigned color which is fed to an X-Y monitor 81, and a video signal at an output 82 representing the green color component of each assigned color which is fed to an X-Y monitor 83. The coordinate output signals from the analog portion 26 are also fed to each of the X-Y monitors 79, 81 and 83, the X-Y monitor 79 producing a black and white display defining the red color components of the sequence filmed by a camera 84 on red encoded black and white film, X-Y monitor 81 producing a black and white display defining the blue color components of the sequence filmed by a camera 85 on blue encoded black and white film, and the X-Y monitor 83 producing a black and white display defining the green color components of the sequence filmed by a camera 86 on green encoded black and white film. The red, blue and green encoded films are then processed in the laboratory by known techniques to produce a composite color film of the animation sequence.

As a variation of this technique, the monitors 79, 81 and 83 could be equipped with high intensity tubes with red, blue and green phosphors, respectively, and the images produced thereon optically combined and photographed on color film.

#### ANALOG COMPUTER PORTION

The analog computer portion 26 will be described in

more detail by referring to the network of FIGS. 4 and 4A. The analog portion 26 includes summation amplifiers 90 and 91. Output conductors 92 and 93, respectively, from the summation amplifiers 90 and 91 carry X and Y coordinate signals for generating a raster section the parameters of which are defined by input signals to the analog portion 26. By changing the values of these input signals, the values of the X and Y coordinate signals at the outputs 92 and 93 are changed, thereby changing the parameters of the raster section generated thereby.

Remembering that the coordinate output signals 92 and 93 produce a raster section the parameters of which are defined by the values of the input signals to the analog portion 26, it can best be understood how these coordinate signals are generated and what these raster parameters are by beginning at the inputs of the analog portion 26.

The analog portion 26 has a plurality of inputs each of which has an analog parameter signal thereon for defining the shape, size, position and structure for a given raster section represented by the X and Y coordinate outputs 92 and 93. The signal at an input 95 defines the overall X position of the entire image comprised of a selected number of raster sections; the signal at an input 96 defines the X position of a given section of the image relative to the other sections of the image; the signal at an input 97 defines the horizontal size or length of raster lines of the given raster section; the signal at an input 98 defines the horizontal axis of rotation of the given raster section; the signal at an input 99 defines the vertical axis of rotation of the given raster section; the signal at an input 100 defines the vertical size or the spacing between raster lines of the given raster section; the signal at an input 101 defines the Y position of the given raster section relative to the other sections of an image; the signal at an input 102 defines the overall Y position of the entire image; the signal at an input 103 defines the depth or overall size of the given raster section; the signal at an input 104 defines the overall depth or overall size of the entire image; signals at inputs 105 and 106 define the cosine and sine, respectively, of an angle R, where R is the angle of rotation of the given raster section with respect to a reference axis; and inputs 107, 108 and 109 carry modulation signals from the function generator 34 to be described, for producing depth, vertical and horizontal modulations of the given raster section. The inputs 95 through 106 are from interface units of the interface network 24, and the inputs 107 through 109 from the function generator 34.

A 1.2 megahertz signal generated by the frequency synthesizer 32, to be described, is fed to an input 110 and through a conductor 111 to a sync generator 112. From the 1.2 megahertz signal, the sync generator 112 produces a 60 hertz signal for comparison with the 60 hertz line frequency and creates therefrom a voltage control oscillator (VCO) control signal at an output 113 which is fed back to the frequency synthesizer 32. The VCO signal adjusts the frequency of the 1.2 megahertz signal to insure that the 60 hertz signal generated by the sync generator 112 is locked to the line frequency. The generator 112 generates at one of its outputs 114 a horizontal reset signal of a frequency of 28.35 kilohertz, and at another output 115 a vertical reset signal of a frequency of 60 hertz. As will be seen, these horizontal and vertical reset signals are used

throughout the system to synchronize various timing operations.

The horizontal reset signal on the conductor 114 is fed through a conductor 116 and a conductor 117 to the reset input of an integrator 118. Another input conductor 119 to the integrator 118 is connected from the input 97 defining the horizontal size of a particular raster section. Therefore, after each horizontal reset pulse the integrator 118 operates as a horizontal ramp generator to draw a line of the raster section of a length determined by the magnitude of the signal at the input 97. The output of the integrator 118 is fed through a conductor 120 to one input of a summation amplifier 121. The summation amplifier 121 has another input conductor 122 connected to the input 109 carrying the horizontal modulation signal from the function generator 34. Another input conductor 123 to the summation amplifier 121 is connected from the output of a multiplier 124 having one input connected by a conductor 125 to the horizontal size input 97, and having another input conductor 126 connected to the input 98 defining the horizontal axis of rotation of the section. Therefore, the purpose of the multiplier 124 is to insure that where there is an increase in the horizontal size of the raster section, there is automatically a corresponding change in the horizontal axis of rotation of that section to insure that the section rotates about the same relative point. The output of the summation amplifier 121, representing the sum of the horizontal ramp function, the horizontal modulation signal, and the product of the signals representing the horizontal axis of rotation and the horizontal size of the raster section is fed through a conductor 130 and a conductor 134 to one input of a multiplier 136.

The vertical reset signal at the output 115 from the sync generator 112 is fed through a conductor 140 and a conductor 142 to the reset input of an integrator 144. The integrator 144 has an input conductor 146 connected from the analog computer input 100 defining the vertical size of each raster section so that each time after the integrator 144 receives a vertical reset pulse, it generates a vertical ramp function defining the height of the raster section (the distance between the raster lines) in accordance with the magnitude of the signal at the input 100. The output of the integrator 144 is fed through a conductor 150 to one input of a summation amplifier 152. Another input conductor 154 to the summation amplifier 152 is connected from the input 108 to the analog portion carrying the vertical modulation signal from the function generator 34. Another input conductor 156 to the summation amplifier 152 is connected from the output of a multiplier 158 having one input connected by a conductor 160 to the input 99 of the analog portion carrying the signal defining the vertical axis of rotation of the raster section, and another input conductor 162 connected from the vertical size input 100. The multiplier 158 performs the same function with respect to compensating for changes in vertical size as the multiplier 124 does for compensating for changes in horizontal size. That is, where there is a change in the vertical size of the raster section, the multiplier 158 insures that there is automatically a corresponding change in the vertical axis of rotation so that the point of rotation remains in the same relative position. The output from the summation amplifier 152 representing the sum of the vertical ramp signal, the vertical modulation signal, and the product of the sig-

nals defining the vertical axis of rotation and vertical size of the raster section is fed through a conductor 165 and a conductor 169 to one input of a multiplier 171.

To provide a wide variety of animation capability, this system includes means for rotating each raster section about a point defined by the horizontal and vertical axes inputs 98 and 99 and through an angle, which will be called R, the cosine and sine of which are defined by voltages at the inputs 105 and 106, respectively, to the analog portion 26. The system further includes means for rotating any raster distortion with the raster. To illustrate the importance of these rotation capabilities, consider the problem of producing an animation sequence of a walking figure. As the figure moves, the legs must bend at the knees and rotate at the hips, and the arms must bend at the elbows and rotate at the shoulders. Taking one of the arms as an example, as the raster section on which the arm is produced rotates causing rotation of the arm about the shoulder, the raster distortion pattern creating the bend at the elbow, which bend is produced by a selected vertical modulation signal from the function generator 34, must also rotate or else an unnatural or undesirable distortion of the arm will result. For this reason, rotation is imparted to the raster section after the modulation signals are summed with the horizontal and vertical ramp signals. Hence, the signal at the input 105 representing the cosine of R is fed through a conductor 180 and a conductor 181 to a second input of the multiplier 136, and through a conductor 183 to a second input of the multiplier 171. The signal at the input 106 representing the sine of R is fed through a conductor 185 and a conductor 187 to one input of a multiplier 189, the other input of which is connected by conductor 191 to the output 35 of the summation amplifier 152. The signal on the conductor 185 representing the sine of R is also fed through a conductor 195 to one input of a multiplier 197 having another input connected by a conductor 200 to the output of the summation amplifier 121. The 40 output from the multiplier 136 representing the product of the output from the summation amplifier 121 and the cosine of R is fed through a conductor 205 to one input of a summation amplifier 207. The output from the multiplier 189 representing the product of the output from the summation amplifier 152 and the sine of R is fed through a conductor 209, an inverter 211 and a conductor 213 to a second input of the summation amplifier 207. The summation amplifier 207 has a third input conductor 215 connected from the input 96 45 of the analog portion 26 carrying the signal defining the X position of the section. The output from the multiplier 197 representing the product of the output from the summation amplifier 121 and the sine of R is fed through a conductor 219 to one input of a summation amplifier 221. The output from the multiplier 171 representing the product of the output from the summation amplifier 152 and the cosine of R is fed through a conductor 223 to a second input of the summation amplifier 221. The summation amplifier 221 has a third input conductor 225 connected from the input 101 of the analog portion 26 carrying a signal defining the Y position of the raster section.

A summation amplifier 230 has one input connected by a conductor 231 to the input 103 of the analog portion carrying the signal defining the depth of the raster section, another input connected by a conductor 232 to the input 107 of the analog portion carrying the

depth modulation signal from the function generator 34, and another input connected by a conductor 233 to the input 104 of the analog portion carrying the signal defining the overall depth of the entire image. The output from the summation amplifier 230 representing the sum of these depth signals is fed through a conductor 235, a conductor 236, and a conductor 237 to one input of a multiplier 238. The output from the summation amplifier 207 is fed through a conductor 240 to a second input of the multiplier 238. The output from the summation amplifier 230 is also fed through the conductors 235 and 236 and through a conductor 242 to one input of a multiplier 244, the other input of which is connected by a conductor 246 to the output of the summation amplifier 221. The output from the multiplier 238 representing the product of the signals from the summation amplifiers 207 and 230 is fed through a conductor 248 to one input of the summation amplifier 90. Another input of the summation amplifier 90 is connected by a conductor 250 to the input 95 of the analog portion 26 carrying the signal defining the overall X position of the image. The output from the multiplier 244 representing the product of the signals from the summation amplifiers 221 and 230 is fed through a conductor 252 to one input of the summation amplifier 91. Another input of the summation amplifier 91 is connected by a conductor 254 to the input 102 of the analog portion 26 carrying the signal representing the overall Y position of the image. Hence, the output from the summation amplifier 90 represents the X coordinate signal for the raster section, and the output of the summation amplifier 91 represents the Y coordinate signal for the raster section.

From the above description, it is apparent that the analog portion 26, given a set of fixed input parameter signals, generates a raster section of a particular size, shape, structure, and position as defined by these input signals. If the input parameter signals remain constant over time, identical raster sections will be generated repeatedly. If, on the other hand, the input parameter signals are changed in a prescribed manner over a prescribed time interval, a plurality of raster sections, each shaped, sized, structured and positioned differently from the others, are generated. Furthermore, by changing the input parameter signals defining each raster section at prescribed increments and at a prescribed frequency, motion can be imparted to the rasters and hence the image, to produce an animation sequence. The control of the input parameter signals to the analog portion 26 to produce and animate the different raster sections is accomplished with the control network of FIG. 5.

#### CONTROL NETWORK.

In generating an animation sequence, the control network operates in basically three modes: a frame reference mode, a record mode, and a playback mode.

In the frame reference mode the digital computer 22 receives digital-parameter information from the director 20 for establishing initial and final frames for the sequence. The director 20 has an analog input portion 270 (which might include, for example, a plurality of potentiometers or the like) for generating analog signals which might be used to define any raster parameter, and a parameter select input 272 (which might include multiposition switches) for selecting the parameter to be defined by the analog signal from the analog

input 270. The analog input signals from the analog input 270 selected for particular ones of these parameters are fed through suitable conductors 274 to one input of a multiplexer 275 having another input connected by conductor 276 to an output from the digital computer 22 which carries signals in accordance with its program to gate the analog information from the analog input 270 in a prescribed sequence through conductors 277 to the input of an analog to digital converter 278. The digital output signals from the converter 278 are fed through conductors 279 to an input of the digital computer 22.

The computer 22 is programmed to store the information received from the analog input 270 and to interrogate through appropriate conductors 282 the parameter select unit 272 to determine the parameter to which each piece of stored information pertains. Having made this determination, the parameter information is transferred to an appropriate storage location 15 within the computer 22. By manipulating the analog input 270 and parameter select 272 controls, digital signals are eventually stored in the appropriate memory locations of the digital computer 22 representing each parameter of each raster section of the image for the 20 initial frame of the sequence. Having stored the initial frame information, the same procedure is followed for establishing the digital parameter information for the 25 last frame of the sequence.

Having established the initial and final frame parameters for the sequence the digital computer 22 can now be placed in the record mode. In this mode the digital computer 22 is programmed to calculate, upon command from the director 20, the digital parameter information defining each raster section of each frame between initial and final frames, and to record this information on the digital recording device such as magnetic tape 36 or disc 38. These calculations are made in accordance with a selected one of several mathematical functions called fairing functions which define the rate 30 of change of each parameter from frame to frame throughout the sequence. For example, if a linear fairing function is selected, the rate of parameter change from frame to frame is constant, while other fairing functions produce varying change rates. The computer 22 is programmed to make its calculations in accordance with any one of the several fairing functions, a particular function selected by command signal from the director 20.

With the digital parameter information defining each raster section of each frame of the sequence recorded on the digital recording device, the control network 35 can be placed in its playback mode to play the recorded digital parameter information back to the digital computer 22 where it is stored in a prescribed sequence in various memory locations.

In this manner digital parameter information is fed to and stored in the digital computer 22 for transfer to other networks of the system such as the analog portion 26, frequency synthesizer 32, and function generator 34, to generate the sequence. The manner in which this transfer is accomplished is the time regardless of its source (whether from the director 20 or digital magnetic recording devices 36 or 38) and will not be explained.

As parameter information from one of the input sources is stored at some memory location within the computer 22, the computer 22 is programmed to trans-

mit this information sequentially at high speed, and in a specified order to interface units within the interface network 24. In this described embodiment of the invention there are basically four types 285, 286, 287 and 288 of these interface units. There may be any number of each type depending on the number of parameters to be defined. For example, in this embodiment there are nine units of the type 285, two units of the type 286, 31 units of the type 287, and one unit of the type 288 for a total of 43 interface units, each to affect a different parameter as each raster section of an image frame is drawn. There are, therefore, at least as many interface units as there are parameters to be defined for each raster section generated.

The information stored in the digital computer 22 defining each parameter is fed in an order determined by the digital computer program to each of the interface units. Hence, for example, the information defining a first parameter of a given raster section is fed through conductors 290, 291 and 292 to an interface unit of the type 285; conductor 290, 291 and 293 to an interface unit of the type 286; conductors 290, 294 and 295 to an interface unit of the type 287; and conductors 290, 294 and 296 to an interface unit of the type 288; and so on for the information defining the second and succeeding parameters until all of the parameter information defining a particular raster section of the frame is fed to the inputs of each interface unit.

As the information defining the first parameter is fed to each interface unit at high speed, a signal is sent from the digital computer 22 through a conductor 300 to a channel address counter 302 which has a separate output connected to a gate input of each interface unit. To illustrate, there is an output conductor 305 connected to a gate input of the interface unit of the type 285, an output conductor 306 connected to a gate input of the interface unit of the type 286, an output conductor 307 connected to a gate input of the interface unit of the type 287, and an output conductor 308 connected to a gate input of the interface unit of the type 288.

In this embodiment with 43 interface units, the counter 302 would have 43 such output conductors each connected to a gate input of an interface unit. Each time the digital computer 22 feeds information defining a parameter for a particular raster section to each interface unit it sends a signal through the conductor 300 to activate an appropriate one of the outputs of the counter 302 which in turn gates the information which is present at the inputs of all the interface units to the appropriate one of these units. The digital computer 22 is programmed to feed the digital parameter information in the proper sequence so it is gated to the appropriate interface unit to effect the appropriate parameter input to the system.

When the last output conductor, which for purposes of illustration, might be the conductor 308, of the counter 302 is activated to gate digital parameter information to the last interface unit, all the units are loaded and the high speed transfer of information from the digital computer 22 relating to the next raster section to be generated must be stopped. Hence, the gating signal on the last output conductor 308 of the counter 302 is also fed through a conductor 309 to an input of the digital computer 22 to stop the high speed transfer. As will be seen, the high speed transfer of information from the digital computer 22 will be started again to reload the

interface units with information by the start of a vertical reset or a section change pulse defining the next raster section.

Each of the types of interface units 285 through 288 is designed differently to perform a different interface function and for interface with a different part of the system. The unit type 285 is used for interface with the frequency synthesizer 32 and function generator 34 for defining the frequency, phase, waveform, and synchronization mode parameters of the modulation oscillators for use in creating vertical, horizontal and depth modulation. In this embodiment, two such units are required to define frequency, and one for phase, waveform and synchronization mode. With vertical, horizontal, and depth modulation, nine such units are used. Each unit type 285 includes a shift register 310 having one input, connected by conductors 290, 291, and 292 to the output of the digital computer 22 for receiving the digital parameter information, an input connected by the conductor 305 to an output of the channel address counter 302 for gating the appropriate digital parameter information into the shift register 310, and an input connected by a conductor 311 and a conductor 312 to the output of a clock control 313. The clock control 313 has an input 315 connected to the output of an OR gate to be described. When the clock control 313 receives a signal at its input 315, it transmits a series of pulses through the conductors 312 and 311 to the input of the shift register 310 to feed the digital parameter information in the shift register 310 serially through an output conductor 316 to the frequency synthesizer 32 or function generator 34. The clock pulses from the output of the clock control 313 are also fed through a conductor 317 to the frequency synthesizer 32 or function generator 34.

Each interface unit of the type 286 includes a first buffer 320 and a second buffer 321, the input of which is connected to the output of the buffer 320. The buffer 320 has one input connected by the conductors 290, 291 and 293 to the output of the digital computer 22 for receiving the digital parameter information, and an input connected by the conductor 306 to an output of the channel address counter 302 for gating the appropriate digital parameter information into the first buffer 320. The second buffer 321 has an input 322 which, like the input 315 of the clock control 313, is connected to the output of the OR gate. On appropriate signal at the input 322, the digital parameter information in the first buffer 320 is transferred to the second buffer 321 and through an output conductor 323 to provide digital control signals available for such operations as section blanking and video switching. In this embodiment, two such unit types 286 are used, although, of course, the number depends on the number of digital control signals required for various controlling operations.

An interface unit of the type 287 is used to interface with each of the parameter inputs 95 through 106 of the analog portion 26, and the function generator 34 to define the amplitudes of the modulation signals. Since there are vertical, horizontal, and depth modulations, three such units are required for this purpose. Units of this type are also used to interface with the color network 67 or RGB color encoder 77 of FIGS. 2 and 3, for defining the colors of each frame of the image. For example, where each frame has five different colors (five discrete shades of gray, a color for each shade), and

since each color has a red, blue and green color component, 15 such units are required. A unit of this type is also used to interface with an intensity compensation network, to be described.

Each interface unit of the type 287 includes a first buffer 324 and a second buffer 325, the input of which is connected to the output of the first buffer 324. The first buffer 324 has an input connected by the conductors 290, 294 and 295 to the output of the digital computer 22 for receiving the digital parameter information. The buffer 324 also has an input connected by the conductor 307 to an output of the channel address counter 302 to gate the appropriate digital parameter information into the buffer 324. For example, this information might define one of the color parameters, or an amplitude parameter for a modulation oscillator, or one of the input parameters 95 through 106 to the analog portion 26. The buffer 325 has an input 326 which, like the inputs 315 and 322 to the units 285 and 286, is connected to the output of the OR gate. Upon receiving an appropriate signal at the input 326, the digital information in the buffer 324 is transferred to the buffer 325 and fed through the conductors 327 to the input of a digital to analog converter 328 which converts the digital information to an analog signal for transmission through its output conductor 329 to the appropriate parameter input of the system.

The interface unit type 288 has the dual function of defining the number of raster lines in each raster section and timing the transfer of parameter information from the digital computer 22 to the parameter inputs of the analog portion 26 and other parameter inputs of the system. In this described embodiment, there is only one interface unit of the type 288 required in the system. It includes a buffer 330 and a section length counter 331, an input of which is connected to the output of the buffer 330. The buffer 330 has one input connected by conductors 290, 294 and 296 to the output of the digital computer for receiving the digital parameter information, and an input connected by the conductor 308 to the appropriate output of the channel address counter 302 for gating into the buffer 330 the digital parameter information from the digital computer 22 which defines the number of lines in the section being generated. The section length counter 331 has an input conductor 332 which is connected by conductors 333, 334, 116 and 114 to the horizontal reset output of the sync generator 112 (FIG. 4). The section length counter 331 continuously counts the horizontal reset pulses on its input conductor 332 and upon reaching a prescribed count, generates an overflow or section change signal at an output conductor 335. How long the counter 331 must count to generate the overflow signal depends on the digital information fed to it from its input buffer 330. The length of the count by the counter 331 defines the number of lines in the section being generated.

When the counter 331 stops counting and generates the section change signal at its output, this signal is fed through the conductor 335 and a conductor 337 to an input of a blanking generator 338 (FIG. 4). The blanking generator 338 has another input connected by a conductor 340, a conductor 341 and the conductors 140 and 115 to the vertical reset output of the sync generator 112, and another input connected by a conductor 342 and in the conductors 334, 116 and 114 to the horizontal reset output of the sync generator 112.

Upon receiving a section change signal from the counter 331, the blanking generator 338 generates a signal at its output which is fed through a conductor 345 for use in blanking the beam of a monitor used for displaying the image or a scan converter used in producing the image in TV format. The blanking generator 338 is set to generate the blanking signal for a period of time equivalent to approximately two horizontal raster lines, although this period is adjustable. The blanking generator 338 generates another signal at an output which is fed through a conductor 346 for use in blanking the beam of the artwork scanning camera, such as the camera 40, 41 or 42, a monitor used for displaying the image, or a scan converter used in producing the image in TV format, between each scan cycle and raster line. For example, where as is customary there are two interlacing scan cycles or fields per frame, the blanking signal on the conductor 346 blanks the beam between each field and raster line. This, of course, is to insure that the beam is turned off during flyback from the end of one scan or line to the beginning of the next.

The section change signal on the output conductor 335 of the section length counter 331 is also fed through a conductor 370 to one input of an OR gate 371 which has another input connected by a conductor 372, a conductor 373, and the conductors 341, 140 and 115 to the vertical reset output of the sync generator 112, (FIG. 4). The output of the OR gate 371 representing either the vertical reset or section change signals is fed through a conductor 374 to a preset input 375 of the section length counter 331 to preset the counter at the end of each raster section and frame, or at the end of each raster section and field if there are two fields per frame, to a count defined by the digital parameter information in the buffer 330. At the same time, the OR gate output signal is fed to the input 326 of each interface unit of the type 287 to transfer the digital parameter information in the buffer 324 through the buffer 325 and converter 328 to the appropriate inputs of the system; to the input 322 of each interface unit of the type 286 to transfer the digital parameter information in the buffer 320 through the buffer 321 and on to the appropriate parameter inputs of the system; and to the input 315 of each unit of the type 285 to transfer the digital parameter information in the shift register 310 serially to the appropriate input of the system. Hence, the section change signal from the section length counter 331 not only initiates the blanking of raster lines between the raster sections, but acts with the vertical reset signal to determine the number of lines in each raster section, by simultaneously transferring the digital parameter information in each interface unit which defines a new raster section to the appropriate inputs of the system.

The signal at the output of the OR gate 371 is also fed through the conductor 374, a conductor 380, and a conductor 381 to a preset input of the channel address counter 302 to preset the counter to a number determined by information fed through a conductor 382 to a buffer input 383. The preset number determines the starting count from which counter 302 addresses the interface units. This count may vary from raster section to raster section. For example, there are certain parameters that remain constant from section to section in a given frame, such as overall X position, overall Y position, and overall depth as these parameters affect all the sections of the image. Color parameters might also

remain constant. To illustrate, if the information for these overall parameters occupies the first 18 interface units, it is necessary to reload these units after each section. Therefore, the counter 322 should be preset to begin its address with the 19th interface unit after the parameter information for the first section of the frame is transferred. The digital computer 22 is programmed to transmit the desired preset value to the input buffer 383.

The channel address counter 302 also has an input connected by a conductor 385 to the output of a delay network 386, the input of which is connected by a conductor 387 and the conductors 373, 341, 140 and 115 to the vertical reset output of the sync generator 112 (FIG. 4), to reset the counter 302 just before the end of each frame or field, if there are two fields per frame. The delay network 386 generates a signal at its output that is delayed from the vertical reset signal by approximately 90 percent of the time period between vertical reset pulses. Therefore, in effect, there is a pulse generated at the output of the delay network 386 that occurs just prior to each vertical reset pulse after the first vertical reset pulse is generated. These pulses from the output of the network 386 are also fed through a conductor 388 to an input of the digital computer 22 to enable the computer 22 for high speed transfer of parameter data.

The signal from the output of the OR gate 371 is also fed through the conductors 374 and 380, and a conductor 391 to an input of the digital computer 22 to start the high speed transfer from the computer 22 to the interface units of the digital parameter information for the next raster section to be generated.

Reviewing the operation of the network of FIG. 6, digital information is stored within the computer 22 defining the parameters of each raster section of the image to be generated for a given frame or frames. The origin of the information may be the director 20 which includes controls for setting the value of each set of parameter information to define each raster section of a particular frame, or may be some digital recording medium such as the digital magnetic tape 36 or disc 38 on which is recorded parameter information defining each raster section of each frame of a sequence. As the information is being stored, an output pulse from the delay network 386 is fed through the conductor 388 to initiate the high speed transfer of the digital parameter information defining the first raster section of the first frame which is fed in a prescribed order in accordance with the digital computer program from the output of the computer 22 to each of the interface units of which in this described embodiment there are a total of 43, including four different basic types. The signal at the output of the delay network 386 initiates the high speed transfer for the first section parameter data only, the high speed transfer of parameter data for subsequent sections being initiated by the vertical reset or section change signals. As the digital parameter information is fed to the interface units, signals are fed through the conductor 300 to the channel address counter 302 generating signals sequentially at a different one of its outputs, each one of which is connected to a different interface unit to get the appropriate digital parameter information from the computer 22 to the appropriate one of the interface units. Hence, the interface units are loaded sequentially with the appropriate information defining the parameters of the first raster section.

When the gating signal appears at the last output of the channel address counter 302, indicating that all the interface units are loaded, this signal is fed through the conductors 308 and 309 to an input of the computer 22 to stop the high speed transfer of digital parameter information.

It will be remembered that the purpose of the interface unit 288 is to define the number of lines in the raster section currently being drawn. However, because 10 the information loaded in the interface units relates to the first raster section, the interface unit 288 cannot perform this function as there is no raster section currently being drawn. Nevertheless, when the next vertical reset signal is generated by the sync generator 112 it is fed through the conductors 115, 140, 341, 373 and 372 the OR gate 371, and the conductor 374 to the inputs 375, 326, 322 and 315 of the interface units to simultaneously transfer the digital parameter information loaded in these units and the others like them defining the parameters of the first raster section to the appropriate parameter inputs of the system, and to preset the section length counter 331. This same vertical reset signal is also fed through the conductors 380 and 391 to an input of the digital computer 22 to start the high speed transfer of digital parameter information defining the second section to the interface units. The vertical reset signal is also fed through the conductor 381 to preset the channel address counter 302 to a number defined by information fed from the computer 22 through the conductor 382 into its input buffer 383. The delayed vertical reset pulse at the input 385 to the counter 302 resets the counter 302 just prior to the start of each frame or field, if there is more than one field per frame.

Along with the other digital parameter information from the digital computer 22 defining the first raster section is information defining the number of lines in the first raster section which is now being generated at the output of the analog portion 26. This information is fed to the interface unit 288 together with the horizontal reset pulses from the sync generator 112. The section length counter 331 counts the horizontal reset pulses at its input 332 for a length of time defined by the value of the information which was transferred from its input buffer 330. When it has reached a full count, a section change signal is generated at its output which is fed through the conductors 335 and 337 to the blanking generator 338 which generates a signal that blanks the first two or three raster lines of the next raster section (which would be the second raster section). This same section change signal is fed through the conductors 335 and 370 and the OR gate 371 to perform the same functions with respect to the second section as the vertical reset signal did with respect to the first, that is, to simultaneously transfer the digital parameter information loaded in the interface units defining the parameters of the second raster section to the appropriate parameter inputs of the system and to preset the section length counter 331, to start the high speed transfer of digital parameter information defining the third section to the interface units, and to preset the channel address counter 302 to a number defined by information fed to its buffer input 383 from the digital computer 22.

The process then repeats itself for the next section and each succeeding section of the first frame, each time sequentially loading the interface units with pa-

parameter information and transferring this information simultaneously to the appropriate inputs of the system to generate the next raster section until the initial frame is generated. The process is then repeated for each succeeding frame of the sequence.

If the parameter information stored within the digital computer 22 remains unchanged, each frame will be identical. If however, the parameter information within the digital computer is changed, as for example by manipulating of the control settings of the analog input 270 and parameter select input 272 of the director 20, or by the sequential playback of digital parameter information recorded on the digital storage medium such as magnetic tape 36 or disc 38 into the digital computer 22, each frame will be different to produce an animation sequence.

#### MODULATION OSCILLATORS (FREQUENCY SYNTHESIZER AND FUNCTION GENERATOR)

Referring to FIGS. 6 and 7, there are shown the frequency synthesizer 32 and function generator 34 of this invention, the purposes of which are to generate modulation signals in response to information received from the digital computer 22 through the interface units for use in producing depth, vertical and horizontal modulations by feeding these signals to the inputs 107, 108 and 109 of the analog portion 26. With these signals each raster section can be formed in a variety of ways to produce a variety of animation effects. For example, vertical modulation can be used to bend the raster lines, horizontal modulation to vary the rate at which a raster line is drawn producing horizontal distortions, and depth modulation to create depth distortions similar to foreshortening effects obtained optically with wide angle lenses.

Many of the major components of the networks of FIGS. 7 and 8 are disclosed in detail in copending U.S. Pat. application Ser. No. 72,642, filed Sept. 16, 1970, now Pat. No. 3,609,917, Frequency Selector and Synthesizer, which will be referenced where appropriate.

The frequency synthesizer 32 generates coherent digital signals of selected frequencies and includes a voltage controlled oscillator (VCO) and master synthesizer 400, the details of which are shown in FIGS. 2 and 2A of the referenced application, Ser. No. 72,642. A minor difference between the synthesizer 400 and the one disclosed in the referenced application is that its master oscillator has a frequency of 9.6 megahertz rather than 10 megahertz so as to be easily divisible to produce the 1.2 megahertz signal. As previously explained, the 1.2 megahertz signal is fed through the conductor 111 to the input of the sync generator 112 of FIG. 4, for generating the horizontal and vertical reset signals. With this difference in master oscillator frequencies, the frequencies generated at the output of the synthesizer 400 are also reduced by factors of 0.96 from the output frequencies of the network of FIG. 2 and 2A of the referenced application. These output frequencies, which need not be described here, since they are clearly described in the referenced application, are fed into a digital frequency selector 401 for producing at its output a digital synthesized signal which appears on an output conductor 402. By simply paralleling digital frequency selectors of the type 401, a plurality of coherent synthesized digital signals can be produced. For example, in this described embodiment, three such signals are necessary for vertical, horizontal and depth modula-

tions and therefore three digital frequency selectors of the type 401 would be required. Since they are identical only one need be described.

The frequency of the output signal from the digital frequency selector 401 is determined by signals from a series of four-to-ten line decoders 404, 405, 406, 407, 408, 409 and 410. The frequency selector 401 is basically the same type as disclosed in FIG. 4 of the referenced application, except that the switches 501 through 507 of that FIG. 4 used to select the synthesized frequency desired are replaced by the decoders 404 through 410 and a shift register 411. The shift register 411 has an input conductor 412 connected to a control mode switch 414 having an internal position 415 and an external position 416. In either mode setting the shift register 411 is loaded with digital parameter information which is fed through suitable conductors to the decoders 404 through 410. Each of these decoders has 10 outputs representing the number 0 through 9, one of which is activated in accordance with the binary coded decimal (BCD) number at its input. The signals at the outputs of the decoders 404 through 410 perform the same function as the switches 501 through 507 of FIG. 4 of the referenced application to gate the appropriate ones of the output frequencies from the synthesizer 400 to a series of frequency adders such as the adders 530, 540 and 544 of FIGS. 4 and 5 of the referenced application to produce at the output of the selector 401 a synthesized signal of the selected frequency.

With the mode switch 414 in its external position, digital parameter information defining the frequency of the synthesized signal is fed serially from the output 316 of an interface unit of the type 285 to an input of the shift register 411. Simultaneously with the transfer of the parameter information, the clock pulses on the output conductor 317 of the same interface unit are fed to an input of the shift register 411 to load the shift register with the parameter information. After the shift register 411 is loaded, the digital parameter information in BCD form is fed to the decoders 404 through 410 which decode the information and feed it to inputs of the selector 401 for defining the frequency of the synthesized signal at its output 402. Hence, with the mode switch 414 in its external position, the frequency of the synthesized signal is controlled by the digital computer 22.

With the switch 414 in the internal position, the frequency of the synthesized signal is selected by setting a series of BCD, seven-place, thumb-wheel switches 420, 421, 422, 423, 424, 425 and 426 to the frequency desired. With the switch 414 in this mode setting, the digital computer 22 has no effect in selecting the frequency of the synthesized signal from the selector 401. The frequency is controlled exclusively by the settings of the BCD switches 420 through 426 which feed information in binary coded decimal form directly through the shift register 411 to the decoders 404 through 410, the outputs of which define the frequency of the synthesized signal as heretofore described.

Having defined the frequencies of the modulation oscillators, it is necessary to define their phases, amplitudes, waveform, and synchronization modes. This is accomplished in the function generator 34 of which there is one for each modulation signal required. The synthesized digital output signal from the digital frequency selector 401 of the frequency synthesizer 32 is fed through the conductor 402 to an input of an up-

down counter 440 which counts the pulses of the synthesized digital signal alternately upward and downward between prescribed limits producing binary weighted outputs which are fed to the input of a digital to analog converter 442. The converter 442 produces at an output conductor 444 a staircase triangular waveform of a frequency depending on the frequency of the signal from the frequency synthesizer 32 and the upper and lower limit settings of the up-down counter 440. For example, if the up-down counter 440 is set to count alternately up and down between counts of zero and 50, the frequency of the triangular waveform output from the converter 442 would be 1/100 of the frequency of the synthesized signal. If the up-down counter 450 is set to count alternately up and down between counts of zero and 500, the frequency of the triangular waveform output from the converter 442 is 1/1,000 of the frequency of the synthesized signal from the frequency synthesizer 32. The up-down counter 450 and digital to analog converter 442 perform generally the same function in this circuit as the up-down counter 22 and digital to analog converter 24 performs in the circuit of FIG. 7 of the above-referenced application Ser. No. 72,642.

The up-down counter 440 has a triangle/sawtooth enable input 446 which controls whether the counter 440 counts alternately up and down to produce a triangular waveform from the output conductor 444 or is reset each time it reaches an upper limit (or lower limit) to produce a sawtooth waveform at the output conductor 444. In either case the triangle or sawtooth waveform output from the converter 442 is fed through the conductor 444, a conductor 448 and a resistor 450 to an analog gate 452. The wave form on the conductor 444 is also fed through a sine shaper 454 to produce at its output a sinusoidal waveform which is fed through a conductor 456 and resistor 458 to an analog gate 460. An output from the up-down counter 440 is fed through a conductor 462 to a level converter 464 to produce at its output a square waveform which is fed through a conductor 466 and a resistor 468 to an analog gate 470. Hence, the counter 440, converter 442, sine shaper 454 and level converter 464 produce triangle, saw tooth, sine and square waveforms from the digital frequency signal of the frequency synthesizer 32. It is still necessary however to select a particular one of these waveforms for each modulation signal, and to define its phase and synchronization mode.

A shift register 472 receives serial parameter information and serial clock pulses at input conductors 316 and 317, respectively, from an interface unit of the type 285, which serial parameter information defines the phase, waveform and synchronization mode of a particular modulation signal. The phase data information is transferred from the shift register 472 through suitable conductors to a phase angle translator 474, while the data defining the waveform and synchronization mode is transferred through suitable conductors to a function control circuit 476. The function control circuit 476 has an input connected by a conductor 478 and the conductors 333, 334, 116 and 114 to the horizontal reset output of the sync generator 112 (FIG. 4), and an input connected by a conductor 480 to the output conductor 337 of the interface unit of the type 288 for receiving the section change signals.

The function of the phase angle translator 474 and shift register 472 are to set the phase of the modulation

waveform. To explain how this is done, suppose that the counter 440 is made to start from a count of zero and count alternately between counts of zero and 50 so that a complete cycle is equivalent to 100 counts. For purposes of establishing a phase for the waveform, a single cycle is divided into 100 counts with each count equivalent to a particular phase. The phase angle translator 474 receives the phase data from the shift register 472 and converts it to binary weighted outputs which are fed to the up-down counter 440 to preset its count to a prescribed number, and also sends a signal either through a conductor 482, causing the counter 440 to count up, or a conductor 484, causing the counter 440 to count down from the prescribed number, thereby setting the phase of the waveforms produced therefrom. The phase data in the shift register 472 has a binary weight of some number between zero and 100, which data is fed to the phase angle translator 474. If the data represents a number between zero and 50, then the phase angle translator 474 transfers that data directly to the counter 440 together with a signal on the conductor 482 causing the counter to count up from that number. However, if the phase data in the shift register 472 represents a count of between 50 and 100, the output of the shift register would have no meaning to the counter 440 since it never reaches counts of over 50. Therefore, the phase angle translator 474 translates data representing numbers between 50 and 100 to numbers between 0 and 50 which do have meaning to the counter 440. For example, if the data from the shift register 472 represents the number 60, to designate a corresponding phase the translator 474 translates the number 60 to the number 40 and transfers data representing the number 40 to the counter 440 together with a signal on the conductor 484 causing the counter 440 to count down. Hence, the purpose of the phase angle translator 474 is to convert phase data from the shift register 472 to data that has meaning to the counter 440 for establishing the phase of its output waveform.

If the counter 440 is in its divide-by-1,000 mode so that it counts alternately between zero and 500, a complete cycle would have 1,000 counts and the phase data fed through the conductor 316 to the shift register 472 would represent some count between zero and 1,000 to define the phase of the output waveform from the counter 440.

The function control circuit 476 has an output conductor 490 which carries an enable signal to the gate input of the analog gate 452, to gate the triangle or sawtooth waveform, as the case may be, from the output of the converter 442 to the input of a summation amplifier 491. The function control circuit 476 has another output conductor 493 which carries an enable signal to the gate input of the analog gate 460 for gating the sinusoidal waveform from the sine shaper 454 to the input of the summation amplifier 491. The function control circuit 476 has another output conductor 494 which carries an enable signal to the gate input to the analog gate 470 to gate the square waveform from the output of the level converter 464 to the input of the summation amplifier 491. The parameter data in the shift register 472 defining the waveform of a particular modulation signal is fed to the function control circuit 476 which sends a signal through the conductor 490, 493 or 494 to gate the selected waveform signal through the summation amplifier 491 to one input of a

multiplier 496. The multiplier 496 has another input connected by a conductor 497 to the output conductor 329 of a interface unit of the type 287 for receiving information defining the amplitude of the selected waveform.

To produce certain animation effects it is desirable to phase-lock the modulation signals with the generation of a particular raster section or with the generation of each raster line of a particular raster section. The latter synchronization mode might be desirable, for example, in producing a bend in a raster section where each line of the raster must be bent at precisely the same place. It might also be desirable to have no phase lock, allowing the modulation signal to free run. To provide these synchronization modes, the parameter data in the shift register 472 defining the synchronization mode is fed to the function control circuit 476 which, in accordance with the synchronization mode selected, transmits either the horizontal reset signals at its input conductor 478, the section change signals at its input conductor 480, or neither if in the free run mode, through an output conductor 498 to an input of the up-down counter 440. At each pulse on the input conductor 498, the counter 440 is reset to begin counting from a phase condition defined by the phase data from the phase angle translator 474. The function control circuit 476 also sends a signal through a conductor 499 to the counter 440 to time the loading of the phase data from the translator 474 into the counter 440.

Hence, in this manner, a modulation signal of a defined frequency, phase, waveform, amplitude and synchronization mode is produced at the output of the multiplier 496 and fed through a conductor 500 to an appropriate modulation input to the analog portion 26 such as the input 107, 108 or 109 for use in producing depth, vertical or horizontal animation. Since three modulation signals are needed in this embodiment, three function generators of the type described are required.

#### INTENSITY COMPENSATION

Referring to FIGS. 4 and 4A, there is shown an intensity compensation network 520 the details of which are disclosed in copending U.S. Patent application Ser. No. 74,662, filed Sept. 23, 1970, entitled Beam Intensity Compensator, having an input conductor 521 connected from an intensity input 522. The input 522 receives analog information from an interface unit of the type 287 defining the beam intensities of the scanning devices of the system on which the image is produced which devices include the scan convertors of the color network 67, the X-Y monitors 79, 81 and 83 of FIG. 3, and the overlap network 56 of FIG. 1. The network 520 also has an input conductor 523 connected from the vertical size input 100 to carry vertical size information, an input conductor 524 connected from the horizontal size input 97 to carry horizontal size information, an input conductor 525 connected to the output of the summation amplifier 230 to carry depth information, an input conductor 526 connected from the vertical modulation input 108 to carry vertical modulation signals, and an input conductor 527 connected from the horizontal modulation input 109 to carry horizontal modulation signals. The purpose of the intensity compensation network 520 is to compensate the beam intensities of the display devices for variations in size of the image and scan velocity of the spot as the beam

travels across the screen of the device. The intensity compensation signal for the monitored display is fed through a conductor 530 to modulate the video signals to the scan converters or X-Y monitors, and the intensity compensation signal for the overlap network is fed through a conductor 531 to modulate the video signal to the overlap network 56.

#### OPERATION

To describe the operation of the system, it is best to consider an example of how a particular animation sequence is generated by referring to FIGS. 8 through 12. Suppose it is desired to produce an animated sequence of a cartoon caricature of a man walking with the wind blowing his hat off. The character includes a right arm 601, a right leg 602, a hat 603, a head and body 604, a left leg 605, and a left arm 606. As previously described, an animation sequence is composed of a finite number of frames, the frequency of which depends upon whether the sequence is to be filmed, recorded on video tape, or displayed on a TV monitor. If the sequence is to be filmed, the frame rate should be 24 frames per second to be compatible with movie film rates; if the image is to be displayed on a TV monitor or recorded on video tape for use in the United States, the frame rate should be 30 frames per second for compatibility with the US TV scan rate. To produce the sequence, it is first necessary to establish image parameters for initial and final frames of the sequence, and the manner in which the image is to move between initial and final frames, requiring among other things the selection of the number of frames in the sequence, the number of raster sections on which the image is to be displayed, and the fairing functions defining the rates of parameter change throughout the sequence. All of these variables are in the control of the operator.

Suppose in this illustration that during the sequence the man takes one half step from an initial position shown in FIG. 9 to a final position shown in FIG. 10, and that as he takes the one half step, his hat is blown off his head as shown in FIG. 10. Further suppose that his arms and legs move from an initial position to a final position as shown.

Because there are a total of six different parts 601 through 606 of the figure moving relative to one another during the sequence, each of these parts are produced on a separate raster section; the right arm 601 on a raster section 610, the right leg 602 on a raster section 611, the hat 603 on a raster section 612, the head and body 604 on a raster section 613, the left leg 605 on a raster section 614, and the left arm 606 on a raster section 615. Therefore, controls on the director 20 are set to select a total of six raster sections and to establish the number of raster lines in each section.

Because the overlap network 56 of FIG. 1 blanks the last generated overlapping information, the parts of the figures should be generated from foreground to background. Hence, the raster sections should be generated in the following order: right arm section 610, right leg section 611, hat section 612, head and body section 613, left leg section 614, and left arm section 615. These various body parts are drawn from top to bottom on a piece of artwork 618 and displayed such as on the light box 45 of FIG. 1. A video camera such as the camera 42 of FIG. 1 is made to scan the artwork 618 from top to bottom at rates determined by the horizontal and

vertical reset pulses from the sync generator 112 of FIG. 1.

At this point it should be mentioned that it is not absolutely necessary that the artwork 618 be arranged so that the camera 42 scans from top to bottom. With the appropriate circuitry, the camera can be made to scan the various parts of the artwork in any desired sequence to accommodate the overlap network 56. In any case, for the purposes of this example, it will be assumed that the artwork is drawn in the sequence shown 10 and that the camera scans from top to bottom.

Appropriate controls of the director 20 are set to feed digital parameter information to the digital computer 22 which in turn is fed to the switcher 52 as the sequence is generated to switch the video information 15 from the camera 42 to the input of the grey level encoder 54 which produces at its output signals representing the artwork 618 in discrete shades of grey. The output signals from the grey level encoder 54 are fed to the video gate 58 together with signals from the output of the overlap network 56 which prevent overlapping on the part of the image where one part is positioned behind another, to produce on the output conductor 60 from the video gate 58 video signals representing the artwork in discrete shades of grey for use in producing the animation sequence.

The operator must also select the number of frames in the sequence. This, of course, depends on how fast the figure is to walk, but assume for purposes of this example that he is to walk at a rate of half a step per second. With TV rates of 30 frames per second, 30 frames are selected for the sequence by setting the controls on the director 20. If the sequence is to be filmed, 24 frames are selected. By adjusting the analog input 270 and channel select 272 controls of the director 20, and by observing the monitor 73 of FIG. 2, the operator varies all of the parameters for each raster section of the image necessary to set up the initial frame of the sequence with the image of FIG. 9. Hence, for each raster section 610 through 615, digital signals are sent sequentially to the digital computer 22 representing X position, Y position, horizontal size, vertical size, section depth, and intensity, as well as signals defining the number of lines in each raster section and the red, blue and green color components of each color in each frame. Additionally, the arms, legs, and hat must be made to rotate about given points between initial and final frames, the hat being made to rotate about a point 620, the legs about a point 621 located at the hips, and the arms about a point 622 located at the shoulders. In this illustration the head and body section 613 remains erect throughout the sequence although, of course, it could be made to rotate if desired. Therefore, for each raster section, digital signals are sent to the digital computer 22 from the director 20 defining the cosine and sine of the angles of rotation, and the vertical and horizontal axes of rotation.

Certain other parameters must also be defined. It will be noted that in the initial frame the left arm has a bend at the elbow and the left leg has a bend at the knee, whereas in the final frame the right arm has a bend at the elbow and the right leg has a bend at the knee. Vertical modulation signals are used to produce these bends, which signals are generated by the frequency synthesizer 32 and function generator 34 from parameter information supplied them from the digital computer 22. Each of these modulation signals has five pa-

rameters that must be defined: frequency, phase, waveform, amplitude and synchronization mode. Since elbow and knee bends are fairly sharp, a triangular waveform is selected as more suitable than a sinusoidal or square waveform. Also, these bends require that each raster line of a raster section on which one of the arms or legs is produced be bent the same amount and in exactly the same place. Therefore, synchronization mode should be selected such that the generation of the modulation signals for each raster section is synchronized with the generation of each raster line in the section, i.e., with the horizontal reset pulses. Hence, the controls of the director 20 are set to select a triangular waveform, synchronization on the horizontal reset signals, and the appropriate frequency, phase and amplitude for the vertical modulation signal for each of the raster sections 610, 611, 614 and 615 for the initial frame of the sequence.

In addition to the individual section parameters, the 20 controls of the director 20 are set to establish the parameters for the overall image of the initial frame, namely, overall X position, overall Y position and overall depth.

As all of this parameter information is sent to and 25 stored in the digital computer 22, the digital computer 22 is programmed to feed the parameter information for each raster section of the initial frame to the various interface units, as heretofore described, beginning with the information defining the raster section 610 and followed by the information for the raster section 611, 612, 613, 614 and 615, in that order. The channel address counter 302 sequentially directs each piece of 30 digital parameter information to the appropriate interface unit as heretofore described. Hence, information defining raster section 610 is first sequentially loaded 35 into the interface units which information is transferred from these units to the appropriate input of the other networks of the system in response to the vertical reset signal from the sync generator 112, at which time the digital parameter information for the next section 611 is loaded sequentially into the interface units and transferred to appropriate inputs of the system in response to the section change signal at the output of the section length counter 331 (FIG. 5), and so on for the section 40 612, 613, 614 and 615 to generate the initial frame of the sequence. The sections 610 to 615 are, of course, drawn continuously, but two or three raster lines are blanked at the beginning of each section. The blanking is produced by signals generated by the blanking network 338 in response to the section change signals from the section length counter 331, the horizontal reset pulses, and the vertical reset pulses. The section change signals in turn are generated in response to information stored in the digital computer 22 defining the 45 number of lines in each raster section. These blanking signals from the blanking generator 338 are fed through the conductor 345 to blank the beam of the scan converter of the color network 67 between sections.

Upon transfer of each block of parameter information defining each raster section from the interface units, digital parameter information is fed from the output of the interface units of the type 285 to the input of the shift register 411 of the frequency synthesizer 32 to define the frequency of the vertical modulation signal, and to the inputs of the shift register 472 of the function generator 34 for defining the phase, waveform

and synchronization mode of the vertical modulation signal. Also for each raster section, an analog parameter signal is fed from the output of an interface unit of the type 287 through the conductor 497 to an input of the multiplier 496 of the function generator 34 defining the amplitude of the modulation signal. The result is to produce at the output conductor 500 of the function generator 34 a series of modulation signals of frequencies, phases, waveforms, amplitudes and synchronization modes that produce the bends in the raster sections 610, 611, 614 and 615 of FIG. 12. The modulation signal for each raster section is fed to the vertical modulation input 108 of the analog portion 26.

Digital parameter information is fed from interface units of the type 286 to perform various network control functions such as the setting of the switch 52 to switch the video information from the camera 42 through to the grey level encoder 54.

Analog parameter signals are fed from the outputs of the interface units of the type 287 to the overall X position input 95, section X position input 96, section horizontal size input 97, section horizontal axis of rotation input 98, section vertical axis of rotation input 99, raster vertical size input 100, raster Y position input 101, overall Y position input 102, section depth input 103, overall depth input 104, cosine R input 105, sine R input 106, and intensity input 522 of the analog portion 26. Where a color display or video tape of the animation sequence is to be produced, analog voltages from other units of the type 287 are fed to the color network 67 of FIG. 2 to define the red, blue and green color components for each discrete grey shade. Where the sequence is to be recorded on color film, these signals are fed to the RGB color encoder 77 of FIG. 3.

As the analog parameter signals defining each raster section are fed to the inputs of the analog portion 26 sequentially by raster section, the integrators 118 and 144 of the analog portion 26 are generating horizontal and vertical ramp functions synchronized with the horizontal and vertical reset pulses from the sync generator 112. Since the video camera 42 also scans in synchronization with the horizontal and vertical reset pulses from the sync generator 112, the generation of the horizontal and vertical ramp functions from the integrators 118 and 144 is synchronized with the scan of the video camera 42 scanning the artwork 618. The horizontal and vertical ramp functions are combined with the analog parameter signals at the input of the analog portion 26 as heretofore described producing, at the output conductors 92 and 93, X and Y coordinate output signals sequentially representing the raster sections of the initial frame.

These X and Y coordinate signals are fed to an input of the overlap network 56 together with the video signals from the output of the grey level encoder 54 to produce the overlap compensated video signal on the conductor 60 representing the artwork in discrete shades of grey. The X and Y coordinate output signals from the analog portion 26 and the video signals on the conductor 60 can be fed to the color network 67 to produce a color display of the initial frame on the color monitor 73. The operator in setting up the initial frame can observe the effects of varying the controls of the director 20 on the color monitor 73 until he is satisfied with the initial frame image.

After he is satisfied with the initial frame, the operator initiates a control signal on the director 20 to the

digital computer 22 to store the initial frame parameter information in the computer 22. Having done this, he sets up the parameters for the final frame in exactly the same manner with the networks sequentially generating each section of the final frame of the sequence as the operator varies the controls of the director 20 until he is satisfied with the final frame parameter.

In most cases, the parameters of the final frame are different from those of the initial frame. For example, 10 in this illustration the X and Y position and angle of rotation parameters of the hat 603, and therefore the section 612 are different. The angles of rotation of the arms about the shoulders and hence the sections 610 and 615 are different, as are the angles of rotation of the legs about the hips and hence the sections 611 and 614; also, since the left arm and left leg go from bent to straight positions, and the right arm and right leg go from straight to bent positions, the amplitudes of the vertical modulation signals for the sections 610, 611, 20 614 and 615 are different.

Having established the initial and final frame parameters for the sequence, appropriate fairing functions must be selected to define the rates of change for these parameters throughout the sequence. Fairing functions can be selected that vary linearly, exponentially, or in some other manner. To put it in terms of this illustration, the hat 620 can be made to leave his head very suddenly and then blow away at a constant rate or perhaps at a decreasing rate; the arms can be made to bend and straighten at the elbows in a constant manner throughout the sequence, or perhaps slowly at first, gradually increasing, and then slowly at the end of the sequence. The same is true with respect to the straightening and bending of the knees as well as the rates of 25 rotation of the hat and each arm and leg throughout the sequence. For purposes of this example, assume that the hat is to blow off suddenly and then proceed at a constant rate, and that further it is to rotate from the position of the initial frame to the position of the final frame at a constant rate. Therefore, linear fairing functions are selected to define the parameter changes between initial and final frames of the raster section 612. Also assume that the bends in the knees and elbows are to change at a constant rate throughout the sequence, 30 so that a linear fairing function is selected to define the amplitude changes of the vertical modulation signals throughout the sequence. Assume, however, that the rotation of each arm about its shoulder point 622 and each leg about its hip point 621 is to begin slowly, gradually increasing at the middle of the sequence, and then gradually slowing to the end of the sequence. Appropriate fairing functions are selected to vary the angles of rotation of the raster sections 610, 611, 614 and 615 accordingly. In selecting these fairing functions, appropriate command signals are sent from the director 20 to the digital computer 22 which is programmed to make parameter calculations in accordance with these selected functions.

The digital computer 22 now has all the information 35 for defining the initial and final frames of the sequence as well as the fairing function information as to how the image parameters should vary throughout the sequence. With the digital computer 22 programmed to calculate the section parameters for each frame between initial and final frames of the sequence in accordance with the fairing functions selected upon appropriate command from the director 20, the digital com-

puter 22 goes into its record mode and automatically computes this information and records it on the digital recording medium such as magnetic tape 36 and/or disc 38.

With the digital parameter information defining the entire animation sequence recorded, upon appropriate command from the director 20 the control network is placed in the playback mode to play the digital parameter information recorded on the tape or disc back to the digital computer 22 from which it is fed section by section and frame by frame as heretofore described to the interface units. From there it is fed to other inputs of the system to produce a color display or video tape of the animation sequence. Also the video signals on the conductor 60 can be fed to the RGB color encoder 77, 15 and the X and Y coordinate output signals from the analog portion 26 fed to the X-Y monitors 79, 81 and 83 (FIG. 3) to produce a color film of the sequence.

With this sequence completed, the operator may wish to create a second sequence with the cartoon character of FIGS. 9 and 10 taking the second half step. To do this, an initial frame is set up for the second sequence the same as the final frame of the first sequence, and a final frame is set up for the second sequence the same as the initial frame of the first sequence. The operation heretofore described is then repeated to produce the second sequence so that now a sequence has been produced of the character taking a full step. By appropriately rerecording the digital parameter data for this sequence on the digital magnetic tape 36, a sequence of any length can be produced of the character walking.

This example is only meant to illustrate the basic operating principles of the system and by no means describes the multitude of animation variations that can be achieved on this system. Indeed, by appropriately selecting scene length and parameter variations such as fairing functions and modulation signals, an unlimited variety of animation effects can be achieved. As still a further illustration of the versatility of this system, it should be noted that once the parameters defining the raster sections, such as the raster sections 610 through 615, have been established for a given animation sequence, any figure can be produced thereon and caused to animate in exactly the same manner by simply changing the artwork 618. For example, if the artwork 618 were changed to show the parts of a rabbit, the rabbit would be produced on the raster sections 610 through 615 and caused to move in the animation sequence just as the characters of FIGS. 9 and 10. In fact, it may be desirable in producing certain effects to have no artwork at all and instead show the raster sections themselves which in the above example would produce an animation sequence of the rasters shown in FIGS. 11 and 12.

Various changes and modifications may be made within the invention as will be readily apparent to those skilled in the art. Such changes and modifications are within the scope and teaching of this invention as defined by the claims appended hereto.

What is claimed is:

1. A method of producing an animation sequence of a subject from a single still view of the subject comprising the steps of generating video signals representing each part of the subject as selectively divided into one or more parts; reproducing each of said parts of the subject on a separate raster section generated from pa-

rameter signals defining its size, shape, position, and structure; the generation of each raster section being synchronized with the generation of the video signals representing the part of the subject produced thereon,

5 establishing a series of digital signals representing the parameters for each raster section over a given time interval, which digital signals are selectively varying over said time interval, and selectively varying the parameter signals over said time interval in response to the digital signals to produce changes in selected ones of the raster sections, thereby producing corresponding changes in the parts of the subject reproduced thereon.

2. The method of claim 1 further comprising the step of recording the sequence.

3. The method of claim 1 further comprising the step of displaying the parts of the image as reproduced on the raster sections.

20 4. The method of claim 1 further comprising the step of generating video signals representing background information in synchronization with the production of the animation sequence of the subject.

5. A method of producing an animation sequence of a subject from a single still view still the subject comprising the steps of scanning each part of the subject as selectively divided into one or more parts to produce video signals representing each part, generating parameter input signals defining the size, shape, position and structure of a raster section for each of said parts of the subject, combining the parameter input signals to generate, in synchronization with the generation of the video signals representing each part of the subject, time varying coordinate signals defining a raster section,

25 modulating the intensity of the electron beam of an electron beam device with the video signals representing each part of the subject while simultaneously directing the scan pattern of the electron beam with the coordinate signals produced in synchronization therewith to reproduce each part of the subject on a separate raster section, establishing a series of digital signals defining the parameters for each raster section over a given time interval, which digital signals are selectively varying over said time interval, and selectively varying the parameter input signals over said time interval in

30 response to the digital signals, thereby producing corresponding changes in the scanning patterns of the electron beam, the raster sections produced thereby and the parts of the subject produced thereon.

6. The method of claim 5 wherein the position parameters include parameters that define the angles of rotation of the raster sections, and the method further comprising the step of varying over time the angle of rotation parameters for selected ones of the rasters to vary the angles of rotation of these rasters with respect to reference axes.

7. The method of claim 5 wherein one of the scanning pattern changes produces changes in the degree of bend in the raster lines of selected ones of the raster sections.

60 8. The method of claim 5 including the step of generating in synchronization with the generation of the video signals horizontal and vertical sweep signals of selected slopes as part of the parameter input signals.

9. The method of claim 5 further comprising the steps of generating modulation signals, and selectively modulating the sweep signals with the modulation signals.

10. The method of claim 5 further comprising the step of generating video signals representing background information for the animation sequence and combining the background information video signals with the video signals representing each part of the subject to produce an animation sequence of the subject with background.

11. The method of claim 10 further comprising the step of blanking parts of the display positioned behind other parts of the display.

12. A method of producing an animation sequence of an image, the sequence divided into frames between initial and final frames, the method comprising the steps of generating first sets of digital signals representing scan pattern parameters for the initial frame of the sequence, generating second sets of digital signals representing scan pattern parameters for the final frame of the sequence, generating further sets of digital signals using the initial and final digital parameter signals as references representing scan pattern parameters for each frame therebetween in accordance with selected functions defining the patterns of parameter change from frame to frame throughout the sequence, each set of digital signals in each frame representing a distinct scan pattern, converting the digital signals in each set of digital signals to analog signals, generating from each first set of analog signals a distinct scan pattern, the scan patterns from the first sets representing the image of the initial frame, and generating a distinct scan pattern from each further set of analog parameter signals for each subsequent frame through the final frame of the sequence to produce a series of images.

13. The method of claim 12 further comprising the step of reproducing the parts of a subject divided into a selected number of parts on the scan patterns in each frame of the sequence.

14. The method of claim 13 further comprising the steps of causing the beam of a video camera to scan each part of the subject in synchronization with the generation of a scan pattern to produce video information representing each part of the subject, there being as many scan patterns generated per frame as there are parts of the subject, and producing an animated sequence of the subject from the scan patterns and video signals.

15. The method of claim 14 further comprising the steps of producing a recording of the sequence.

16. The method of claim 12 further comprising the steps of recording the digital signals representing the scan pattern parameters, and playing the digital recording back to produce the analog signals.

17. The method of claim 12 including the steps of loading the digital signals in each set sequentially into buffer units, and transferring the digital signals in each set simultaneously from the buffer units to generate a scan pattern.

18. A method of generating an image comprising the steps of generating horizontal and vertical sweep signals representing straight-line raster scans, establishing sets of digital data, the digital data in each set defining parameters representing the size, shape, position and structure of a raster section, converting the digital data in each set to analog parameter signals, combining the analog parameter signals in each set and the sweep signals in a selected manner and sequence to produce time varying coordinate signals representing a series of raster sections, generating modulation signals, combin-

ing the modulation signals with selected ones of the analog parameter signals, the digital signals defining selected raster sections including signals defining the frequency phase, amplitude, and waveform of the modulation signals.

19. The method of claim 18 further comprising the step of applying the coordinate signals to the beam deflection inputs of a cathode ray tube display device to produce a display of the image.

10 20. The method of claim 19 further comprising the steps of generating video signals representing each part of a subject divided into a selected number of parts in synchronization with the generation of a raster section, and modulating the intensity of the cathode ray tube beam with the video signals to produce a display of the subject, whereby the size, shape and position of each part of the displayed subject are determined by the size, shape and position of the raster section on which it is reproduced.

20 21. The method of claim 20 further comprising the step of compensating the beam intensity of the cathode ray tube for variations in size of the image and velocity of the spot as the beam scans.

25 22. The method of claim 18 further comprising the step of combining a modulation signal of a selected frequency, phase, amplitude and waveform with the vertical sweep signal to bend the lines of selected raster sections.

30 23. The method of claim 18 further comprising the step of combining a modulation signal of a selected frequency, phase, amplitude and waveform with the horizontal sweep signal to vary the rate at which the raster lines of particular raster sections are drawn.

35 24. The method of claim 18 further comprising the step of combining a modulation signal of a selected frequency, phase, amplitude and waveform with the analog parameter signals defining the depth of particular raster sections.

40 25. The method of claim 18 further comprising the step of synchronizing the generation of a modulation signal with the generation of each line of a selected raster section.

45 26. The method of claim 18 wherein the analog parameter signals include signals representing the sines and cosines of angles through which selected raster sections are to be rotated with respect to a reference axis, and further comprising the step of combining combinations of other analog parameter signals with the sine and cosine parameter signals to rotate the selected raster sections.

50 27. The method of claim 26 wherein some of the combinations of other analog parameter signals combined with the sine and cosine parameter signals are produced by combining modulation signals with selected ones of the analog parameter signals.

55 28. A computer animation system for generating an animation sequence comprising a digital computer means, means for feeding digital data to the digital computer means defining certain parameters of an image for an initial frame of the sequence, means for feeding digital data to the digital computer means defining certain parameters of the image for a final frame of the sequence, means associated with the digital computer means for automatically calculating, upon command, the digital data defining certain parameters of the image for each frame between initial and final frames in accordance with selected patterns of param-

ter change throughout the sequence, means for converting the digital data to analog signals, and means for combining the analog signals to produce signals representing the animation sequence.

29. The system of claim 28 including digital recording means, means for recording the digital data defining the parameters of the image for each frame of the sequence on the digital recording means, and means to play back the digital data on the digital recording means.

30. The system of claim 28 including a subject, means for producing video signals representing the subject, and means responsive to the video signals and signals representing the animation sequence to produce an animation sequence of the subject.

31. The system of claim 30 including means responsive to the video signals and signals representing the animation sequence to produce a color representation of the animation sequence of the subject.

33. The system of claim 30 including means for blanking parts of the subject positioned other parts of the subject.

33. A system for generating time varying coordinate signals for producing an animation sequence of an image on a display device, film, or video tape, the system comprising an analog network means having analog inputs and outputs, means for generating horizontal and vertical sweep signals representing straight line raster scans, circuit means associated with the analog network means for generating time varying coordinate signals representing a particular raster section at its outputs from a given set of parameter signals and the sweep signals at its inputs, the parameter signals defining the size, shape, position and structure of the raster section, digital input means for storing and transferring sets of digital data defining the parameters of each raster section to be generated, means for feeding digital data to the digital input means defining each parameter of each raster section to be generated, means for feeding the sets of digital data relating to the parameters of each raster section to be generated from the digital input means in a prescribed sequence, means for converting the sets of digital data from the digital input means to sets of analog parameter signals, means for feeding the sets of analog parameter signals and the sweep signals to the appropriate inputs of the analog network means to produce the time varying coordinate signals representing a series of raster sections, means to generate video signals representing each part of a subject divided into a selected number of parts, the generation of the video signals for each part of the subject being synchronized with the generation of the time varying coordinate signals representing a raster section, and means responsive to the video and coordinate signals for producing a display of the animation sequence of the subject.

34. The system of claim 33 wherein the sequence is comprised of frames, the video signals representing each part of the subject being generated during each frame.

35. The system of claim 33 including means for generating signals defining the red, blue and green color components for colors assigned to each part of the sub-

ject, and means responsive to the video signals, coordinate signals, and color component signals to produce a color representation of the animated subject.

36. The system of claim 33 including means defining the number of lines in each raster section.

37. The system of claim 33 including means for rotating each raster section about a selected axis of rotation.

38. The system of claim 37 wherein the analog parameter input signals defining each raster section include signals defining the size and axis of rotation of the raster section, and including means for automatically compensating the axis of rotation for variations in size of the raster section.

39. The system of claim 38 wherein the size parameter signals are multiplied by the axis of rotation parameter signals.

40. The system of claim 33 including means for feeding the digital data within each set of digital data to the conversion means in a prescribed sequence, and means to simultaneously transfer the digital data in each set of digital data to the inputs of the analog network means.

41. A system for generating an animation sequence of an image composed of one or more raster sections, each raster section being defined in terms of size, shape, position and structure by analog parameter signals, the sequence divided into frames between initial and final frames, the system comprising analog network means having coordinate outputs and parameter inputs, means associated with the analog network means for generating time varying coordinate signals at its outputs representing a raster section of a size, shape, position and structure defined from a set of analog parameter signals at its input, digital computer means, means for feeding digital data to the digital computer means defining the parameters of the raster sections for the initial frame of the sequence, means for feeding digital data to the digital computer means defining the parameters of the raster sections for the final frame of the sequence, means associated with the digital computer means for establishing certain parameters of the raster sections between initial and final frames, means for generating as many sets of digital signals as there are raster sections composing the image for each frame, each set of digital signals defining the parameters of a raster section, means for loading the digital signals in each set in timed sequence into a series of interface units beginning with the set defining the first raster section to be generated in the first frame and continuing in timed sequence through the set defining the last raster section to be generated in the last frame, means for simultaneously initiating the transfer of the digital signals from the interface units after the last of the series of interface units is loaded with one set of signals, means for loading the digital signals of the next set into the interface units as soon as the prior set is transferred therefrom, means for converting the digital signals from the interface units to analog signals, and means for applying the analog signals to the inputs of the analog network means to produce time varying coordinate signals at its outputs sequentially representing the raster sections of the image for each frame throughout the sequence.

# United States Patent [19]

Cooper

[11] Patent Number: 4,789,894  
[45] Date of Patent: Dec. 6, 1988

[54] MOTION ANALYZER WITH INTERLEAVED IMAGE REPRODUCTION

[75] Inventor: Todd H. Cooper, San Diego, Calif.

[73] Assignee: Eastman Kodak Company, Rochester, N.Y.

[21] Appl. No.: 26,078

[22] Filed: Mar. 16, 1987

[51] Int. Cl.<sup>4</sup> ..... H04N 7/18; H04N 7/01; H04N 7/12

[52] U.S. Cl. ..... 358/105; 358/140;

358/134; 358/312; 358/335; 360/10.1

[58] Field of Search ..... 358/105, 134, 312, 335, 358/140; 360/9.1, 10.1

## [56] References Cited

### U.S. PATENT DOCUMENTS

- 2,943,141 6/1960 Knight ..... 358/108  
3,924,267 12/1975 Holt ..... 360/9.1  
4,280,151 7/1981 Tsunekawa et al. ..... 360/9.1  
4,322,638 3/1982 Lee et al. ..... 307/311  
4,322,752 3/1982 Bixby ..... 358/213.27

4,330,796 5/1982 Anagnostopoulos et al. ..... 358/213.29  
4,395,236 7/1983 Gotthold ..... 358/335  
4,496,995 1/1985 Colles et al. ..... 360/9.1  
4,511,886 4/1985 Rodriguez ..... 358/105  
4,575,771 3/1986 Oota et al. ..... 360/10.1

Primary Examiner—James J. Groody

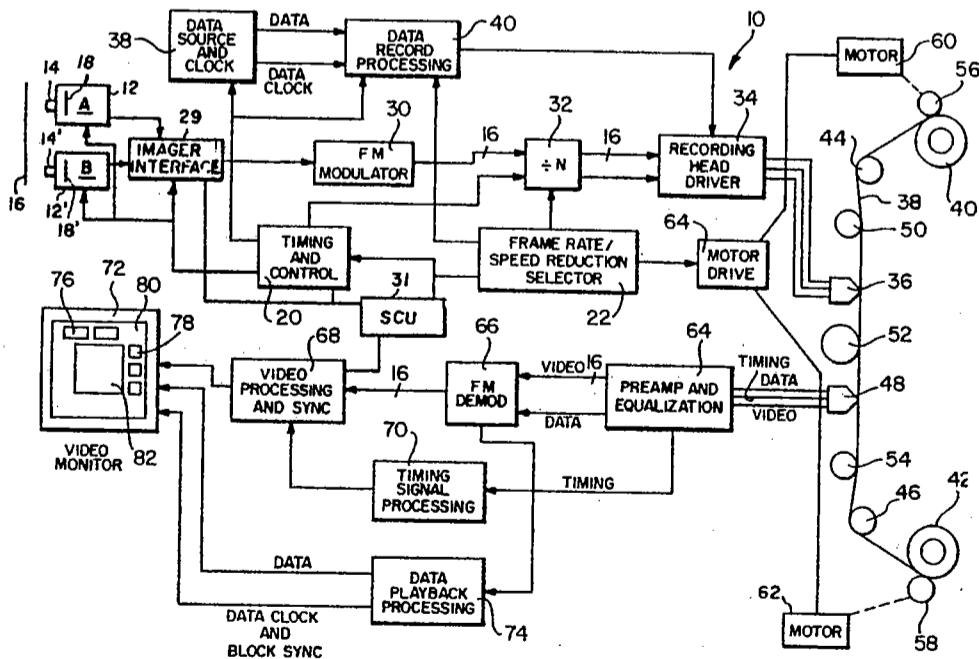
Assistant Examiner—John K. Peng

Attorney, Agent, or Firm—William F. Naval

## [57] ABSTRACT

A motion analyzer in which full frames of video information from two video imagers are recorded in interleaved fashion on magnetic tape. The motion analyzer records at multiple frame rates in a block sequential format in which blocks of parallel lines of video information are recorded sequentially on magnetic tape. Frames of video in block format from one imager are alternately recorded on magnetic tape with frames of video in block format from the other imager. Although the effective recorded frame rate for each imager is one-half of the analyzer frame rate, each recorded frame has been exposed at the analyzer frame speed.

3 Claims, 8 Drawing Sheets

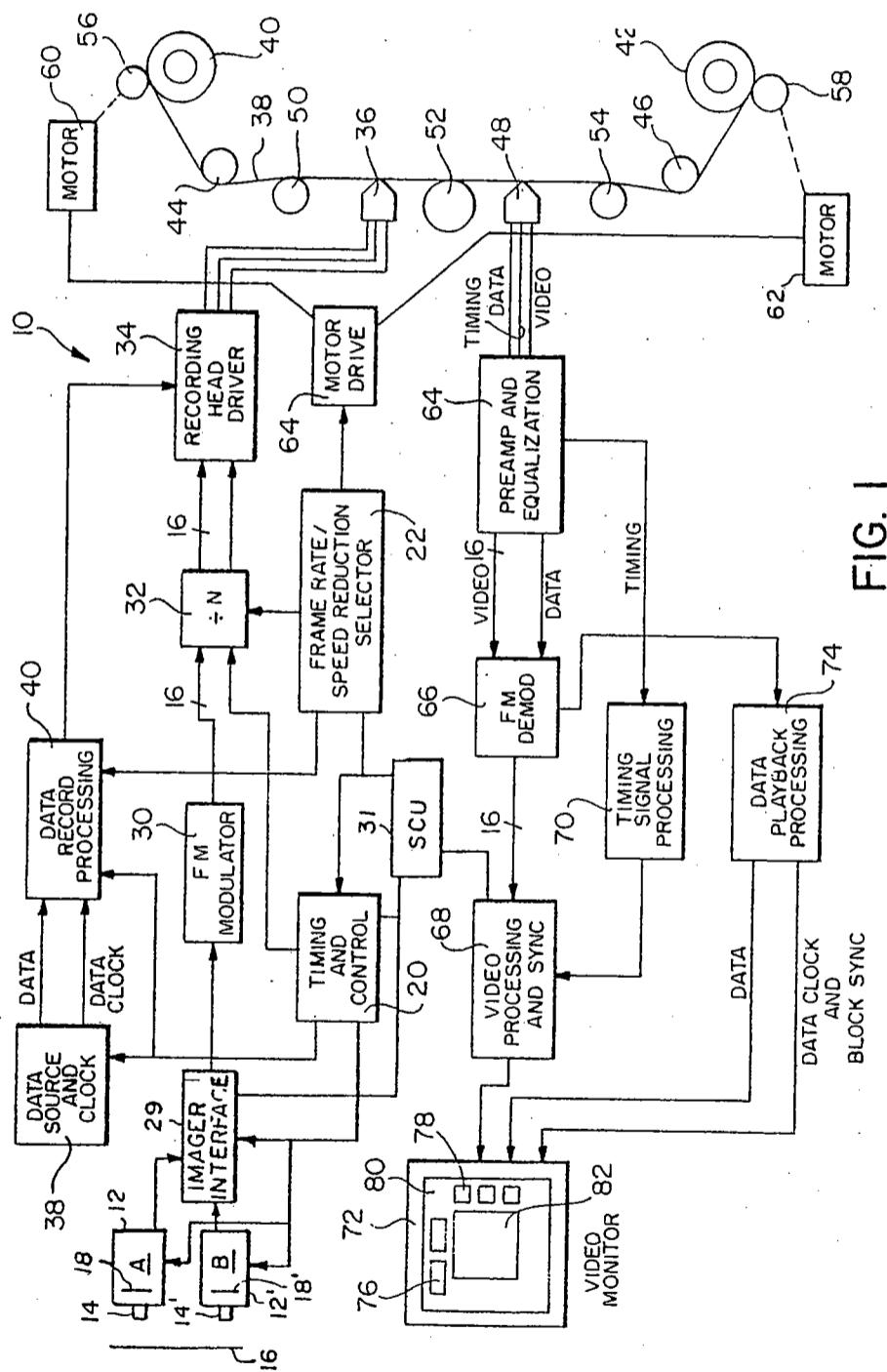


U.S. Patent

Dec. 6, 1988

Sheet 1 of 8

4,789,894



一  
五

U.S. Patent

Dec. 6, 1988

Sheet 2 of 8

4,789,894

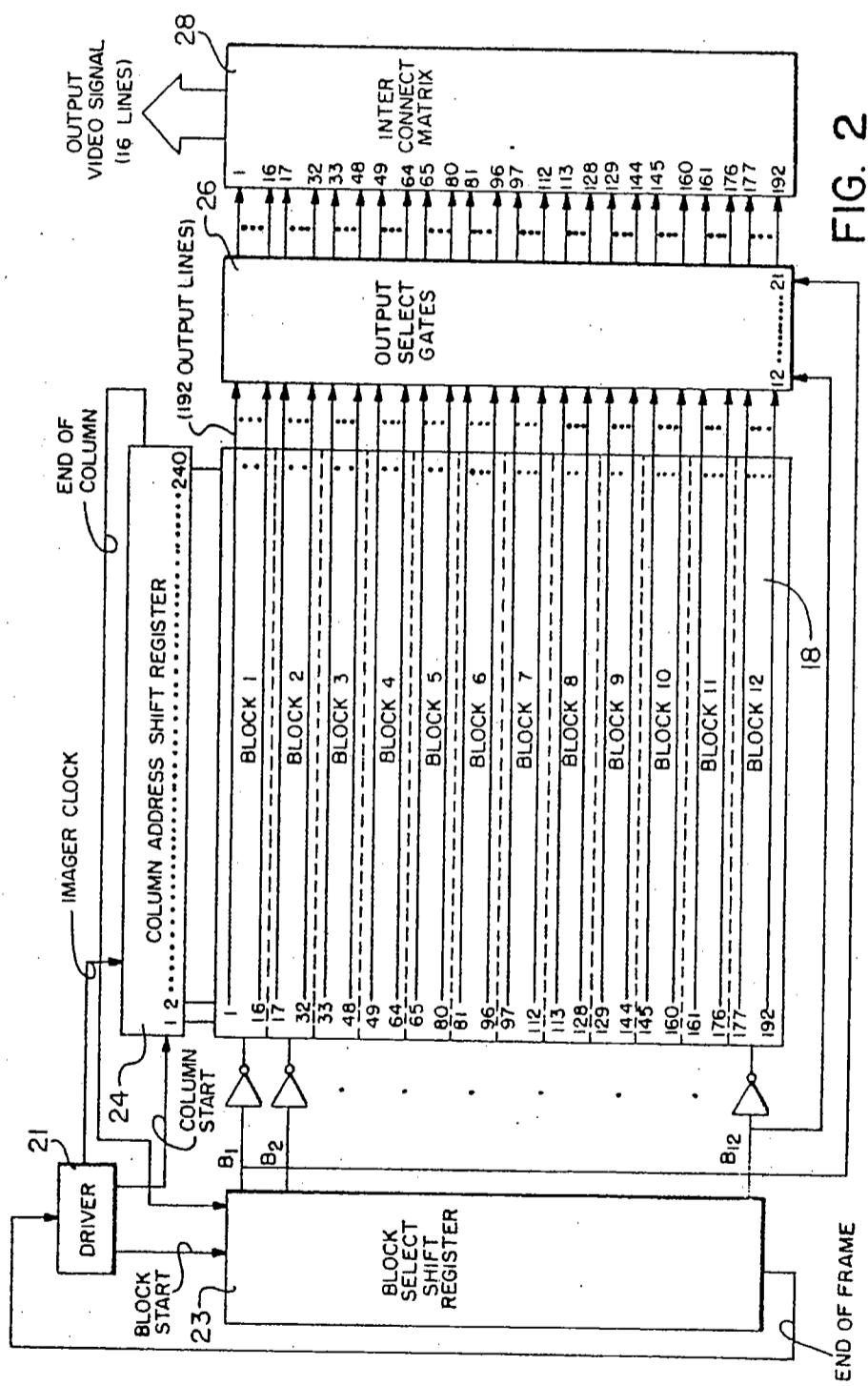


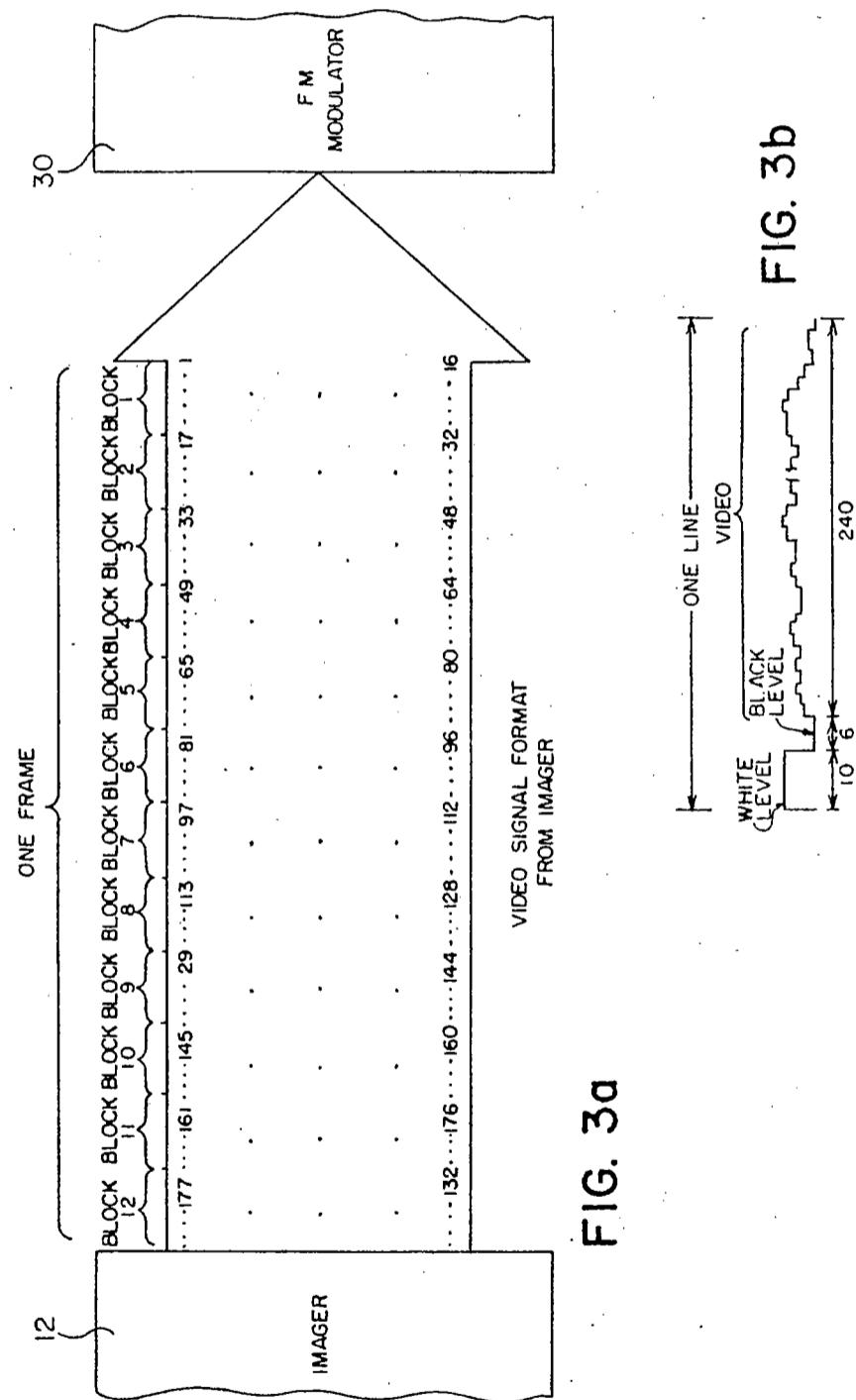
FIG. 2

U.S. Patent

Dec. 6, 1988

Sheet 3 of 8

4,789,894

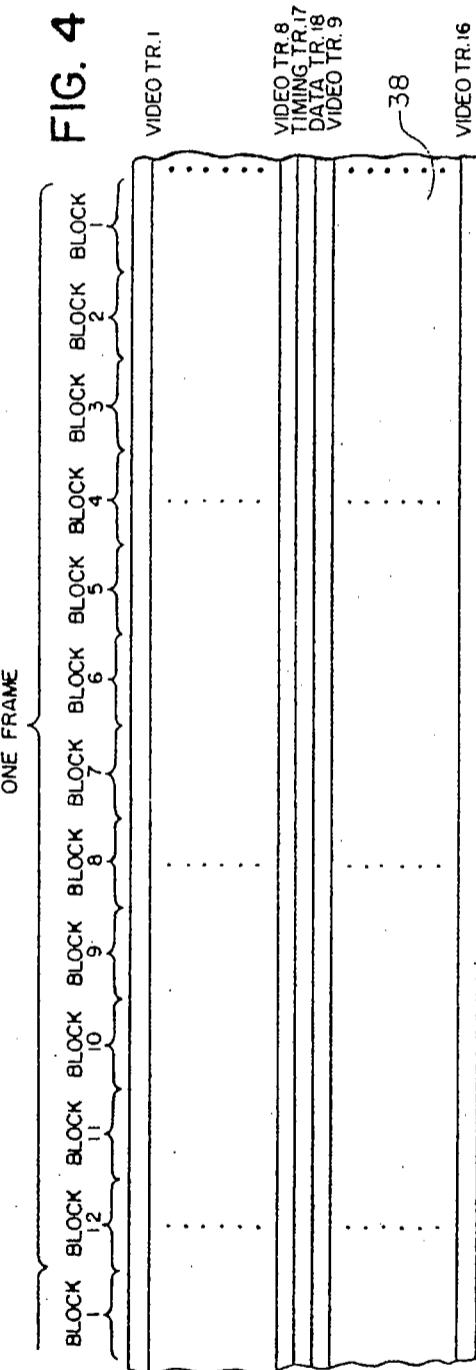


U.S. Patent

Dec. 6, 1988

Sheet 4 of 8

4,789,894



COL. 1 CAMERA AND RECORD FRAME RATE (FPS)	COL. 2 RECORD TAPE SPEED (IPS)	COL. 3 PLAYBACK FRAME RATE (FPS)	COL. 4 PLAYBACK TAPE SPEED (IPS)	COL. 5 N	COL. 6 SPEED REDUCTION R
30	7 1/2	30	7 1/2	33 1/3	1
60	15	30	7 1/2	16 2/3	2
125	3 1/4	30	7 1/2	8	4
250	62 1/2	30	7 1/2	4	8
500	125	30	7 1/2	2	16 2/3
1000	250	30	7 1/2	1	33 1/3

**FIG. 5**

U.S. Patent

Dec. 6, 1988

Sheet 5 of 8

4,789,894

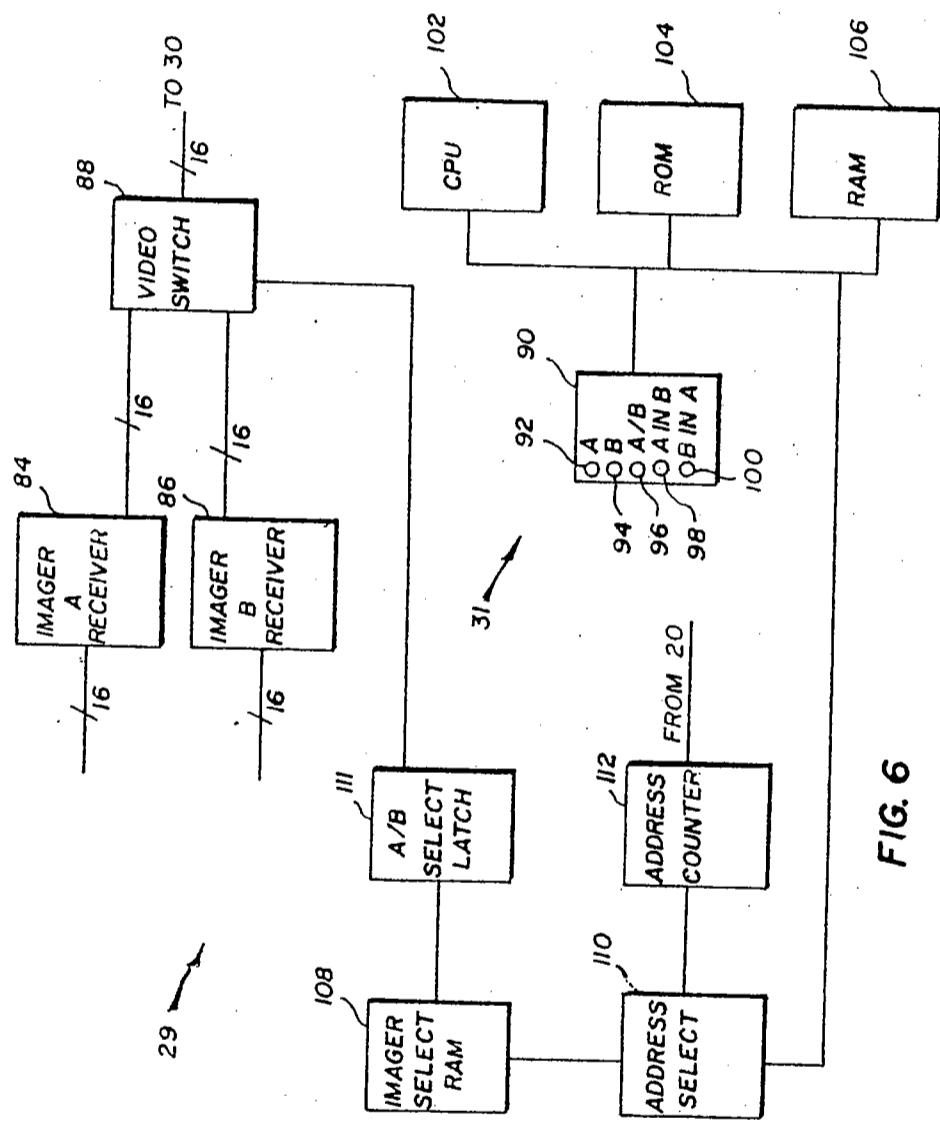


FIG. 6

U.S. Patent Dec. 6, 1988

Sheet 6 of 8

4,789,894

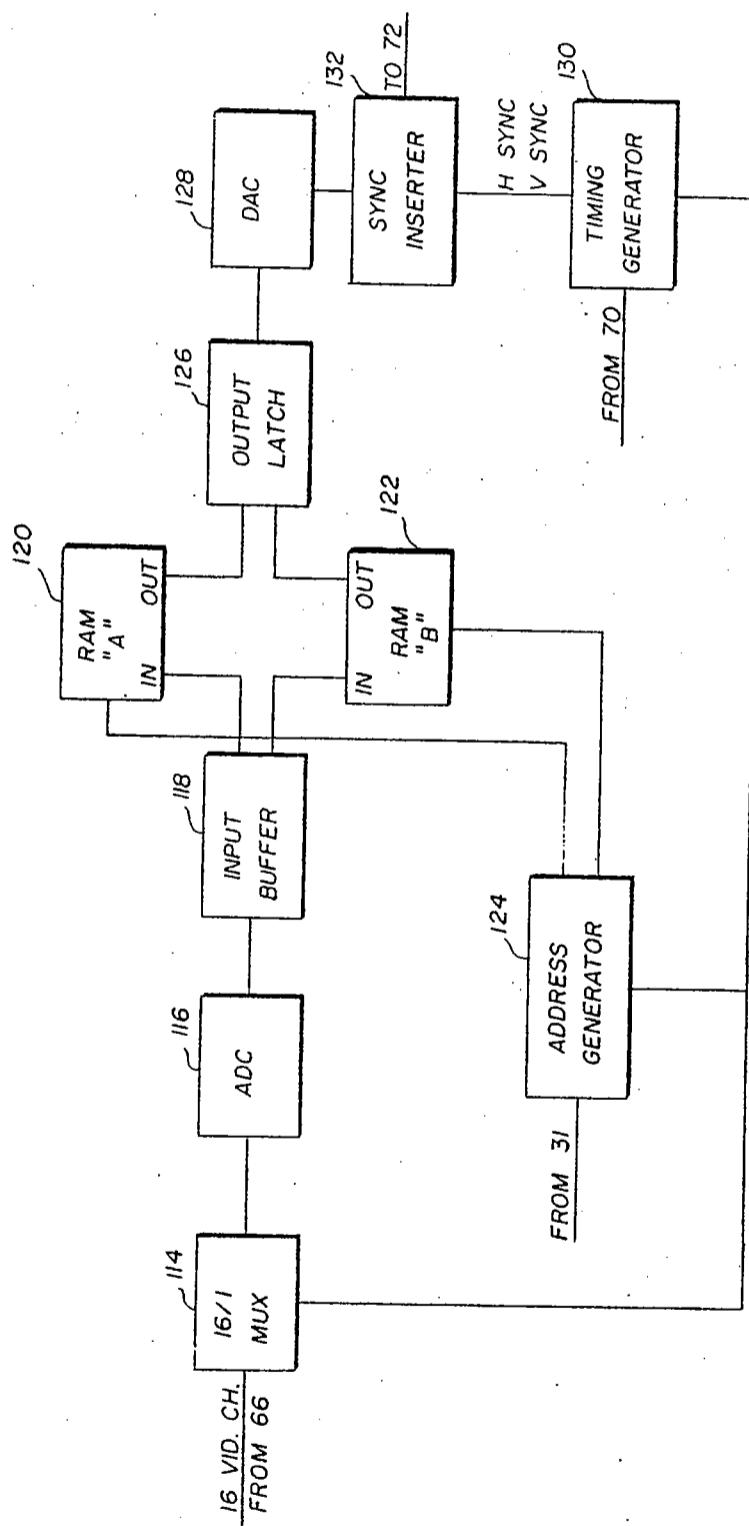


FIG. 7

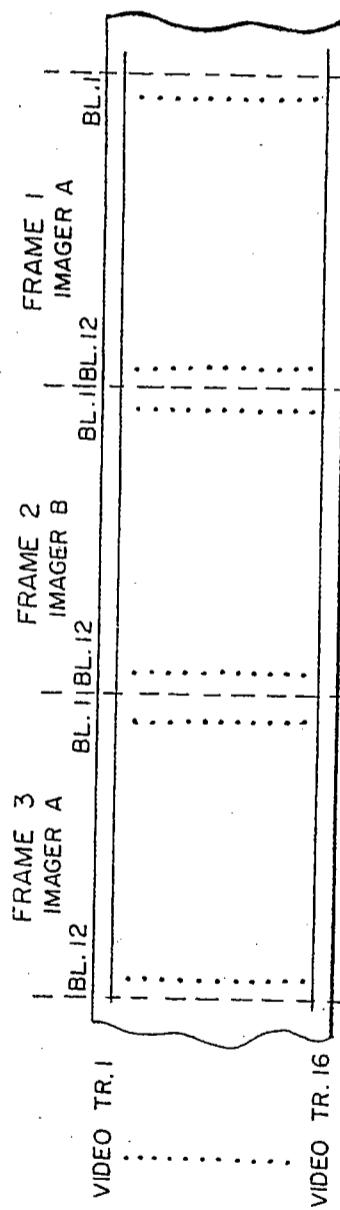


FIG. 8

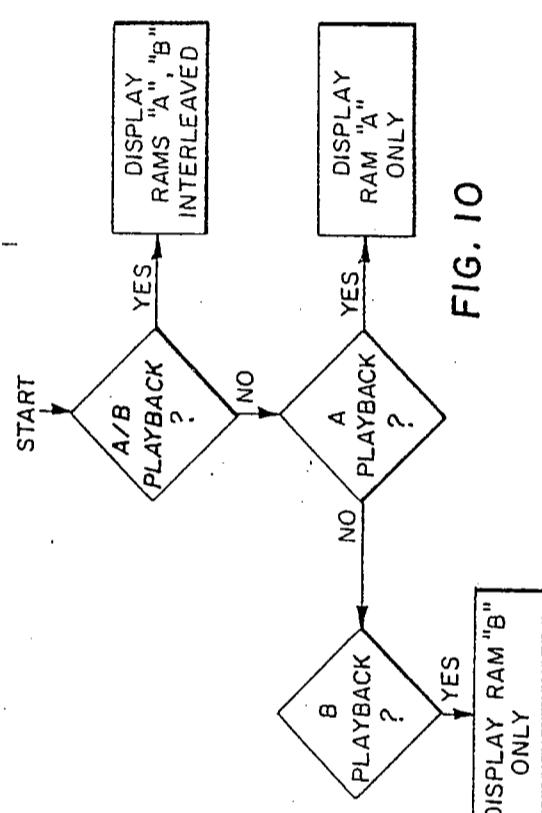


FIG. 10

U.S. Patent Dec. 6, 1988 Sheet 8 of 8 4,789,894

IMAGER COLUMN	IMAGER LINE	BLOCK	BLOCK LINE	DISPLAY "A" OR "B"
I-240	I-16	I	I-16	B
I-240	17-32	2	I-16	B
I-240	33-48	3	I-16	B
I-240	49-64	4	I-16	B
I-240	65-79	5	I-15	B
I-119	80	5	I6	B
I20-160	80	5	I6	A
I6I-240	80	5	I6	B
I-119	8I-96	6	I-16	B
I20-160	8I-96	6	I-16	A
I6I-240	8I-96	6	I-16	B
I-119	97-110	7	I-14	B
I20-160	97-110	7	I-14	A
I6I-240	97-110	7	I-14	B
I-240	III-112	7	I5-16	B
I-240	I13-128	8	I-16	B
I-240	I29-144	9	I-16	B
I-240	I45-160	10	I-16	B
I-240	I6I-176	II	I-16	B
I-240	I77-192	I2	I-16	B

FIG. 9b

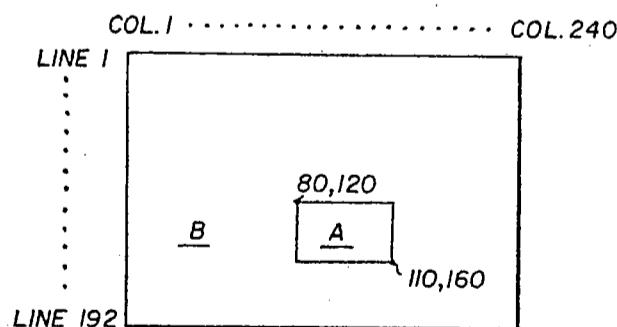


FIG. 9a

## MOTION ANALYZER WITH INTERLEAVED IMAGE REPRODUCTION

### BACKGROUND OF THE INVENTION

In general, this invention relates to motion analysis systems and more particularly, this invention relates to motion analysis systems in which images from two video imagers are recorded and reproduced in interleaved format.

Motion analysis of fast moving phenomena in slow motion entails the recording of a great number of images during an event at high speed, and then playing the images back slowly to analyze the movement which has occurred in step-by-step progression. Applications for motion analysis include malfunctions in high speed machinery, movements of an athlete, failure of safety equipment, shattering of an object and physical reactions to a tire hitting a pothole at high speed. One type of motion analyzer is the SP 2000 Motion Analysis System available from the Spin Physics Division of the Eastman Kodak Co. This system includes a video camera, a variable speed magnetic tape processor and a cathode ray tube (CRT) display monitor. The camera is capable of producing signals corresponding to selected frame rates of from 60 to about 2000 frames per second. The video is read out from the imager in block format, i.e., a plurality of lines of video simultaneously and is recorded in sequential blocks on a plurality of longitudinal tracks on tape. The magnetic tape processing system is capable of recording at one tape speed and appropriately slowing down the tape during playback to a certain predetermined speed to down convert the camera signals regardless of the camera frame rate, to a nominal frame rate of 30 or 60 frames per second. The CRT display monitor receives the second frame rate playback signal from the magnetic tape processing system and displays the scene in question at an appropriate slow motion, depending upon the selected camera frame rate.

The SP 2000 Motion Analysis System can record (1) all of the frames from one imager; (2) all of the frames from the other imager and (3) a frame which is a composite of part of a frame from one imager overlaid with a frame from the other imager. Although this system eminently satisfies the applications for which it is designed, there are certain applications where the need arises to record full frames from each imager simultaneously. The overlay technique does not satisfy this need since image areas from both frames are deleted in the recorded composite frame and it is sometimes desirable to have the deleted areas available for viewing. Using two motion analysis systems to effect recording of full frames from both imagers is costly and complex.

### SUMMARY OF THE INVENTION

The present invention solves these problems by providing a motion analyzer having two video imagers which are read out in block format such that frames from each imager are recorded alternately on magnetic tape in block sequential format. Thus, full frames from each imager are interleaved on magnetic tape and are selectively reproducible on a video monitor for analysis of the recorded event. Moreover, the frames from each imager are recorded at the selected frame rate exposure speed thus preserving the motion stopping ability selected for an event. According to an aspect of the invention, the recorded frames from one or the

other of the imagers may be played back or the images may be played back in an interleaved manner.

### DESCRIPTION OF THE DRAWINGS

5 In a detailed description of the preferred embodiments of the invention presented below, reference is made to the accompanying drawings in which like numerals refer to like elements.

10 FIG. 1 is functional block diagram of a motion analyzer including an embodiment of the present invention;

15 FIG. 2 is a functional block schematic diagram of a block readable area imager;

20 FIGS. 3a and 3b depict the format and content of a video signal produced by block readout of an area image sensor;

25 FIG. 4 shows the multitrack format of video information and digital data recorded longitudinally on magnetic tape;

30 FIG. 5 shows the relationship between various operational parameters of the motion analyzer shown in FIG. 1;

35 FIG. 6 is a block schematic diagram of the imager interface circuit and system control unit circuit of the analyzer of FIG. 1;

40 FIGS. 7 is a block schematic diagram of the video processing circuit of the analyzer of FIG. 1;

45 FIG. 8 is a diagram illustrating interleaving of frames of video information from two imagers recorded in block format on magnetic tape;

50 FIGS. 9a and 9b are a diagram and table respectively illustrating the overlay on a video monitor of video frames from two imagers; and

55 FIG. 10 is a flow chart for operating the analyzer of FIG. 1 in selected playback modes.

### DESCRIPTION OF THE PREFERRED EMBODIMENT

The application in which the embodiments of the present invention will be described relates to a motion analyzer which records scene information at a high frame rate and plays back such information at a slower frame rate, thereby allowing slow motion analysis of a moving object. The imager frame rate used for recording is variable between 30 and 1,000 frames per second, depending upon the desired speed reduction while the display frame rate is constant at 30 frames per second. Accordingly, the apparent speed at which an object moves when viewed upon playback will be reduced by a factor equal to the ratio of the recording frame rate to the playback frame rate. The maximum speed reduction is therefore about 33 (i.e. 1,000 divided by 30). At this speed reduction, the exposure time for each frame is 1/1000 of a second which is short enough to provide high resolution images, with very little image smear of rapidly moving objects.

To accomplish such frame rate conversion, the variable speed magnetic tape processor records and reproduces an imager signal with a recording tape speed to reproduction tape speed ratio that equals the ratio of the imager frame rate to the display frame rate. The magnetic tape processor operates in cooperation with a divide-by-N circuit that selectively alters the frequency content of the video signal to be recorded, and in a manner to be described in detail below, both the value of (N) and the ratio of the recording to playback tape speed are controlled by the selected speed reduction.

Referring to FIG. 1, there is shown a functional block, schematic diagram showing a motion analyzer including an embodiment of the present invention. The motion analyzer 10 includes "A" and "B" imagers 12 and 12' having lenses 14 and 14' which image a scene 16 onto sensors 18 and 18'. Imagers 12 and 12' are controlled by timing circuit 20 which supplies suitable timing signals to imagers 12 and 12' as a function of the operator selectable frame rate and speed reduction entered into by selector 22. The timing signals produced by circuit 20 are in accordance with the parameters tabulated in Columns 1 and 5 of FIG. 5. Thus, if a speed reduction of 8 (Col. 5) is selected, the imager will image scene 16 at a frame rate of 250 frames per second (Col. 1).

Sensors 18 and 18' are "block" readable area image sensors. The basic concept of a block readout of a solid state area image sensor is disclosed in U.S. Pat. No. 4,322,752 in the name of James A. Bixby which is incorporated herein by reference. Specific types of block readable sensors are disclosed in U.S. Pat. No. 4,322,638, issued Mar. 30, 1982 in the names of T. H. Lee and R. P. Khosla; and U.S. Pat. No. 4,330,796 in the name of C. N. Anagnostopoulos et al, both of which are herein incorporated by reference.

Although the referenced patents provide detailed information, the basic concept of block readout is illustrated in FIGS. 2 and 3. FIG. 2 shows a block readable sensor 18 (or 18') that includes an array of photosites (not shown individually) arranged in 192 rows and 240 columns. For purposes of readout, sensor 18 (18') is formatted into 12 blocks of 16 photosite rows each. Although demarcation between blocks is indicated by dashed lines, it will be understood that no physical demarcation on the sensor itself is necessary. Each photosite is readable upon the application thereto of an enablement signal and an address signal. To begin readout, a driver 21 produces a BLOCK START signal that causes a block select shift register 23 to produce an enablement signal that enables (via block enable line B<sub>1</sub>) all photosite rows within block 1, i.e. rows 1-16. In response to a COLUMN START signal from the driver 21 column address electronics in the form of a shift register 24 sequentially addresses the 240 photosite columns of the entire area image sensor 18. Because the photosite rows within blocks 2-6 (rows 17-192) are not enabled, only photosite rows 1-16 (block 1) are read out at this time. The remaining photosites in the not-enabled blocks continue to integrate charge in response to incident radiation from scene 16. After all columns have been addressed an END OF COLUMN signal sequences the block select shift register 23 to enable via block enable line B<sub>2</sub>, the block 2 photosite rows, i.e., rows 17-32. Column-wise readout then proceeds as described above for the block 1 photosite rows. This process is repeated until all 12 blocks of photosite rows are read out at which time END OF FRAME signal from block select shift register 23 resets driver 21 for readout of the next frame.

Output select gates 26 and an interconnect matrix 28 of conductive bus lines perform the function of a block multiplexer that causes only signals from the 16 photosite rows within the block that is being read out to appear as an output signal. Reference is made to U.S. Pat. No. 4,322,752 referenced above for a more detailed discussion of the construction of output select gates 26 and matrix 28.

As a result of such read out, block information is produced in series and each block of information contains 16 row signals arranged in parallel. A result of such a readout technique is a reduction of the time required for sensor readouts by a factor of 16 (i.e. the number of photosite rows in a block).

As shown in FIG. 3a, the video signal resulting from block readout of a single frame is comprised of a serial train of block information wherein each block is comprised of 16 lines of video information that correspond to the 16 rows of photosites within each block. Each individual line of video information (see FIG. 3a) is an analog signal varying in level proportionate to the level of scene illuminance, and each line contains 240 picture elements (pixels) that correspond respectively to the 240 photosites in each row of photosites. Although, as shown in FIG. 3b, only 240 pixels represent the active video information of a line of video information, each line is in reality 256 pixel periods in length with 16 pixel periods representing a black level reference signal of six pixel periods and a white level reference signal of 10 pixel periods.

As shown in FIG. 3a, each channel of information includes the video information of every 16th line of image sensor 18. Thus, the first video information channel includes lines 1, 17, 33, 49, 65, 81, 97, 113, 129, 145, 161, and 177, and the last video information channel includes lines 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, and 192.

Referring again to FIG. 1, the sixteen lines of signals from each of imagers 12 and 12' are processed in imager interface circuit 29 in accordance with image format signals received from system control unit (SCU) 31. As will be described in more detail below, interface circuit 29 selectively processes the block signals from imagers 12 and 12' to produce an output of 16 lines of video signals. Each of the 16 line signals that constitutes the analog video signal from circuit 29 is frequency modulated in an FM modulator circuit 30 on a carrier.

The frequency modulated video signals undergo a divide-by-N process in a divide-by-N circuit 32. A suitable divide-by-N circuit which may be adapted to the apparatus shown in FIG. 1 is illustrated in FIG. 7 of commonly-assigned U.S. Pat. No. 4,496,995 issued Jan. 29, 1985, by J. H. Colles et al. The value of "N" is equal (to the nearest integer) to the maximum selectable speed reduction divided by the selected speed reduction. The relationship between "N" and various values of the speed reduction "R" is given in columns 5 and 6 of FIG. 5. It will be noted that the selected speed reduction has been used to determine both the frame rate at which sensor 18 is read out and the value of "N" in the divide-by-N circuit 32. While it may not be apparent how these two parameters (frame rate and "N") relate to slow motion replay, the discussion which follows will show that the selection of these two parameters as described above in conjunction with the proper selection of a third parameter (recording tape speed) will produce the desired speed reduction of scene information upon playback.

A timing signal from timing circuit 20 is also applied to circuit 32 to be divided by the same factor "N" as the FM video signals.

The output of the divide-by-N circuit 32 is comprised of seventeen frequency divided frequency modulated signals. These signals are applied to a recording head driver circuit 34 that drives an 18 channel magnetic recording head 36. Channel 18 of the recording head is

used for recording digital data. The digital data is produced by a data source 38 which produces digital data signals which are processed by data record processing circuit 40. Source 38 also provides data clock signals which are in synchronism with the digital data signals to data recording processing circuit 40. The processed data signal is supplied to recording head driver 34 and then to the magnetic recording head for track 18 in multihead 36.

The 18 signals are recorded along 18 separate channels or tracks on magnetic tape 38. Magnetic tape 38 is provided in a cassette (not shown) having supply reel 40, takeup reel 42, and tape guides 44 and 46. When the cartridge is inserted into apparatus 10, tape 38 is pressed against recording head 36 and reproducing head 48 as well as external guides 50, 52, and 54. Tape is advanced from reel 40 to reel 42 by means of capstans 56 and 58 respectively driven by capstan motors 60 and 62, controlled by motor drive 64.

The speed at which magnetic tape 38 is advanced during recording is selected to be proportional to the selected speed reduction and frame rate of recording. The relationship of the record frame rate, record tape speed and speed reduction is tabulated in columns 1, 2, and 6, respectively of FIG. 5. For example, if a record frame rate of 250 frames per second (Column 1) is chosen with a speed reduction R of 8 (column 6), then the magnetic tape 38 would be advanced at a speed of  $62\frac{1}{2}$  inches per second (Column 2).

Upon recording, the signals retain the block format (as shown in FIG. 4) in which a timing track 17 and a data track 18 are located between video tracks 1-8 and video tracks 9-16.

Having recorded information on magnetic tape 38 that corresponds to a scene including an object under study, a slow motion video display of such object is produced by playing back the recorded information at a constant tape speed of say,  $7\frac{1}{2}$  inches per second irrespective of the originally selected recording tape speed. As a result, the ratio of the recording tape speed to the playback tape speed yields a tape speed reduction ratio that equals the selected speed reduction. Further, all reproduced signals have the same black to white level frequency spread, thereby enabling a fixed frequency demodulator to be used irrespective of the selected speed reduction. To understand why the above-described selection of recording frame rate, the factor "N", recording tape speed and playback tape speed results in the desired speed reduction and signal form upon playback, reference is made to FIG. 5. It is apparent from inspection of FIG. 5 that "N" is equal to the maximum selected camera (record) frame rate divided by the selected camera frame rate. The effect of the divide-by-N circuit 32 is to reduce the black level to white level frequency spread by a factor "N". But because the ratio of record tape speed to playback tape speed varies in inverse proportion to "N", all signals produced upon playback have the same black level to white level frequency spread. Further, because the playback tape speed is always  $7\frac{1}{2}$  inches per second, all video information is reproduced at a frame rate of 30 frames per second, thereby resulting in the desired speed reduction.

Referring again to FIG. 1, the sixteen video signals produced by playback head 48 undergo signal processing in a preamplification and equalization circuit 64. The processed signals are then demodulated in an FM demodulator 66. After demodulation, the video signals

(which are still in the block format shown in FIG. 3a) are converted to a line sequential video signal by a video processing circuit 68 (see FIG. 7).

A timing signal reproduced from timing track 17 is processed by circuit 64 and is applied to timing signal processing circuit 70 which extracts suitable timing and sync signals which are used in video processing circuit 68 to produce a signal to be displayed on monitor 72. The displayed scene information consists of a slow motion replay of the originally recorded scene at the selected speed reduction.

Data from data track 18 is reproduced by reproducing head 48 and preamplified and equalized in circuit 64. The data signal is then processed by data playback processing circuit 74 to be shown in window areas such as 76, 78 in a data frame 80 surrounding the main image area 82 of monitor 72.

Referring now to FIGS. 6-10, there will be described a preferred embodiment of the image interleave technique of the present invention. As shown in FIG. 6, the 16 lines of video information from imagers A(12) and B(12') are received simultaneously by imager receivers 84 and 86 respectively when both imagers A (12) and B (12') are imaging a scene 16 at the same frame rate. Receivers 84 and 86 supply the 32 lines of video information from imagers A (12) and B (12') to video switch 88 which is controlled to selectively pass one of the inputs A or B on each of 16 video lines to a 16 line video output which is applied to FM modulator 30. Switching control of video switch 88 is effected in accordance with the mode of recording selected by an operator. According to the present invention, the motion analysis system described is operable to record all of the frames of imager A (12), all of the frames of imager B (12'), interleaving of the frames of imager A (12) with the frames of imager B (12'), overlay of the frames from imager A on the frames from imager B, and overlay of the frames from imager B (12') on the frames from imager A (12).

To assist in selecting the desired mode of operation of the motion analyzer, system control unit 31 is illustratively shown as including a control unit 90 having manually operable switches 92, 94, 96, 98 and 100 to effect the desired image recording mode. If switch 92 is actuated, all of the frames from imager A (12) are recorded at the selected frame rate on magnetic tape 38. If switch 94 is actuated, all of the frames from imager B (12') are recorded at the selected frame rate. If switch 96 is actuated according to the present invention, the frames from imager A (12) are interleaved with the frames from imager B (12') at the recorded frame rate. Thus, every other frame from imager A (12) and every other frame from imager B (12') are recorded, effectively cutting the frame rate for each imager to half the selected frame rate. However, the exposure speed of each frame that is recorded is at the frame rate speed. Thus, for example if each imager is operated at 1,000 frames per second, each frame has an exposure of 1/1000 sec. In the interleave mode, only 500 frames from each imager will be recorded at the 1,000 frames per second frame rate, but each frame that is recorded will be imaged at 1/1000 sec. exposure.

Switch 98 is actuated when a frame from imager A (12) is to be overlaid on a frame from imager B (12'). Actuation of switch 100 causes a frame from imager B (12') to be overlaid on a frame from imager A (12).

SCU 31 includes a central processing unit (CPU) 102, read only memory (ROM) 104, and random access

memory (RAM) 106. CPU 102 may for example, be a microprocessor which is operated in conjunction with ROM 104 and RAM 106 to receive and transmit data and instructions to operate the various instrumentalities of the motion analyzer. The operation and programming of a microprocessor is well known to those skilled in the art, and is, for example, explained in the Harvard textbook, "The Art of Electronics", by Horowitz and Hill, Cambridge University Press, Cambridge, 1980, Chapter 11, entitled "Microprocessors", p. 484 and following. SCU 31 in accordance with the image mode selected loads information into imager select RAM 108 by means of address select circuit 110. This information defines the content of each recorded frame of video information on magnetic tape 38. Information is read out of imager select RAM 108 by address counter 112 through address select circuit 110 that advances column by column through blocks of video information in a frame. The output of RAM 108 is applied to A/B select latch 111 which supplies 16 select signals to video switch 88 to select which of the video signals from imager A (12) or imager B (12') is to be outputted for each of the 16 channels to FM modulator 30. If the imager A mode is selected then the data signals from select latch 110 actuate video switch 88 so as to cause all 16 channels of video information from imager A (12) to be outputted to modulator 30. If imager B output is selected, then the data signals from select latch 110 causes video switch 88 to output all 16 channels of video information from imager B to FM modulator 30.

If the interleave mode is selected, then the gating signals applied to video switch 88 from A/B select latch 110 gates through all 16 channels from imager A (12) for one frame of video and all 16 channels from imager B (12') for the next frame. This process is repeated so that alternate frames of imager A (12) and imager B (12') are recorded on tape 38. This is illustrated more clearly in FIG. 8 wherein 3 frames of video information are shown recorded on tape. Frame 1 includes blocks 1-12 of 16 lines of video information from imager A (12), frame 2 includes blocks 1-12 each of 16 lines of video information from imager B (12'), and frame 3 includes blocks 1-12 of 16 lines each of video information from imager A (12), etc.

In either of the overlay modes, the size of the overlay region will determine the pattern of video information from imagers A (12) and B (12') to be recorded. As an example, reference is made to FIGS. 9a and 9b in conjunction with the imager shown in FIG. 2. As shown in FIGS. 9a and 9b, an image from imager A (12) is overlaid on an image from imager B (12'). The area of image A includes the pixels in lines 80-110 from columns 120-160. The boundaries of the image to be overlaid is entered into SCU 31 by suitable control (not shown). SCU 31 will then load the address information in imager select RAM 108 to produce the 16 gate select signals to control video switch 88. For example, in the example of FIGS. 9a and 9b, columns of video information are read out block by block from imagers A (12) and B (12'), gate select signals will be applied to video switch 88 to pass through the video information from one of the images according to the table shown in FIG. 9b. For example, as block 6 is read out column by column, for columns 1-119, switch 88 will be actuated to pass through video information from imager B (12') on all 16 lines; for columns 120-160, information from imager A (12) on all 16 lines; and for columns 161-240, video information from imager B (12') on all 16 lines.

Referring now to FIG. 7, there will be described in greater detail video processing circuit 68 of FIG. 1. As described above, FM demodulator 66 produces 16 simultaneous channels or lines of video information which are supplied to video processing circuit 68. In order to display this video information on a standard video monitor, the simultaneous lines of video information must be converted into line sequential information. According to the present invention, if frames of video information from two imagers have been recorded in an interleave mode on magnetic tape 38, upon playback of the tape, either all of the imager A frames may be displayed or all of the imager B frames may be displayed, or the imager A frames and imager B frames may be displayed in an interleaved fashion. To effect these playback modes, video processing circuit 68 includes a 16:1 multiplexer (MUX) 114, analog to digital converter (ADC) 116, input buffer 118, random access memories (RAM) 120 and 122, address generator 124, output latch 126, digital to analog converter (DAC) 128, timing generator 130 and sync inserter circuit 132. Multiplexer 114 is controlled by a timing signal which is 16 times as fast as the signal rate of the 16 lines of video information derived from demodulator 66. Multiplexer 114 thus sequentially passes through the video information appearing on each of the 16 video input channels to the output line at a rate which is 16 times faster than the input rate. The multiplexer 114 thus samples all 16 input video lines before the pixel information corresponding to column 2 of a block appears on the input lines.

The output signal from multiplexer 114 is applied to ADC 116 which converts the analog pixel signal to a digital pixel signal which is loaded into input buffer 118. Address generator 124 generates address signals for storing the digital information in input buffer 118 in either "A" RAM 120 or "B" RAM 122. Address generator 124 receives signals from SCU 31 to generate sequential addresses for storing the digital information either in RAM 120 or RAM 122, depending upon whether the frame of video information to be stored has been generated by imager A or imager B.

If the operator has selected interleave playback mode, then as a frame from imager A is written into RAM 120, a frame from imager B is read out of RAM 122. During the next frame period, a frame of video information from imager B is written into RAM 122 and a frame of video information from imager A is read out from RAM 120. It will be appreciated that, whereas the pixels of video information written into either RAM 120 or RAM 122 is sequenced so that column 1 of lines 1-16 is followed by column 2 of lines 1-16 and so on to column 240 of lines 1-16, the pixels of video information are read out of RAMs 120, 122 in line sequential format so that, for example, line 1, columns 1-240 are read out first, then line 2, columns 1-240, etc.

The digital data read out of RAM 120 and RAM 122 is loaded into output latch 126 and converted into an analog video signal by DAC 128. Timing generator 130 produces H sync and V sync signals which are combined in sync inserter 132 with the analog video information from DAC 128 to produce line sequential video to be displayed on monitor 72. If a playback mode is selected in which either all of the frames of imager A or all of the frames of imager B are played back on monitor 72, then the address signals produced by address generator 124 will cause only frames from either RAM 120 or RAM 122 to be played back. FIG. 10 illustrates a pro-

gram for controlling playback in accordance with the playback mode selected by an operator.

The invention has been described in detail with particular reference to preferred embodiments thereof, but it will be understood that variations and modifications can be effected within the spirit and scope of the invention. Imagers A and B may be read out with different numbers of video lines per block. For example, in the pixel format described above, imagers A and B could be read out in blocks of 6 with 32 video channels per block. In such case, the various timing signals will be adjusted according to the pixel matrix size of the imagers, and according to the block format and frame rate chosen.

What is claimed is:

1. A motion analysis system comprising:  
a first motion video imager;  
a second motion video imager;  
means for continuously reading out each of said video imagers simultaneously at the same frame rate of F frames per second, where F is selectively variable, in a block format in which a frame of video information comprises a sequence of blocks of parallel lines of video information which have been read out simultaneously; and  
means for recording every other frame from each of said first and second imagers in interleaved fashion

on the same magnetic tape, each of said frames on magnetic tape being recorded in a plurality of parallel recording tracks in a sequence of blocks of parallel lines of video information.

2. The system of claim 1 including selectively controllable means for playing back on a video monitor frames of video information from said magnetic tape (1) in a first mode, wherein only frames from said first imager are played back; (2) in a second mode, wherein frames from said second imager are played back; and (3) in a third mode, wherein frames from said first and second imagers are played back in interleaved fashion.

3. The system of claim 1 including imager interface circuit means for receiving video information simultaneously from said first and second video imagers and for selectively providing to said recording means for recording on said magnetic tape, one of the following sequence of frames (1) all of the frames from said first imager; (2) all of the frames from said second imager; (3) every other frame from each of said first and second imagers in interleaved fashion; (4) overlay of the frames from said first imager on the frames from said second imager; or (5) overlay of the frames from said second imager on the frames from the first imager.

\* \* \* \* \*



US005359468A

**United States Patent [19]**

Rhodes et al.

**Patent Number: 5,359,468****Date of Patent: Oct. 25, 1994****[54] DIGITAL DATA STORAGE TAPE FORMATTER****[75] Inventors:** Edward J. Rhodes; Daryl K. Adams, both of San Jose; Clifford R. Berry, Fremont; Ta-Lin Chang, Cupertino; David R. Lee, San Jose, all of Calif.**[73] Assignee:** R-Byte, Inc., San Jose, Calif.**[21] Appl. No.:** 740,755**[22] Filed:** Aug. 6, 1991**[51] Int. Cl.<sup>5</sup>** G11B 5/09**[52] U.S. Cl.** 360/48; 360/32; 360/53; 360/78.07; 360/78.15**[58] Field of Search** 360/48, 32, 46, 53, 360/64, 78.07, 78.15**[56] References Cited****U.S. PATENT DOCUMENTS**

- 3,838,453 9/1974 Buslik et al. .  
3,860,960 1/1975 Akamine .  
3,883,785 5/1975 Fulcher et al. .  
4,060,840 11/1977 Umeda .  
4,258,399 3/1981 Iijima et al. .  
4,408,235 10/1983 Ito .  
4,410,919 10/1983 Umeda .  
4,459,625 7/1984 Kawase .  
4,510,534 4/1985 Maeda .  
4,541,024 9/1985 Seiben .  
4,599,561 7/1986 Takahashi et al. .  
4,740,850 4/1988 Ogata .  
4,742,407 5/1988 Smith et al. .  
4,758,398 7/1988 Nishida et al. .  
4,800,450 1/1989 Rodal et al. .  
4,833,549 5/1989 Yoshimoto et al. .  
4,835,629 5/1989 Ueno et al. .

- 4,873,589 10/1989 Inazawa et al. .  
4,891,644 1/1990 Noro .  
4,899,233 2/1990 Yoshida .  
4,910,448 3/1990 Tomisawa et al. .  
4,939,595 7/1990 Yoshimoto et al. .

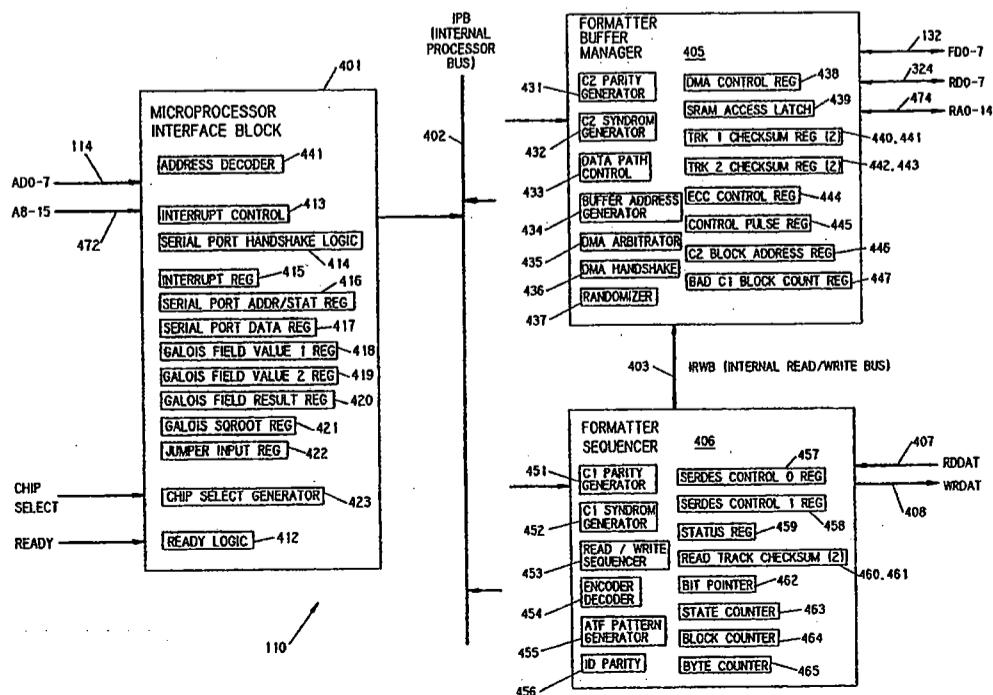
**OTHER PUBLICATIONS**

"Flexible Magnetic Media for Digital Data Interchange", Secretariat: U.S.A. (ANSI), ISO/IEC JTC 1/SC 11 N 1026, 1990-07-13.

*Primary Examiner*—Robert J. Pascal  
*Assistant Examiner*—Haissa Philogene  
*Attorney, Agent, or Firm*—Lyon & Lyon

**[57] ABSTRACT**

In a digital data storage audio tape system having a host device interface unit, a main data buffer manager, a main data buffer, a processor, a data separator, a tape drive unit, and a read/write channel, this invention comprises a formatter providing track construction, encoding and decoding, error correction, and direct memory access of digital data transmitted between the host device unit and the tape drive unit and a frame buffer for buffering digital data for encoding and decoding, and error correction. The formatter is coupled between the main data buffer manager, via a direct memory access channel, and the read/write channel of the tape drive, and further is coupled to the frame buffer. The formatter is coupled to and responsive to instruction control by the processor in conjunction with instruction control of the tape drive unit by the processor.

**38 Claims, 13 Drawing Sheets****Microfiche Appendix Included  
(16 Microfiche, 93/2 Pages)**

U.S. Patent

Oct. 25, 1994

Sheet 1 of 13

5,359,468

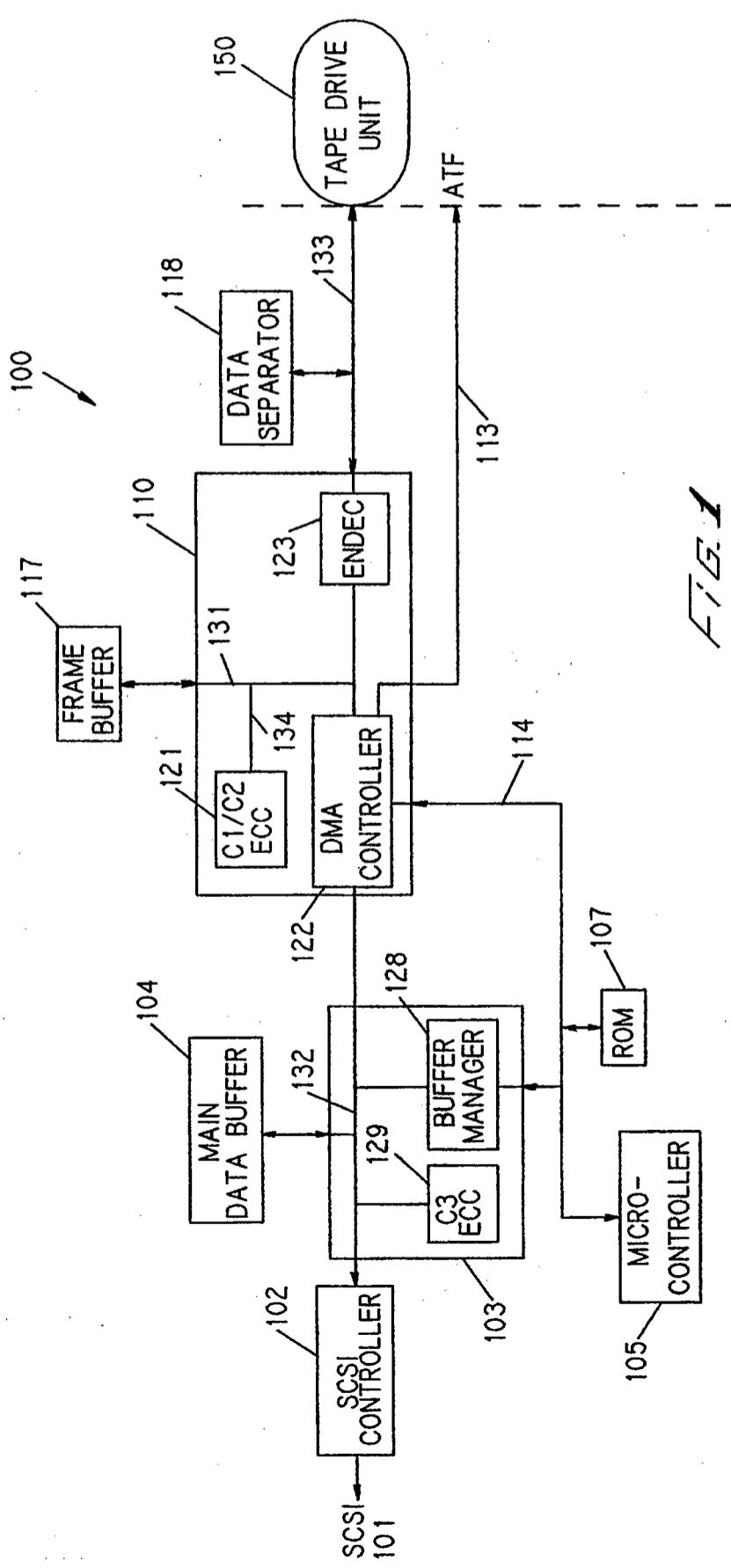


FIG. 1

U.S. Patent

Oct. 25, 1994

Sheet 2 of 13

5,359,468

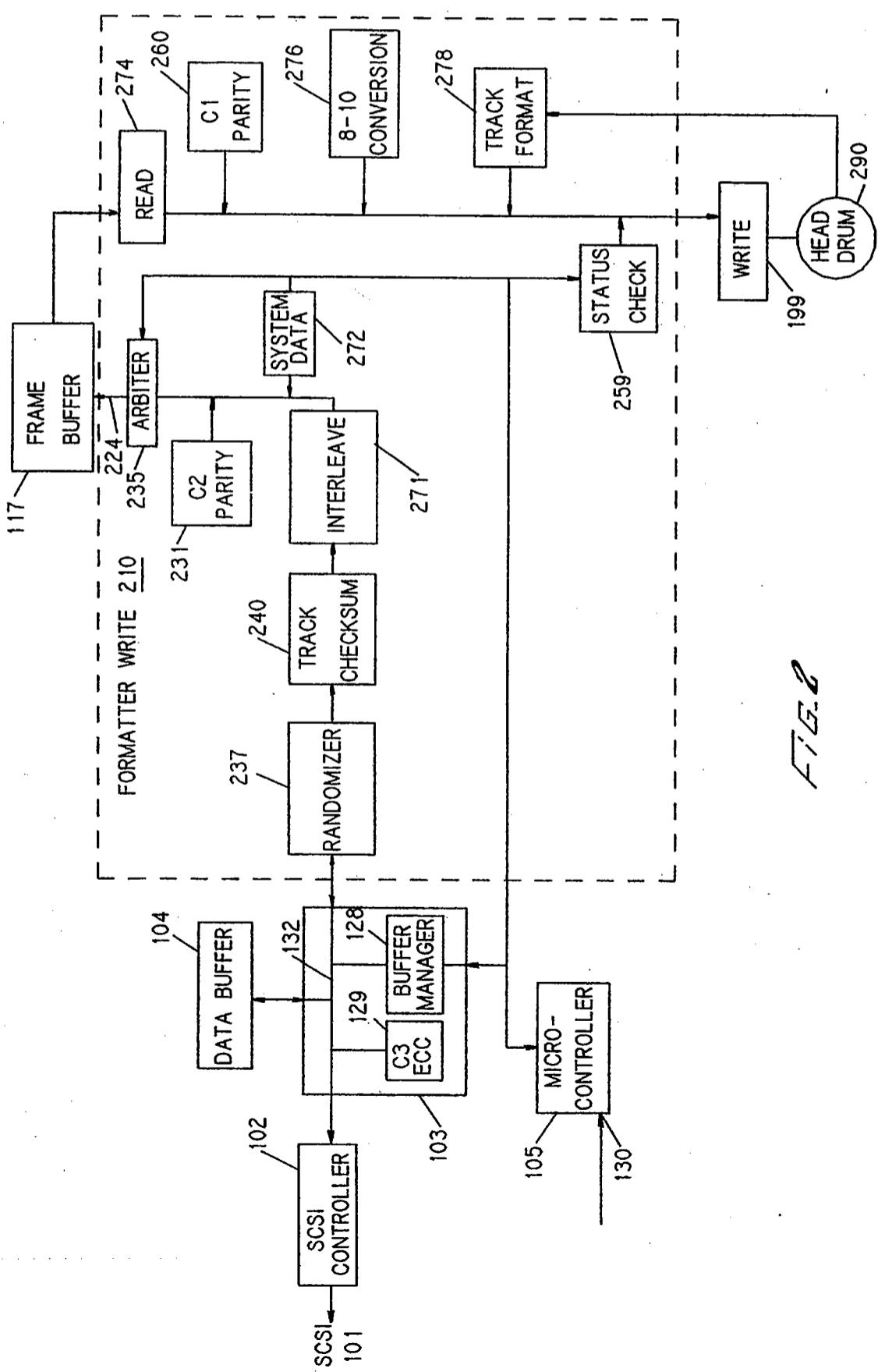


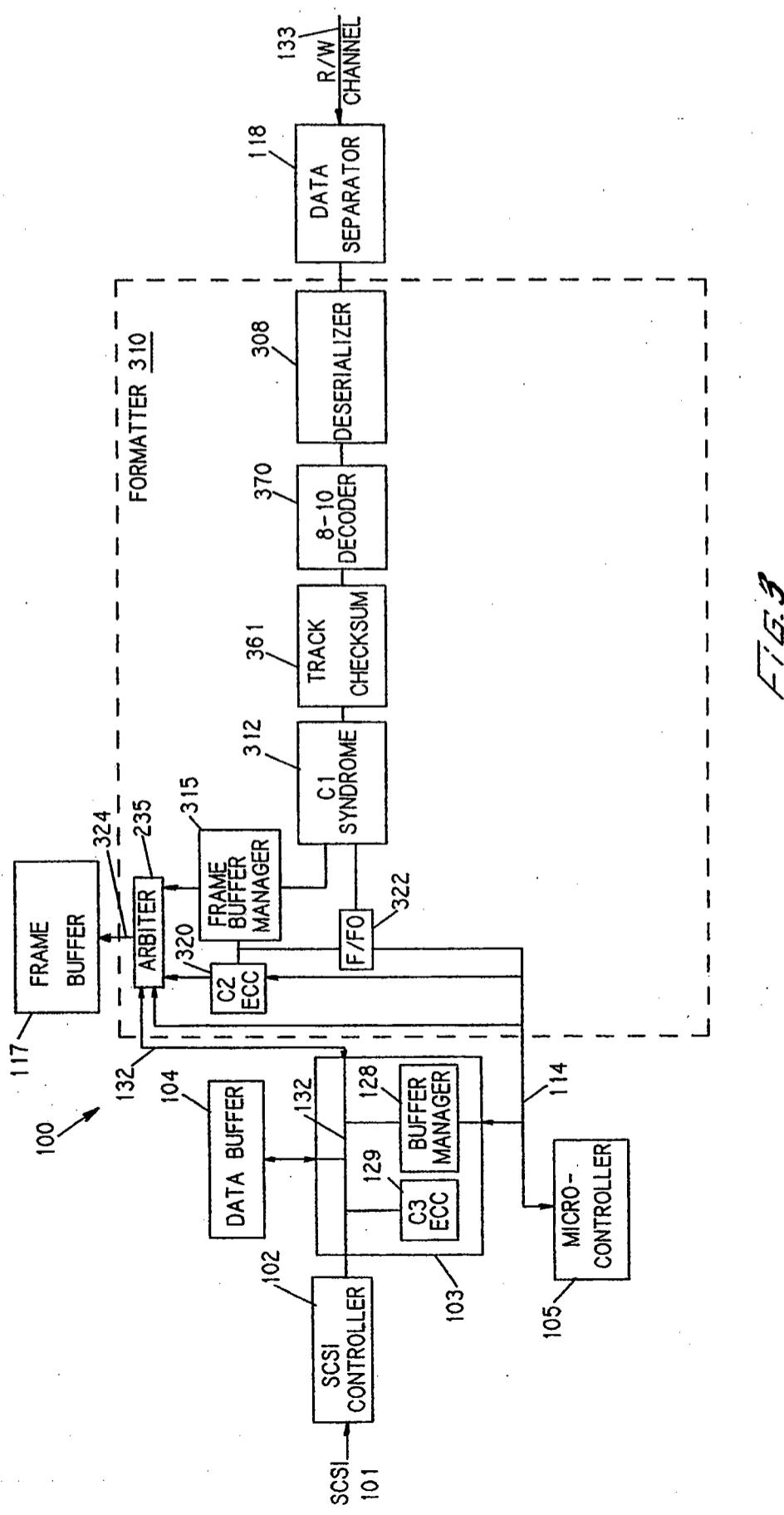
FIG. 2

U.S. Patent

Oct. 25, 1994

Sheet 3 of 13

5,359,468

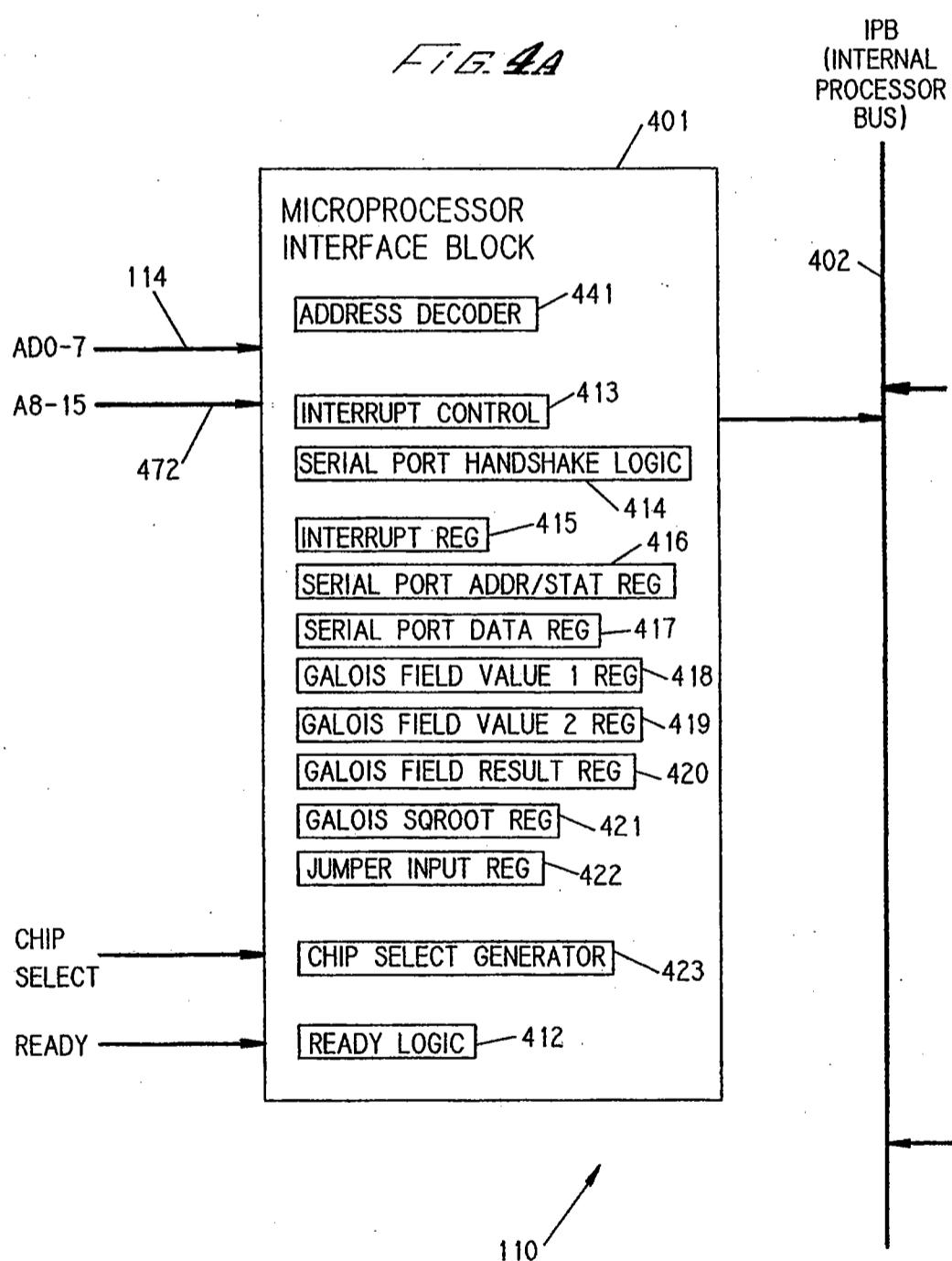


U.S. Patent

Oct. 25, 1994

Sheet 4 of 13

5,359,468



*FIG. 4*

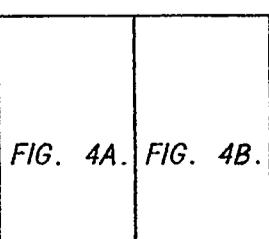
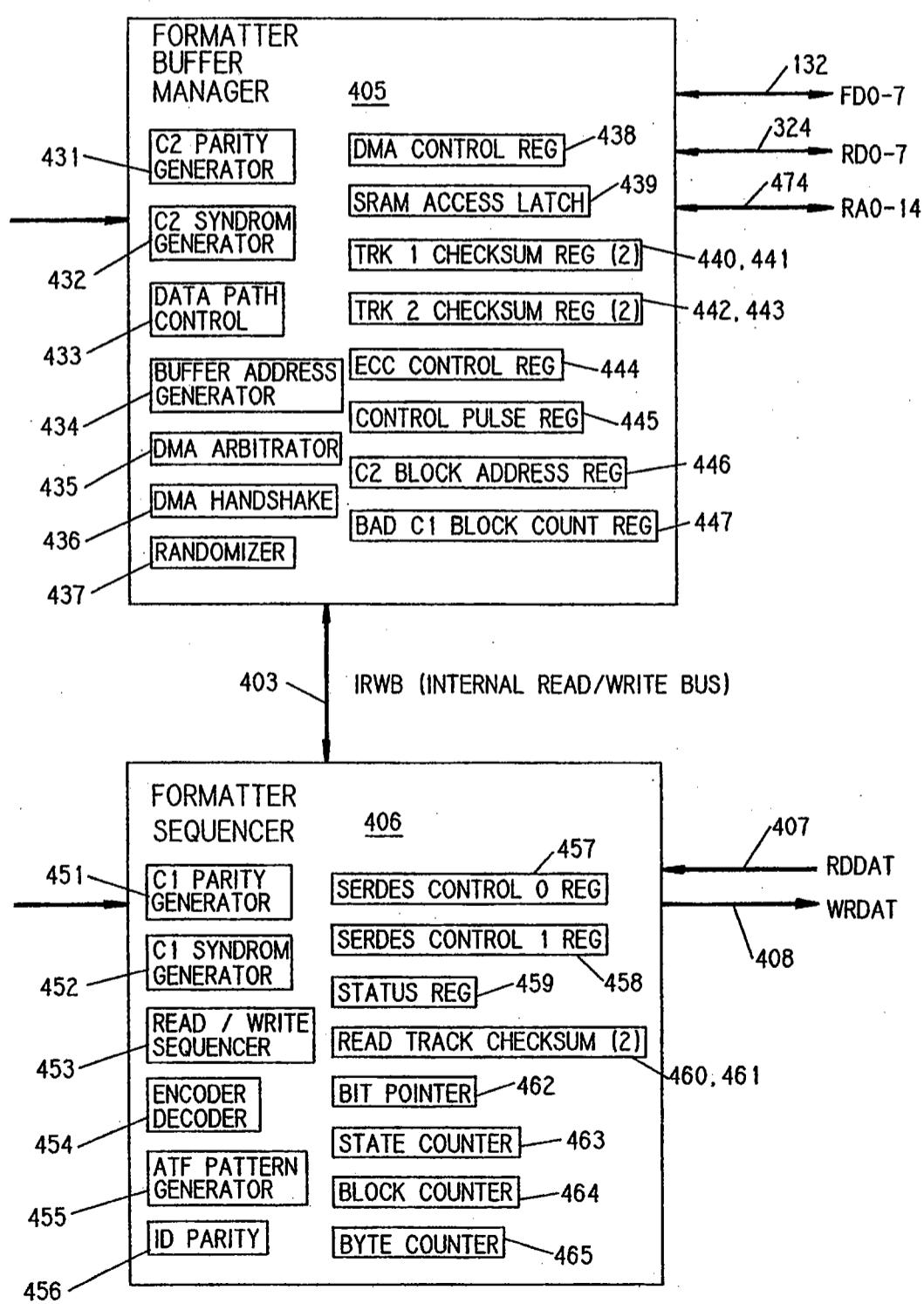


FIG. 4B



U.S. Patent

Oct. 25, 1994

Sheet 6 of 13

5,359,468

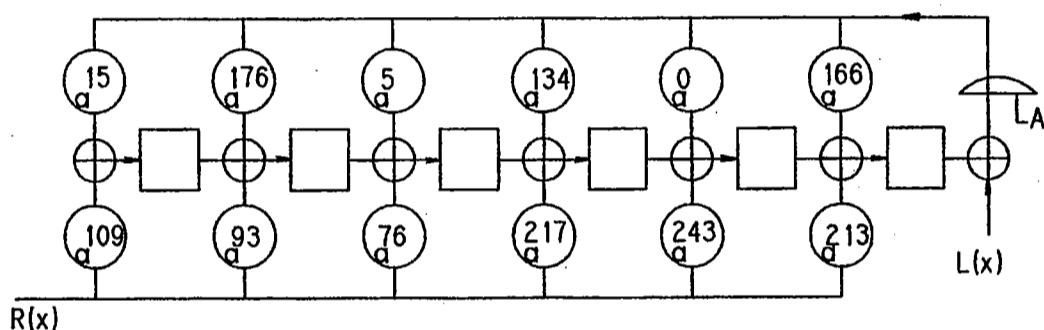


FIG. 5

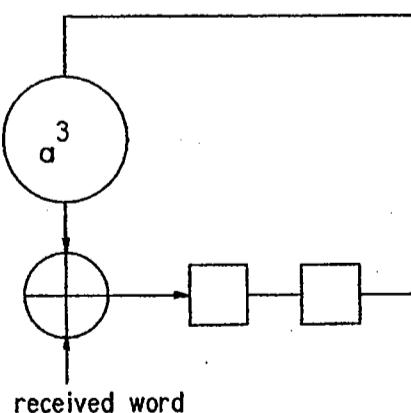


FIG. 6

U.S. Patent

Oct. 25, 1994

Sheet 7 of 13

5,359,468

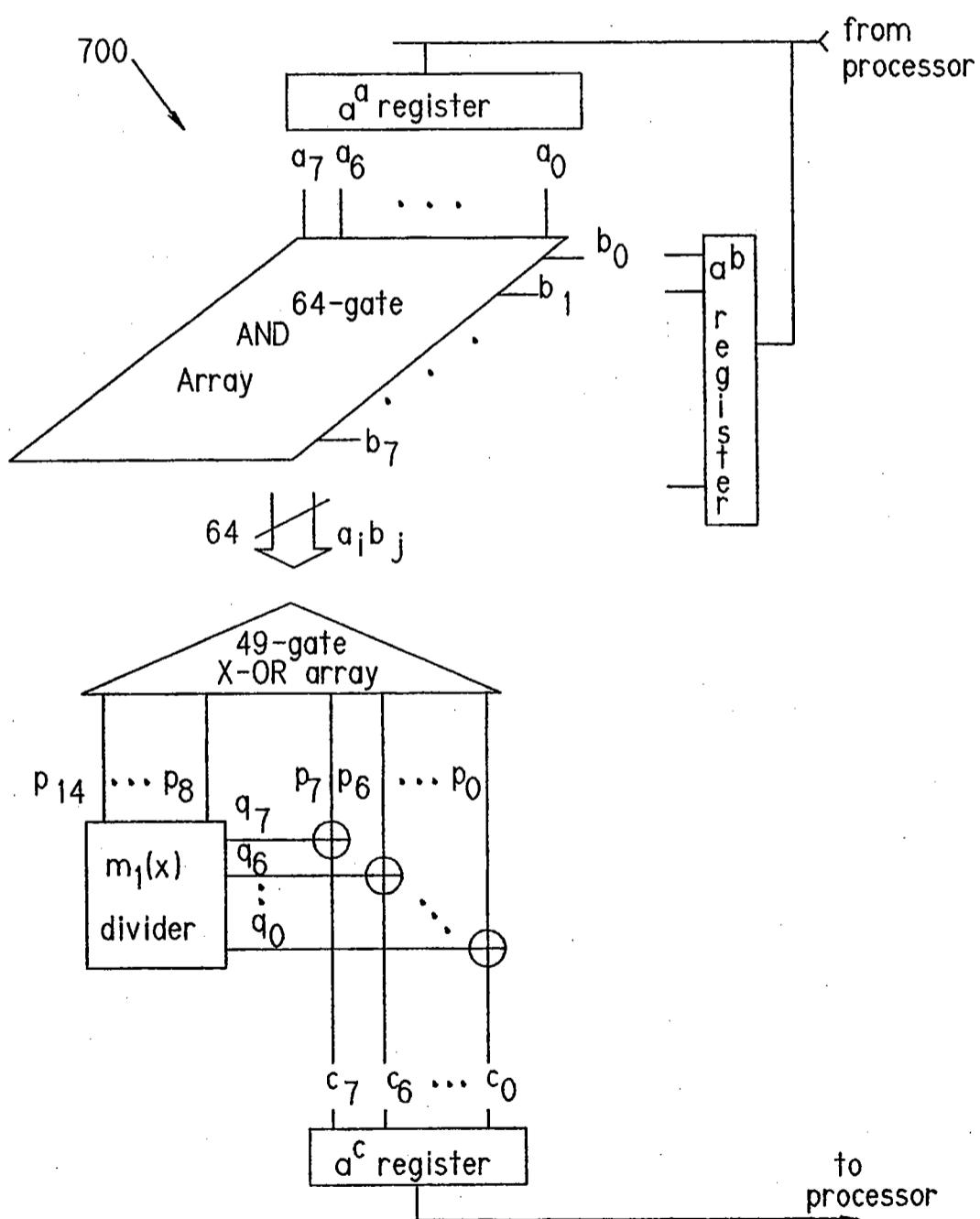
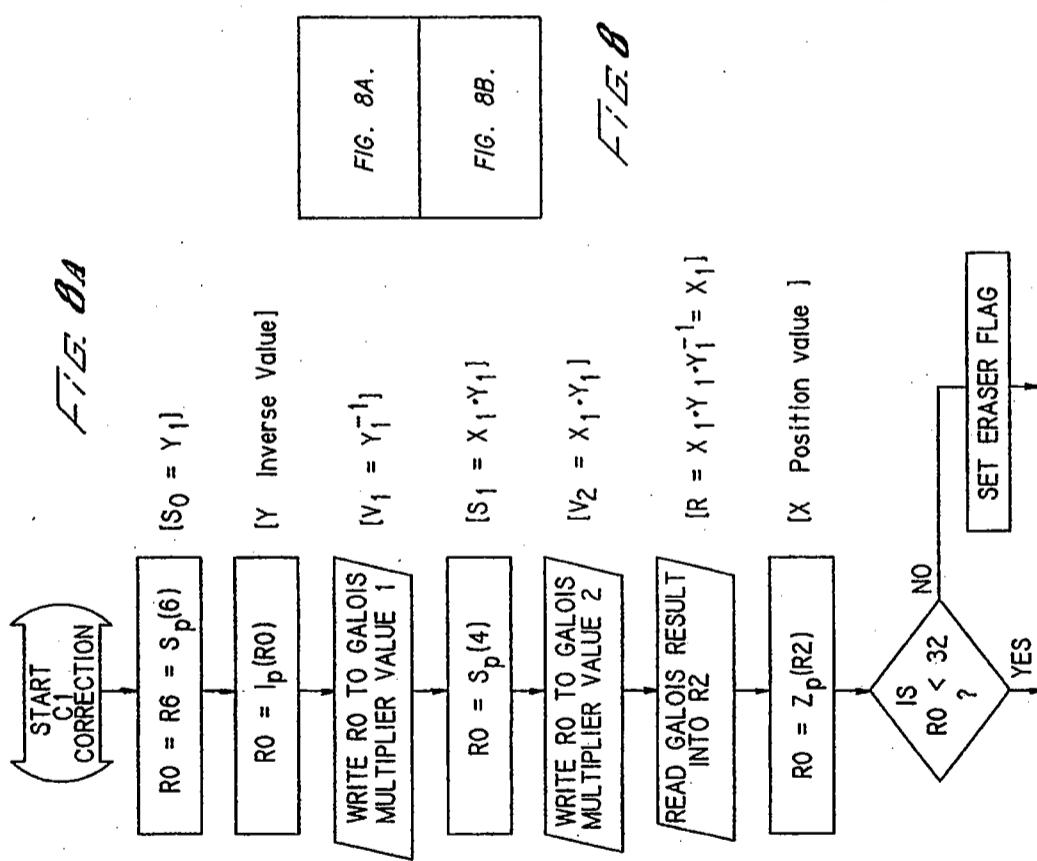
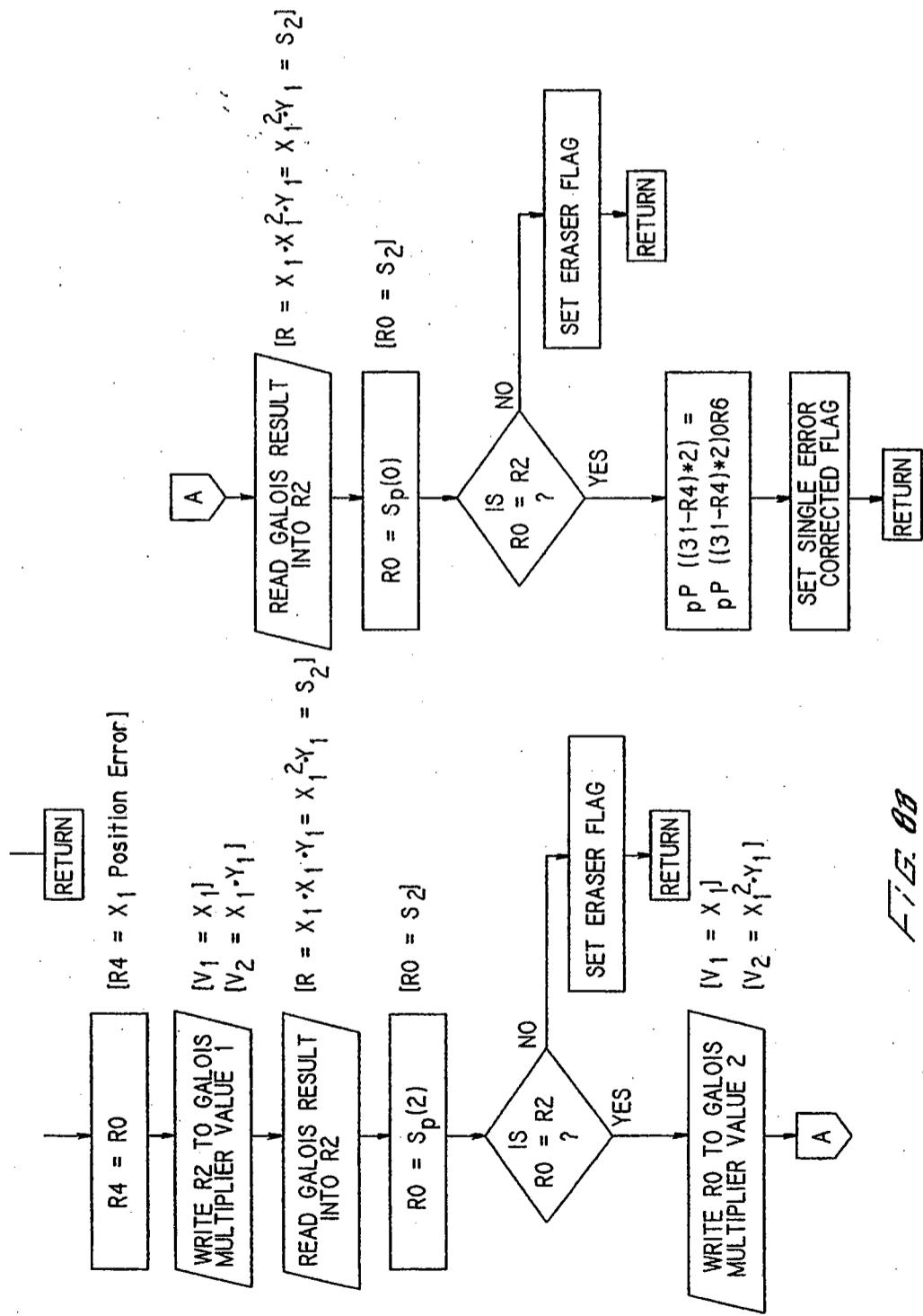
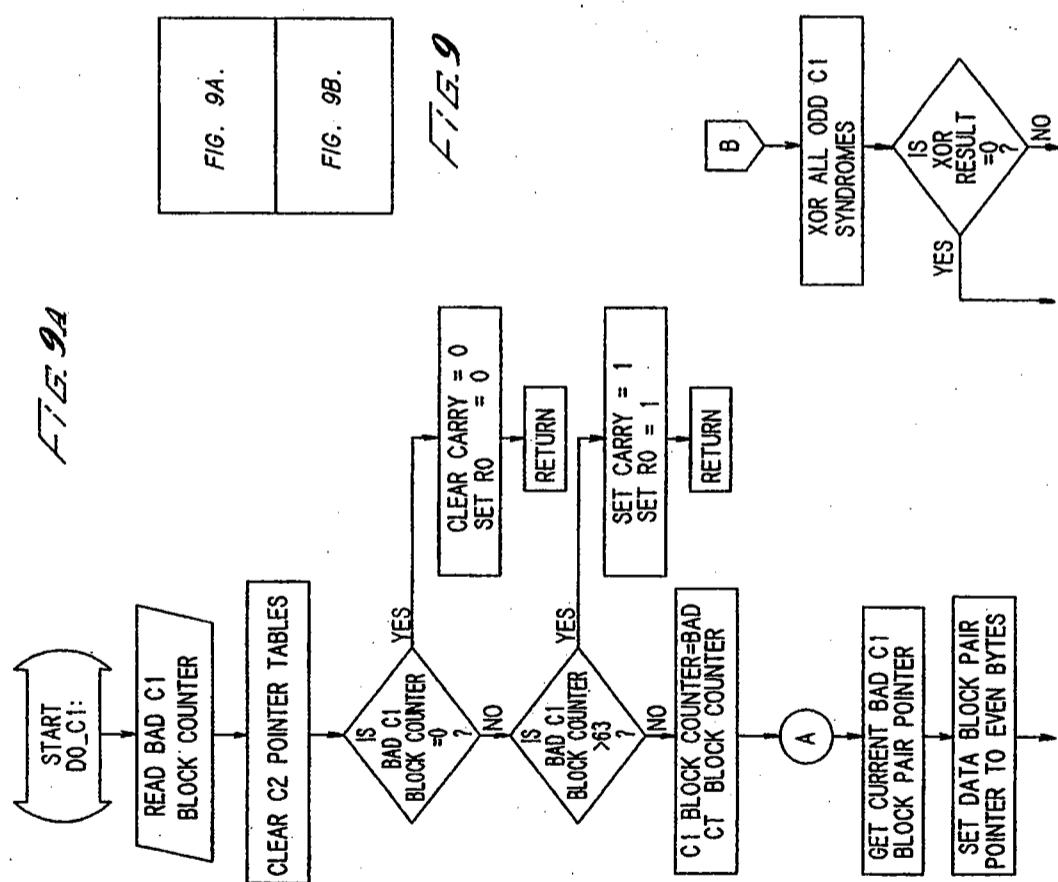
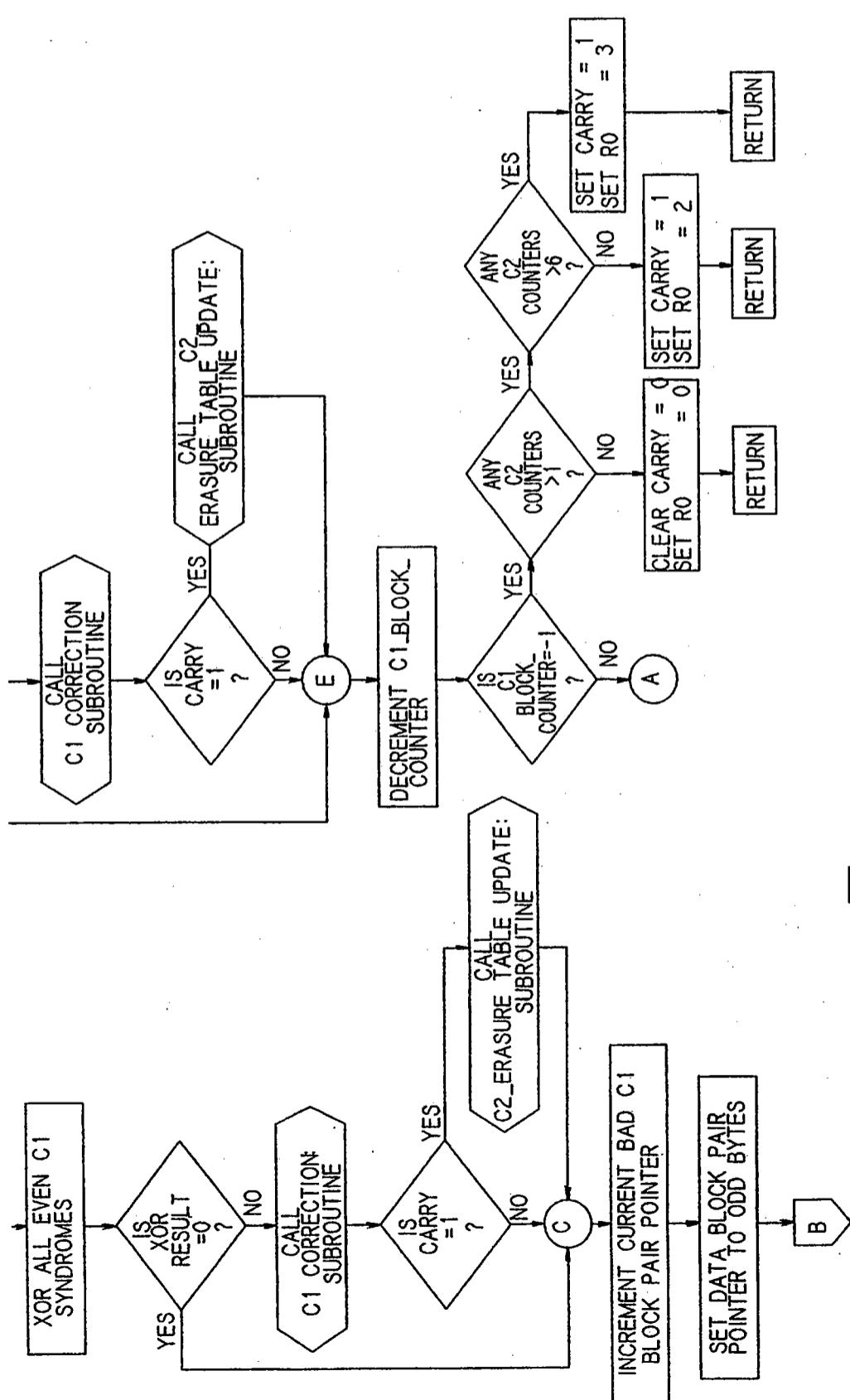


FIG. 7









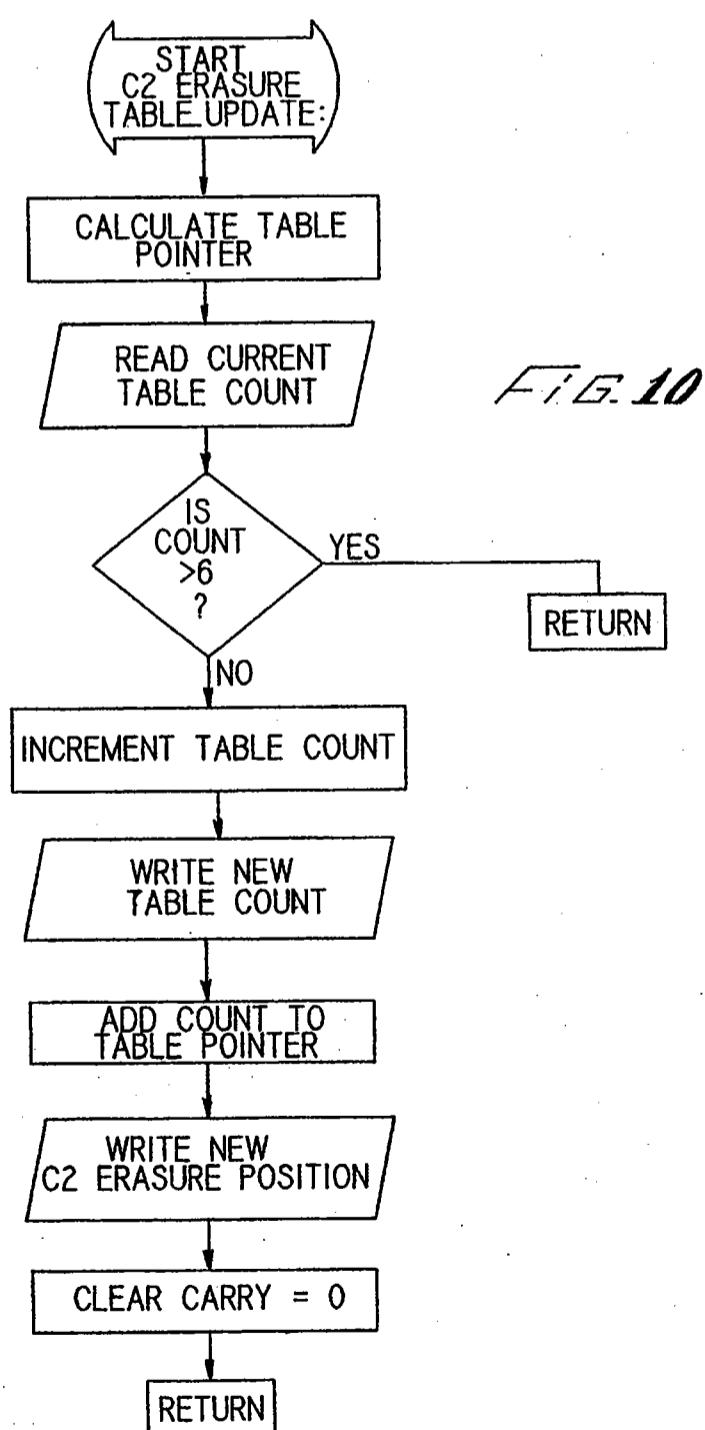


FIG. 10

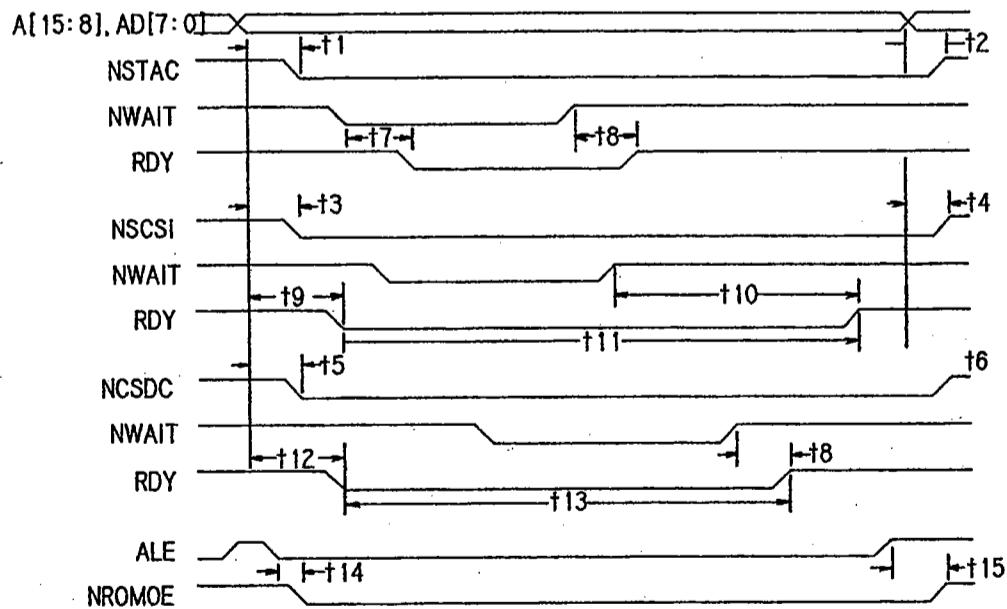
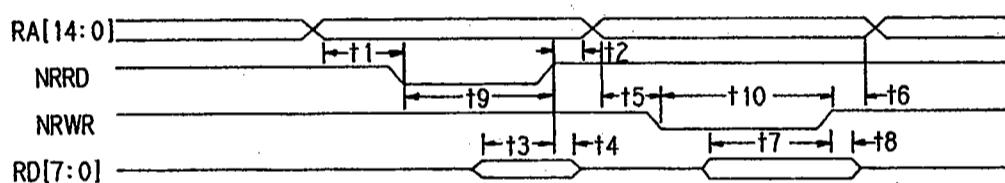
**U.S. Patent**

**Oct. 25, 1994**

**Sheet 13 of 13**

**5,359,468**

*FIG. 11*



*FIG. 12*

**DIGITAL DATA STORAGE TAPE FORMATTER**

Reference is made to a microfiche appendix A comprising ninety-one frames of electronic schematic drawings including gate logic diagrams showing an embodiment of this invention for the facilitation of understanding the invention.

**FIELD OF THE INVENTION**

The present invention relates generally to the field of digital data storage in audio tape and more particularly to an efficient formatting apparatus and method for high-performance tape systems.

**BACKGROUND OF THE INVENTION**

The mechanism of coding on magnetic tape is by a formatter device which codes and writes bits represented by magnetic flux reversals on a ferromagnetic tape medium. There are many different types of coding used in the prior art, varying according to polarities (return to zero or not during a transition), bit train compression, and clocking capability. The most common coding schemes for high-performance tapes are non-return-to-zero-inverted (NRZI), phase encoding (PE), and group coded recording (GCR) which is a combination of NRZI and PE. A code is self-clocking if a signal pulse is generated for every stored bit.

Characters are recorded on tape by tracks with each character stored in a column across the tape with embedded parity bits for error checking. Typically, each track has one write head and usually one read head. To further limit errors, information written on tape is often read immediately after being written (so-called read-after-write or RAW) by a separate read head mounted closely to the write head. On a typical tape there is a stored addressing information (SAI) section for locating a record and a data section which may also provide additional addressing information. The SAI typically includes (in sequence) a postamble immediately adjacent the previous data record, an interrecord gap (IRG) providing a space interval for tape motion changes, beginning and end of tape characters, various other markers, clocking and deskewing information, and a preamble immediately adjacent the next data record. The preamble utilizes sync marks to synchronize detection circuits for distinguishing bits. The postamble signals the end of a data record or block. To save space and access time, IRGs may be placed between blocks (IBGs) rather than records and related blocks may be grouped into a file and designated by an end of file marker. "Load point" and an "end of reel" markers indicate the beginning and end of the tape respectively and are typically reflective for detection by a photocell in the tape drive unit.

A standard format for digital data storage (DDS) using 3.81 mm digital audio tape (DAT) magnetic tape is set forth by the European Computer Manufacturers Association in the document "Flexible Magnetic Media for Digital Data Interchange" (ISO/IEC JTC 1/SC 11N 1026, hereinafter "DDS standard", 1990-07-13).

Briefly, DDS format data has two types of separator marks indicating logical separations of the data. Separator 1 is a "file mark" and separator 2 is a "set mark". User data, separator marks, and associated information are formed into groups occupying groups of tracks in a "main zone" of the track. Additional information about the contents of the group, the location of the tracks and

the contents of the tracks is recorded in two parts of each track called "sub zones". The two sub zones constitute the "subdata" area of the track. In addition, there are margin zones at the extreme ends of the tape and 5 Automatic Track Finding (ATF) zones between the sub zones and the main zone. Each zone in a track is further segmented into blocks called margin blocks (in the margin zone), preamble, subdata, and postamble blocks (in the sub zones), spacer and ATF blocks (in the ATF zone), and preamble and main data blocks (in the main zone). A "frame" is a pair of adjacent tracks with azimuths of opposite polarity (where the azimuth is the angle between the means flux transition line with a line normal to the centerline of the track). Data to be recorded is grouped into "basic groups" of 126632 bytes. Each basic group is identified by a running number from 1 to 126632. Data and separator marks are grouped into the basic groups starting with basic group no. 1. Error Correction Codes (ECC) are termed C1, C2, and C3 which are computed bytes determined from the data using Read-Solomon error correction techniques (see below for details). C3 is capable of correcting any two tracks which are bad.

Write data channel functions, including coding and error correction code, are typically performed by a controller operating through a write amplifier positioned near the write head. The write amplifier drives the write current through the write head.

Read data channel functions, including amplification and equalization of the read signals and data retrieval, 30 are typically performed by automatic track-oriented gain-adjustment by a read amplifier and timing, deskew-ing, decoding, error detection and correction by a controller. The fundamental function of readback is to accurately convert the amplified read signal waveform into its binary equivalent. During writing, an external clock (oscillator) spaces recorded bits. An accurate readback therefore must be synchronous; and a code which inherently strobes the readback signal is desirable, such as self-clocking pulse generation in PE and GCR. One special type of GCR coding is 8-10 GCR which, briefly, is a coding scheme mapping each 8 bit group of data into a 10 bit code group. When the resulting 10 bit code groups are concatenated in any sequence, the resulting bit stream has the characteristic that there are never more than 3 0's in a row.

Prior art formatter systems for DDS/DAT typically require 96 to 192K bytes of static random access memory (SRAM). DDS type drives operate at 9.408 million code bits per second. In prior art read-after-write (RAW) systems, rewriting a bad frame after RAW-checking typically requires 2 to 5 intervening frames which wastes that amount of tape capacity per bad frame. Prior art systems were unreliable primarily because of multiple components and complex electronic interconnections. This resulted in relatively physically large formatter systems and high costs.

There is therefore a need for a smaller, less expensive formatter having fewer components, but achieving greater reliability and more efficient performance in terms of speed and capacity.

**SUMMARY OF THE INVENTION**

Accordingly, in order to advance the magnetic digital data storage art, the present invention has the following objects:

It is a principal object of the present invention to provide an improved formatter for digital data storage.

It is another object of the present invention to provide a formatter having fewer components and reduced electronic complexity, while achieving greater reliability and more efficient performance.

It is a further object of the present invention to achieve all of the formatter functions on a compact single chip.

It is still a further object of the present invention to provide more efficient tape usage and thereby increase tape capacity.

It is yet a further object of the present invention to reduce the formatter static random access memory requirement.

It is another object of the present invention to increase the speed of magnetic tape digital data storage 15 read and write operations.

It is still another object of the present invention to improve the reliability of magnetic tape digital data storage read and write operations.

To achieve these and other objects, the present invention is contemplated to operate with a digital data storage audio tape system having a host device interface unit, a main data buffer manager, a main data buffer, a processor, a data separator, a tape drive unit, and a read/write channel. This invention comprises a formatter providing track construction, encoding and decoding, error correction, and direct memory access of digital data transmitted between the host device unit and the tape drive unit and a frame buffer for buffering digital data for encoding and decoding, and error correction. The formatter is coupled between the main data buffer manager, via a direct memory access channel, and the read/write channel of the tape drive, and further is coupled to the frame buffer. The formatter is coupled to and responsive to instruction control by the processor in conjunction with instruction control of the tape drive unit by the processor.

The method of this invention is for accessing and encoding and decoding the data on a per frame and per track basis, comprising the operational steps of interleaving, randomizing, C1/C2 ECC generation, subdata block processing, sync generation, automatic track finding signal generation, deinterleaving, derandomizing, C1/C22 ECC, syndrome generation, and subdata block capture.

This invention provides a significant overall reduction in parts count, costs, and electronic complexity, and subsequent increased reliability over the prior art. A reduction in RAM requirements is realized while still performing (1) buffering two or three frames of data during tape read operations, (2) buffering three frames of write data during write operations, (3) storing subdata information for each frame, (4) storing ID information for each frame, (5) storing ECC information such as parity symbols, syndromes, bad block pointers, and the like, and (6) providing storage for microcontroller scratch memory for storage of variables, pointers, data stack, and the like. In the preferred embodiment, the above operations are achieved by a frame buffer of 32 Kbytes, which is one-third to one-sixth the RAM required in prior art formatter systems. Data transfer rates are higher than the standard DAT of 9.408 million code bits per second. This is achieved by an independent oscillator allowing bit rates of 12 million code bits per second. A parallel-functioning randomizer/derandomizer is utilized that is eight times faster (for 8-bit systems) than the prior art. Tracks with missing sync marks at the beginning of its main data field may still be read

with only simple C2 error correction. In RAW operation, a bad frame is re-written with only one intervening frame as contrasted with prior art systems requiring two to five intervening frames which wasted tape space. An independent track checksum unit acts on RAW data which allows more stringent RAW criteria resulting in higher data integrity. The invention also provides ATE testability and diagnostic loopback capability.

A further understanding of the nature and advantages 10 of the present invention may be realized by reference to the Detailed Description of the Invention and the attached drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic block diagram of a digital data tape storage system including a system according to the present invention.

FIG. 2 is a block diagram of a simplified write operation according to one embodiment of the present invention.

FIG. 3 is a block diagram of a simplified read operation according to the present invention.

FIG. 4 is an organizational and functional block diagram of the preferred embodiment of a formatter according to the present invention.

FIG. 5 is an exemplary linear feedback shift register circuit for an error correction encoder according to the present invention.

FIG. 6 shows a linear feedback shift register for calculating error correction code syndromes according to the present invention.

FIG. 7 shows the structure of an embodiment of an array multiplier for GF(2<sup>8</sup>) according to the present invention.

FIG. 8 is a flowchart of a C1 error correction subroutine to correct one C1 error according to the present invention.

FIG. 9 is a flowchart of a C1 error correction and C2 mark erasures main subroutine according to the present invention.

FIG. 10 is a flowchart of a C2 mark erasure main subroutine according to the present invention.

FIG. 11 is a frame buffer (SRAM) access timing diagram according to the present invention.

FIG. 12 is an address decode and RDY timing diagram according to the present invention.

#### DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 is a schematic block diagram of a magnetic tape digital data storage system 100 according to the present invention. System 100 in this embodiment includes an interface unit 102, a buffer manager 103, a data main buffer 104, a microcontroller 105, a formatter 110, a data separator 118, a frame buffer 117, a high-speed serial expansion bus 113, and other data buses 114, 131, and 132. Formatter 110 includes a C1/C2 error correction code (ECC) unit 121, a direct memory access (DMA) controller 122, and an encode/decode (ENDEC) unit 123. In the preferred embodiment, formatter 110 is implemented in a single chip and frame buffer 117 is a single chip static random access memory (SRAM) unit. C1/C2 ECC unit 121, DMA controller 122, and ENDEC 123 are coupled to each other and to RAM frame buffer 117 via high-speed parallel bus 131. ENDEC 123 is coupled to bus 133.

Interface unit 102 is for communication with a host computer 101 (not shown), and is coupled to buffer

manager 103 which in turn is coupled to main data buffer 104 and to microcontroller 105 via bus 132.

Buffer manager 103 includes a C3 ECC unit 129 and a buffer manager sub unit 128. Both are coupled to bus 132 which couples interface 102 to DMA controller 122 and main buffer 104.

ENDEC 123 serially interfaces an analog read/write channel 133 which is coupled to a tape drive unit 150. Main buffer 104 and buffer manager 103 are parallel-interfaced through bus 132 to DMA controller 122. Frame buffer 117 is parallel-interfaced through Internal data path bus 131. DMA controller 122 is linked to main data buffer 104 via bus 132 and to frame buffer 117 via bus 131. Microcontroller 105 is interfaced directly to internal registers (not shown) in formatter 110 and there is arbitrated access from microcontroller 105 to frame buffer 117. A serial to parallel path from read and write channels 133 to frame buffer 117 travels through ENDEC 123. A high-speed parallel bus 134 couples frame buffer 117 to C1/C2 ECC unit 121. For diagnostic convenience, all data paths include loop back capability.

In operation, broadly, formatter 110 is disposed between main data buffer 104 and write/read channel 133 of the tape drive. Bus 132, acting as a DMA channel, carries frames between main data buffer 104 and frame buffer 117. Formatter 110 formats user data on a per frame and per track basis, encodes/decodes and serializes/deserializes the data and SAI for read and write operations. User data is received from a host computer through interface 102 and buffer manager 103, and operations are performed under the control of microcontroller 105.

In a write operation, formatter 110 does read after write (RAW) checking, which in conjunction with microcontroller 105, determines whether a track was successfully written. If a frame fails to be properly written, microcontroller 105 immediately instructs formatter 110 to rewrite the failed frame by simply changing a buffer pointer and updating specific subdata areas. Triple buffering provided by frame buffer 117 allows microcontroller 105 to keep a copy of the frame which is being RAW-checked in a buffer while simultaneously processing new frames for writing. The formatting operation includes interleaving/deinterleaving, randomizing/derandomizing, C1/C2 ECC generation, sub-data block processing, sync generation, and automatic track finding (ATF) signal generation. Formatter 110, through DMA controller 122, interfaces multiplexed address/data bus 114 to communicate with microcontroller 105. The registers in formatter 110 are mapped to the memory of microcontroller 105. Microcontroller 105 can access all of frame buffer 117 in the data space of microcontroller 105's address space. Formatter 110 also includes system support logic (such as decoding microcontroller's address space, generation of chip selects and wait states) for microcontroller 105, buffer manager 103, interface 102, and a data compression chip (if desired).

The function of DMA controller 122 is to move data between buffer manager 103 and frame buffer 117. In the preferred embodiment, DMA controller 122 is programmable. Frame buffer 117 is divided into four logical buffers (of 8K bytes each, in the preferred embodiment). Three of the logical buffers are used for buffering read and write data. During a read as part of a RAW cycle, data is not stored in any buffer. This is because C1 ECC syndromes are calculated and non-zero syn-

dromes are stored in a dedicated area of frame buffer 117. Track checksums are also computed and are stored in registers in formatter 110. The stored syndrome and checksum information is sufficient for checking the correctness of the written data, so the actual data being read back is discarded. The fourth buffer is used as storage area for microcontroller 105 and formatter 110. Microcontroller 105 accesses frame buffer 117 through a window register on formatter 110.

FIG. 2 is a block diagram of a simplified write operation according to one embodiment of the invention. User data flows from a host computer at 101 to tape at 199 in communication with a head drum 290 first through interface 102 and buffer manager 103 to main data buffer 104. Data is assembled in main data buffer 104 until there is one group of data, organized as 22 frames. Microcontroller 105 then generates specific system data (such as group number, filemark, save set marks, C3 ECC frame, etc.) for the data group in main data buffer 104. Next, the data group is moved under the control of microcontroller 105 to frame buffer 117 (which in the preferred embodiment is 32K×8 SRAM). Microcontroller 105 then initializes DMA controller 122 in formatter 110 and BFR MGR 128 in buffer manager 103 (all shown in FIG. 1) to move one frame of data. Returning to FIG. 2, data now begins to move from main data buffer 104 to a write formatter 210. As each byte is transferred from buffer manager 103 to formatter 210, the following operations are performed on the fly before that data is written to frame buffer 117: (1) data is randomized (compatible with the DDS standard, in the preferred embodiment) by randomizer 237, a running track checksum of transferred bytes is computed by track checksum 240 and (3) data is interleaved (compatible with the DDS standard, in the preferred embodiment) by interleave unit 271. In the preferred embodiment, randomizer 237 is parallel-functional as opposed to prior art systems which are serial-functional. The present invention is therefore eight times faster than prior art formatters for 8-bit systems. Further, randomization and interleaving is performed on the fly. Microcontroller 105 next moves system area data 272 for the single frame to frame buffer 117. Formatter 210 then computes C2 parity bytes using C2 parity generator 231 and writes to frame buffer 117 on a frame basis. Formatter 210 then begins to read data at 274 from frame buffer 117 and, before outputting the serial data stream, performs the following operations: (1) C1 parity bytes are computed on the fly by C1 parity generator unit 260, (2) an 8-10 conversion is performed at 276, and (3) other track format data is generated at 278 (such as ATF signals, margins, IBG, preambles, subdata and the like compatible with the DDS standard) in synchronization with head drum 290 rotation. In the preferred embodiment, head drum 290 includes four read/write heads which rotate synchronized with read/write operations. This is described in detail in System patent application Ser. No. 07/741,783 which is hereby incorporated by reference. At the end of frame, microcontroller 105 checks various status registers 259 to determine if a proper operation was completed on that frame. If proper, the next data frame is written from the beginning of the generation of system area data. Arbiter 235, coupled to frame buffer 117 by write frame buffer bus 224, manages and interleaves access to frame buffer 117. In the preferred embodiment, 8-10 GCR encoding and decoding are utilized.

FIG. 3 is a block diagram of a simplified read operation according to the invention. R/W channel 133 transmits read data to data separator 118 which transmits synchronized serial data and clock pulses to a read formatter 310. A deserializer 308 converts the serial data to parallel, an 8-10 decoder 370 decodes the data, a track checksum unit 361 performs independent track checksum, a C1 syndrome generator 312 computes C1 syndromes on the fly and writes both the data and syndromes into frame buffer 117 after buffer management by a frame buffer manager 315. If all C1 syndromes are zero, there are no read errors in the track. When a non-zero syndrome is detected, the block location is stored into a specific area in frame buffer 117 for quick microcontroller 105 access. It should be noted that error correcting software for microcontroller 105 may be invoked at this point to correct errors indicated by non-zero syndromes. After C1 correction is completed, microcontroller 105 enables C2 ECC unit 320 to generate C2 syndromes for the frame. Each time a non-zero 20 syndrome is generated, C2 ECC unit 320 pauses so that programmed control in microcontroller 105 can perform error correction for that C2 block. Note that the power of the C2 error correction is increased if the C1 control passes so-called "erasure pointers" to the C2 25 control. Erasure pointers are indicators that certain errors were not correctable by the C1 control, but the locations of such errors are known to the C2 control from this information. In a similar manner, if a track is known to have errors which are detected but are not 30 correctable by C2 control, these tracks are marked as erasure pointers for the C3 error correction control. Once a frame of data is assembled into frame buffer 117, and all C1 and C2 errors corrected, the data is moved into main data buffer 104 via bus 132. From there, the 35 data is moved to interface 102 for communication with the host computer. Arbiter 235, coupled to frame buffer 117 by a read frame buffer bus 324, manages and interleaves access to frame buffer 117. In the invention, C1 parity and syndromes are generated on the fly as data is moved to and from the tape, for main data as well as subdata.

FIG. 4 is an organizational and functional block diagram of the preferred embodiment of formatter 110. A microprocessor interface 401 interfaces microcontroller 105 (of FIG. 1) via multiplexed address/data bus 114 and a high-order microcontroller address lines 472. An internal processor bus 402 couples a formatter buffer manager (FBM) 405 and a formatter-sequencer 406 to interface 401. An internal read/write bus (IRWB) 403 couples FBM 405 and formatter-sequencer 406. FBM 405 is coupled to DMA channel bus 132, read frame buffer bus 324 and to address lines 474 for frame buffer 117 (of FIG. 3). Formatter-sequencer 406 is coupled to read/write channel 133 by buses RDDAT 407 and WRDAT 408. A detailed description of system signals in formatter 110 is provided below.

Functions are organized in the block components as follows. Microprocessor interface block 401 includes an address map decoder 411, ready logic 412, interrupt control 413, serial port handshake logic 414, and the following registers: interrupt 415, serial port addr/stat 416, serial port data 417, Galois field value 1 418, Galois field value 2 419, Galois field result 420, Galois square root 421, jumper input 422, and chip select generator 423.

FBM 405 includes a C2 parity generator 431, a C2 syndrome generator 432, a data path control 433, a

buffer address generator 434, a direct memory access (DMA) arbitrator 435, a DMA handshake 436, and a randomizer 437, and the following registers: DMA controller control 438, SRAM access latch 439, track 1 checksum (2) 440, 441, track 2 checksum 442, 443, ECC control 444, control pulse 445, C2 block address 446, and bad C1 block count 447.

Formatter-sequencer 406 includes a C1 parity generator 451, a C1 syndrome generator 452, a read/write sequencer 453, an encoder/decoder 454, an automatic track follower (ATF) pattern generator 455, and an ID parity 456, and the following registers: serial/deserializer (serdes) control 0 457, serdes control 1 458, status 459, two read track checksum units (2) 460, 461, and a bit pointer 462, a state counter 463, a block counter 464, and a byte counter 465.

In the preferred embodiment, formatter 110 also includes specific "glue" logic to integrate other system functions. These include microcontroller address map decoder 411, ready logic 412, interrupts 413, serial port handshake logic 414, microcontroller address/data bus demultiplexing, serial communication channel to an AFT, full support for microcontroller-based diagnostics, tri-state and activity modes to support ATE testing and the like. In particular, address map decoder 411, chip select generator 423 and ready line/wait state logic 412 as system functions reduce parts count, cost, and electronic complexity.

Interface block 401 performs address decoding at 411, wait state generation at ready logic 412, and interrupt control at 413.

Read-after-write (RAW), with re-write of bad blocks, is implemented to be compatible with the DDS standard. A frame is re-written using the following procedure. C1 ECC is checked on the main data area of each frame. The frame is rewritten if any block in the main data area of the frame has a C1 syndrome indicating two or more symbols in error. Also, the frame is re-written if more than N blocks in the main data area of either track of the frame have syndromes indicating single symbol errors. (N is between 1 and 14, and preferably 3.) This guards against potentially unreadable data areas. Next, C1 ECC is checked on the subdata areas of each frame. The frame is re-written if more than M blocks in the subdata area of either track of the frame have syndromes indicating any type of error. (M is between 1 and 8, and preferably 2.)

According to the present invention, under RAW operation, a bad frame may be re-written with only one intervening frame as contrasted with prior art systems requiring two to five intervening frames which wasted tape space. The frame is re-written if more than 14 blocks in any area of either track of the frame have C1 syndromes indicating any type of error. Read track checksum 461 calculates DDS track checksums as data is transferred from the read channel to frame buffer 117. The frame is re-written if any discrepancy is found in the track checksums. The track checksums are re-computed during the main data area read (and error-corrected if necessary), and compared against the stored value that was written for that track. The checksum is also compared to the value read back from one good sub data block pair. This guards against drop-ins in the main data area of the frame. The independent track checksum unit 361 (FIG. 3) or 461 (FIG. 4) acts on RAW data which allows more stringent RAW criteria resulting in higher data integrity.

In more detail, read after write (RAW) is user selectable. When RAW is selected, only group data frames will be checked for read errors. Amble frames are not checked for read errors. For RAW, a track's worth of data is checked for errors. First, the number of C1 errors over the main data is checked. The error checking is done by comparing the track checksum of the track read with the checksum of the track that was written. Also, the number of C1 ECC errors are counted. The criteria used to decide if a frame needs to be re-written or not re-written, while doing single byte C1 corrections over the user data area, is the following: (1) greater than 32 C1 ECC errors on a track; (2) greater than 8 C1 ECC single errors on a track's user data blocks; (3) greater than 8 C1 ECC multiple errors on a track; (4) greater than 0 C1 ECC multiple errors on a track's user data blocks; and (5) track checksums are not equal even after C1 ECC checksum correction. These criteria are contained in microcontroller firmware which can be easily modified to satisfy different user requirements.

When a track has enough errors to enable a RAW re-write, a flag is set in memory signaling that there has been a RAW error. Before track A+ of each frame is to be written, the RAW flag is checked. If the flag is set, track A+ data is taken from the buffer that had the RAW error. Otherwise, the track A+ data is taken from the next buffer to be written. Until a frame has passed the RAW check, no other frames will be checked for RAW errors. If a frame fails the RAW check 127 times, a fatal error will have occurred and the R/W block 303 will stop writing data. For a detailed description of the command function environment implementing the RAW operation, see the System patent application Ser. No. 07/741,783.

Formatter 110 can write raw code bit patterns from main data buffer 104 and transmit them to write channel 133. This may be utilized for creating special test, diagnostic, and calibration tapes with formats that deliberately deviate from the DDS standard.

#### C1/C2 ECC

Error correcting code (ECC) is embodied in an algorithm producing check bytes used for the detection and correction of errors. A detailed description of ECC can be found in Peterson, W. W. and E. J. Weldon, Jr., *Error-Correcting Codes: Ed. II*, MIT Press (1972). The ECC utilized in an embodiment of the present invention is specified in the DDS standard and is briefly described following.

#### Encoder

The present invention employs two separate orthogonal ECCs known as C1 and C2 level ECC. The inner C1 (32, 28) Reed-Solomon (RS) code using  $\alpha$  field elements has

$$\begin{aligned} g_1(x) &= (x + \alpha^0)(x + \alpha^1)(x + \alpha^2)(x + \alpha^3) \\ &= x^4 + \alpha^{75}x^3 + \alpha^{249}x^2 + \alpha^{78}x + \alpha^6 \end{aligned}$$

The outer C2 (32, 26) RS code using  $\alpha$  field elements has

$$\begin{aligned} g_2(x) &= (x + \alpha^0)(x + \alpha^1)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4)(x + \alpha^5) \\ &= x^6 + \alpha^{165}x^5 + \alpha^0x^4 + \alpha^{134}x^3 + \alpha^5x^2 + \alpha^{176}x + \alpha^{15} \end{aligned}$$

Encoders to implement these codes are multi-stage linear feedback shift registers with connections corre-

sponding to  $g_i(x)$  and are known in the magnetic tape recording art.

#### Decoding and Correction

C1 coding produces four parity symbols for each 32 symbol block. It is capable of correcting 0, 1, or 2 errors in a block and detecting many errors of higher weight. Flags may be passed to the C2 error correction control logic giving location information for errors which are not corrected (or may have been misdetected) at the C1 level.

C2 coding produces six parity symbols for each 32 symbol block. It is capable of correcting up to six erasures (errors having known locations) or three errors in a block, or combinations of the same according to the following table.

Number of Erasures	Number of Errors
0	0,1,2,3
1	0,1,2
2	0,1,2
3	0,1
4	0,1
5	0
6	0

Further, many errors of higher weight also can be detected. Flags may be passed to the higher level control logic giving location information for errors which are not corrected (or may have been misdetected) at the C2 level. The higher level control logic may invoke further recovery processes, such as C3 level ECC correction or read re-tries.

In the present invention, the decoding of C1 and single erasure correction (error weight=0.5) in C2 is performed "on the fly". Higher-weight error patterns in C2 will require a tape stoppage for correction.

In the present invention, decoding and error correction are performed by a combination of hardware circuits in the formatter and software control logic in the microcontroller. Specifically, formatter 310 includes a C1 syndrome generator, a C2 syndrome generator, a GF(2<sup>8</sup>) Galois field multiplier, and a GF(2<sup>8</sup>) Galois field square root table. Other than the calculations performed by these circuits, all decoding and correction is done in microcontroller 105. A typical decoding computation performed by microcontroller 105 is:

- (a) multiplication of two local operands (GF(2<sup>8</sup>) elements)
- (b) addition (XOR) of a third element
- (c) test for zero.

Since many such computations are involved in ECC decoding and correction, it will be appreciated that the Galois multiplier function in the formatter hardware provides considerable performance benefit.

In the present invention, microcontroller 105 performs a number of other functions in addition to error correction. See the System patent application Ser. No. 07/741,783 for detailed descriptions. Roughly speaking, only about 75% of the processor cycles will be available for ECC.

Decoding and error correction techniques utilizing syndrome information and Galois arithmetic are known in the magnetic tape recording art. In the present invention, such techniques are implemented in software control algorithms embodied in microcontroller 105.

#### Syndrome Calculation

The code C1 is interleaved to degree 2. For this code, four syndrome calculators are needed. FIG. 6 shows a

linear feedback shift register for calculating  $S_3$ . The other circuits for calculating  $S_0$ ,  $S_1$ , and  $S_2$  are similar.

FIG. 5 shows an exemplary linear feedback shift register circuit for a C2 encoder. The code C2 is interleaved to degree 4 and has six parity checks. Thus circuits similar to that of FIG. 5 for computing  $S_0$ ,  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$  and  $S_5$  are utilized in this embodiment.

#### GF(2<sup>8</sup>) Array Multiplier

FIG. 7 shows the structure of an embodiment of an array multiplier. The inputs to circuit 700 are the field elements

$$\alpha^a = a_7x^7 + a_6x^6 + \dots + a_0$$

and

$$\alpha^b = b_7x^7 + b_6x^6 + \dots + b_0$$

The product of these polynomials has the form

$$P(x) = P_{14}x^{14} + P_{13}x^{13} + \dots + P_0$$

where

$$P_k = \sum_{i=0}^k a_i b_{k-i}$$

In the field GF(2<sup>8</sup>), as generated by

$$m_1(x) = x^8 + x^4 + x^3 + x^2 + 1,$$

the following relationships hold

$$\begin{aligned} x^{14} &= &+ x^4 &+ x + 1 \\ x^{13} &= x^7 &+ x^2 + x + 1 \\ x^{12} &= x^7 + x^6 &+ x^3 + x^2 &+ 1 \\ x^{11} &= x^7 + x^6 + x^5 &+ x^3 \\ x^{10} &= x^6 + x^5 + x^4 &+ x^2 \\ x^9 &= x^5 + x^4 + x^3 &+ x \\ x^8 &= x^4 + x^3 + x^2 &+ 1 \end{aligned}$$

Thus the coefficients of the final product

$$\alpha^c = \alpha^a + \alpha^b = c_7x^7 + c_6x^6 + \dots + c_0$$

can be calculated as follows:

$$\begin{aligned} C_7 &= P_{13} + P_{12} + P_{11} &+ P_7 \\ C_6 &= P_{12} + P_{11} + P_{10} &+ P_6 \\ C_5 &= P_{11} + P_{10} + P_9 &+ P_5 \\ C_4 = P_{14} & &P_{10} + P_9 + P_8 + P_4 \\ C_3 = P_{12} + P_{11} & &+ P_9 + P_8 + P_3 \\ C_2 = P_{13} + P_{12} & &+ P_{10} + P_8 + P_2 \\ C_1 = P_{14} + P_{13} & &+ P_9 + P_1 \\ C_0 = P_{14} + P_{13} + P_{12} & &+ P_8 + P_0 \end{aligned}$$

Array multiplier 700 employs 64 2-input AND gates and 49 + 28 (= 77) 2-input XOR gates

Single erasure correction requires a total of six "computations", which the decoder must perform in one

word time to correct one error in C1 and one erasure in C2 "on the fly". Since microcontroller 105 is capable of performing 12 computations per word-time, "on the fly" correction is easily achieved by the decoder. For details of C1 and C2 decoding and single erasure correction, see the Peterson book referenced above.

FIG. 8 is a flowchart of a C1 error correction subroutine to correct one C1 error. Note that C1 is only checked over main data and that if there is more than one error in a block, then that block is flagged as a C2 erasure block. It is assumed that a non-zero C2 has been found and the FIG. 8 flowchart is then begun. All pointers are set up before calling the subroutine and four registers are used and indexes are utilized for the pointers such that the pointers are not changed (see below for detailed descriptions of the registers).

FIGS. 9 and 10 are flowcharts of a C1 error correction subroutine to correct C1 and to mark C2 erasures.

The timing diagram for frame buffer 117 is shown in FIG. 11 with the timing descriptions listed in Appendix II. The address decode and ready timing diagram is shown in FIG. 12 with the timing descriptions listed in Appendix III.

Following is a detailed description of a preferred embodiment of formatter 110 including system environment. Appendix I shows a register map for the System for reference.

#### REGISTERS

All formatter 110 registers are accessed by microcontroller 105 one byte at a time. Registers that are larger than one byte are arranged in the memory map to match the processor's view of most significant byte to least significant byte. The memory map in microcontroller 105 places the lsbyte at the lower address (A+0) and the msbyte at the higher address (A+1). Table I shows the register map for formatter 110. A description of each register in Appendix I will be given.

**SCSI Select Input Register**  
The SCSI Select Input Register in SCSI interface 102 (also referred to hereafter as SCSI controller 102) allows the microcontroller 105 to read the three SCSI address select pins. It is a read only register.

Bits 7-13 not used. Read back as 0.  
Bit 2—SCSI Address Select Most Significant Bit.  
Bit 1—SCSI Address Select Next Significant Bit.  
Bit 0—SCSI Address Select Least Significant Bit.

When the Test Out bit in the Serdes Control Register 1458 is set, the SCSI address select pins become output pins. The level of the output signals are readable by microcontroller 105. However, the levels read by microcontroller 105 which are not useful for the signals may be actively changing during the read cycle.

Referring to FIG. 4, the registers of formatter 110 will be described below.

#### DMA Control Register 438

This is a read/write register. It is cleared upon reset. The bit descriptions follow.

Bit 7—Enable Interleaving—This bit, when set, enables the interleaved address generation circuitry 504. When clear, addresses are generated sequentially.

Bit 6—Enable Randomizing—This bit, when set, enables randomizing and derandomizing of data at 506 and 507 respectively as the data is transferred between DMA channel 132 and frame buffer 117 (also referred to hereafter as SRAM 117). When cleared, data is transferred without change.

**Bit 5—Throttle DMA.** When cleared, DMA grants will occur as fast as formatter 110 can generate them (1 byte/10 clocks; @ 24 MHz=2.4 Mbyte/second). When set, DMA will not be granted another SRAM 117 cycle until time for five other SRAM 117 cycles to occur has passed. This will slow down the byte transfer rate to 1.33 Mbyte/second @ 24 MHz (1 byte/18 clocks). This is necessary because main data buffer 104 (also referred to hereafter as DRAM 104) buffer manager 103 allocates the DRAM bandwidth between the Controller 102 and the formatter 110 and gives formatter 110 the priority. In order to prevent the SCSI transfer from stopping, formatter 110's DMA requests must be slowed down.

**Bit 4—Interleave Mode.** When cleared, the interleave will follow the DDS standard formulas, and "i" will be initialized to 1. If the data bytes are DMAed in the following order, D0, D1, D2, D3, D4, etc.; the DDS standard formulas put these bytes at A1L, A1U, B1L, B1U, A2L, etc. The header is interleaved and placed in memory by microcontroller 105. When this bit is set, the interleave will follow the DATA DAT standard formulas, and "i" will be initialized to 0. If the data bytes are DMAed in the following order, D0, D1, D2, D3, D4, etc.; the DATA DAT standard formulas put these bytes at A0U, A0L, B0U, B0L, A1U, etc. DATA DAT has no header on a frame. If interleaving is disabled and this bit is clear, addresses are generated sequentially starting from 4. If interleaving is disabled and this bit is set, addresses are generated sequentially starting from 0.

**Bit 3—Enable DMA.** This bit, when set, enables formatter 110 to generate DMA requests. When cleared, the DMA request line immediately goes inactive, but any current SRAM 117 accesses complete.

**Bit 2—RD/WR.** This bit, when set, reads data from DMA channel 132 and writes it to SRAM 117. When cleared, it reads data from SRAM 117 and writes it to DMA channel 132.

**Note:** Since formatter 110 generates the DMA request signal and has no byte counter in it when set up to read from DMA channel 132 and write to SRAM 117, it will request a byte even if DRAM buffer manager 103 thinks the DMA is complete.

**Bits 1-0—Buffer selection.** These bits select the quarter of SRAM buffer 117 to be used in the DMA transfer. They are used as the upper two bits of the address when accessing SRAM 117. Note that the maximum DMA transfer in normal mode is limited to 8 Kbytes. Larger transfers will "wrap" around. With no interleaving enabled, an address counter wraps around after transferring either 8192 or 8188 bytes of data (dependent on the state of the Interleave Mode bit). With DDS interleaving enabled, the address counter wraps around after transferring 6652 bytes of data. With DATA DAT interleaving enabled, the address counter wraps around after transferring 6656 bytes of data. The buffer selection bits address SRAM 117 as follows:

- 00: Frame buffer at SRAM 117 addresses 8000-9FFFh.
- 01: Frame buffer at SRAM 117 addresses A000-BFFFh.
- 10: Frame buffer at SRAM 117 addresses C000-DFFFh.
- 11: Frame buffer at SRAM 117 addresses E000-FFFFh.

(Note that writing to buffer 11 will overwrite the special reserved data areas.)

#### Track 1 checksum Register 440

This 16 bit read only register keeps a running XOR "sum" on the data sent over DMA channel 132. As bytes transfer over DMA channel 132, the first two bytes are skipped by this register. All bytes "skipped" are used by the track 2 checksum register 442, 443. The next byte is XORed into the least significant byte of the register. The next byte is XORed into the most significant byte of the register. The next byte is XORed into the least significant byte of the register. The next byte is XORed into the most significant byte of the register. Then four bytes are skipped. The sequence of XORing four bytes and skipping four bytes is repeated until the DMA transfer ends. Checksums are calculated on non-randomized data. Checksums are always calculated. Checksum registers are cleared upon reset and when writing to the initialize DMA register. Note that for checksums to be correctly calculated according to the ANSI specification, the checksums must be cleared before each DMA transfer. Also because the header is not being DMAed over, it must be XORed into the checksum before the result is placed in the pack item.

#### Initialize DMA Register 438

Writing any value to the initialize DMA register will initialize the DMA transfer logic. It will initialize the byte counter, address pointer, interleave counters, checksum registers, and randomizer circuits used in transferring the data between main data buffer 117 on the buffer manager 103 chip and SRAM 117 on formatter 110. This is a write only register. Note that the Interleave Mode in the DMA control register must be set before initializing the DMA logic. Otherwise the DMA address counters will not initialize correctly.

#### Start ECC Register

Writing any value to start ECC register will start microcontroller 105-initiated ECC process as specified in ECC control register 444. This is a write only register.

#### Track 2 Checksum Register 442

This 16 bit read only register keeps a running XOR "sum" on the data sent over the DMA channel. As bytes transfer over the DMA channel, the first byte is XORed into the least significant byte of the register. The second byte is XORed into the most significant byte of the register. Four bytes are skipped in the DMA transfer. All bytes "skipped" are used by the Track 1 Checksum Register 440, 441. The next byte is XORed into the least significant byte of the register. The next byte is XORed into the most significant byte of the register, then another four bytes are skipped. The sequence of XORing four bytes and skipping four bytes is repeated until the DMA transfer ends. Checksums are calculated on non-randomized data. Checksums are always calculated. Checksum registers 442, and 443 are cleared upon reset and when writing any value to the initialize DMA register. Note that for checksums to be correctly calculated according to the ANSI specification, the checksums must be cleared before each DMA transfer. Also because the header is not being DMAed over, it must be XORed into the checksum before the result is placed in the pack item.

#### Clear Arbitration Error Register

Writing any value to the clear arbitration error register will clear the arbitration error bit in the interrupt register 413. This is a write only register.

## 5,359,468

### 15

#### Bad C1 Block Counter 447

This read register is a counter. It is set to 0 by reset and by writing any value to this register. During normal reading from the tape and diagnostic read, it is equal to the count of the number of block pairs in the main data area which had non-zero C1 syndromes. Microcontroller 105 should clear it at the beginning of the read of each frame or track. It increments up to 255 and then rolls over to 0. This count can be used to indicate how many of the C1 block pointers are significant. Even though only the first 128 C1 block pointers detected since the last reset will be saved in memory, the counter will continue to count until reset. During read-after-write, for those block pairs in both the subdata area and the main data area which had non-zero syndromes, the eight bytes of syndromes associated with those blocks along with the block number are saved in SRAM 117 so microcontroller 105 can determine if they were single or multiple byte errors. The counter starts out at 0 and increments once for each byte saved in SRAM 117. In the end, it is equal to the (count) \*9 of the number of block pairs which had non-zero C1 syndromes. Microcontroller 105 should clear it at the beginning of the read of each frame or track. It increments up to a maximum of 128. A maximum of 14 full sets of 9 bytes will be saved. The 8 syndrome bytes are written to SRAM 117 first followed by the block number. See the "SRAM 117 Memory Map" section entitled "Definition of entries" for further information on the locations of the syndromes and block numbers and the format of the block number bytes.

#### Read Track Checksum Register 260

During normal reading from the tape, diagnostic reads, and read-after-write, the data will be XORed into this 16 bit register to form a track checksum. ECC parity bytes will be skipped over and the correct bytes will be XORed in the high and low bytes of this register. It can be used to double check the data read. Note that errors detected in ECC must be taken into account when using the value calculated by this register. The read track checksum low byte register is a read only register and is cleared by writing to the read track checksum high byte register. The read track checksum high byte register is a read/clear register and is cleared by writing to the read track checksum high byte register. Microcontroller 105 should clear it at the start of each normal read, diagnostic read, and read-after-write of a track.

#### C2 Block Address Register 446

This register is a read only register and points to the block which just had non-zero ECC syndrome bytes saved in memory. Reading this register will clear the C2 Status bit in interrupt control register 413. This register only has significant information if the C2 Status bit is set and the non-zero C2 syndrome detected bit set and before restarting the ECC syndrome generation. This register contains an offset from the start of the track in frame buffer 117 to the first byte of the bad block. The register allows read access to a counter used to generate the addresses accessed while doing the microcontroller 105-initiated ECC syndrome and parity generation. If ECC syndrome or parity generation is in progress, writing a zero to the enable processor initiated ECC bit in ECC control register 444 will terminate the operation at the next block boundary. At reset and when operation is completed or terminated, this register is set to 00h. During microcontroller 105 initiated C1 generation, the value in this register is not used by microcontroller

### 16

105 as sequence 206 does not stop when non-zero syndromes are detected. The counter may be changing when microcontroller 105 tries to read it. Bit 7 is not used and is always read back as 0. Bits 6-0 is offset, and 5 offset from the start of a track in frame buffer 117.

For C2 ECC generation, this register "counts" in the following fashion: 00h, 40h, 01h, 41h, 02h, . . . 1Fh, 5Fh, 20h, 60h, . . . 36h, 76h, 37h, 77h. The hardware contains a normal counter which counts from 00h to 6Fh. The relationship of the counter's outputs to the address generated and value read from the register are as follows:

Address Bits:	6	5	4	3	2	1	0
Counter Outputs:	Q0	Q6	Q5	Q4	Q3	Q2	Q1

If the offset is 00h, the bytes of the block worked on are at offsets: Byte0 0000; byte1 0080; byte 2 0100; byte3 0180; byte 4 0200; byte 5 0280; byte 6 0300; byte7 0380; 20 byte 8 0400; byte9 0480; byte10 0500; byte 11 0580; byte 12 0600; byte 13 0680 parity byte P5/syndrome byte S5; byte14 0700 parity byte P4/syndrome byte S4; byte15 0780 parity byte P3/syndrome byte S3; byte16 parity byte P2/syndrome byte S2; byte17 0880 parity byte P1/syndrome byte S1; byte18 0900 parity byte P0/syndrome byte S0; byte19 0980; byte20 0A00; byte21 0A80; byte22 0B00; byte23 0B80; byte24 0C00; byte 25 0C80; byte26 0D00; byte27 0D80; byte28 0E00; byte29 0E80; byte 30 0F00; byte 31 0F80. If the offset is 77h, the bytes of the block worked on are at offsets: byte0 0077; byte1 00F7; byte2 0177; byte3 01F7; byte4 02F7 . . . byte29 0EF7; byte30 0F77; byte31 0FF7.

#### ECC Control Register 444

This is a read/write register. It is cleared upon reset. Following are the preferred bit settings.

Bit 7—Software Reset. When set, formatter 110 goes to its reset state. When clear, formatter 110 is not reset. This bit is set by an external reset or a microcontroller 105 write of a 1 to this bit. All other logic "cleared by reset" is cleared by both an external reset and a software reset.

Bit 6—Enable On-the-fly C1 Generation. When set, C1 parity and syndrome generation is done on the fly as data is sent to/received from the serdes. The operation, reading or writing, determines whether syndromes or parity is generated. When cleared and reading, the parity bytes read from tape are sent to memory. When cleared and writing, the parity bytes read from memory are written to the tape. Note that there is only one set of C1 generation logic so C1 can not be generated on the fly and at the same time be generated by microcontroller 105 initiation.

Bit 5—Enable Processor Initiated ECC. When clear, the "C1 and C2 ECC to Memory" sequencers are stopped and set to their initial state. When set, the ECC specified by bits 4-1 is enabled and can be started by writing any value to the Start ECC Register. An ECC process already in progress can be terminated by clearing this bit. Bits 4-1 are significant only when this bit is set.

Bit 4—C1/C2. When clear, C1 ECC generation is performed. When set, C2 ECC generation is performed. Note that this bit must be set up prior to setting the "Enable Processor Initiated ECC" bit. This bit and the "Enable Processor Initiated ECC" bit are not to be changed at the same time. Note that if this bit is clear and the "Enable Processor Initiated ECC" bit is set,

then the serdes can do nothing else; it does not matter whether the ECC has been "started" or not.

**Bit 3—Syndrome/Parity.** When clear, parity bytes are generated. When set, syndrome bytes are generated.

**Bit 2—Subdata/Main Data.** When clear, ECC is generated over the track buffer area controlled by bit 1 of this register and bits 1 and 0 of DMA Control Register 438 and place the results in the appropriate memory locations. When set, ECC is generated over the subdata areas of the frame pointed to by bits 1 and 0 of DMA Control Register 438 (three block pairs are processed). Note that only C1 can be done on the subdata areas. This bit is a "don't care" when C2 is selected; C2 always works on main data.

**Bit 1—Track A/B.** When clear, ECC is generated over track A of frame buffer 117 pointed to by bits 1 and 0 of DMA Control Register 438. When set, ECC is generated over track B. When C1 and Subdata is selected, this bit is a "don't care".

**Bit 0—Force Arbitration Error.** When this bit is set, the Arbitration Error is forced active. When this bit is set, the Arbitration Error works normally and if active can be cleared by writing to the Clear Arbitration Error Register.

Examples for using ECC control register 444 follows:  
Set this register to the following value in order to:

01x1.xxxx	1. Generate C1 parity on fly when writing
01x1.xxxx	2. Generate C1 syndrome on fly when reading
0010.000x	3. Generate C1 parity over main data track A and write values to SRAM 117
0010.001x	4. Generate C1 parity over main data track B and write values to SRAM 117.
0010.100x	5. Generate C1 syndrome over main data track A and write values to SRAM 117.
0010.101x	6. Generate C1 syndrome over main data track B and write values to SRAM 117.
0010.01xx	7. Generate C1 parity over write and read subdata areas and write values to SRAM 117.
0010.11xx	8. Generate C1 syndrome over write and read subdata areas and write values to SRAM 117.
0111.0x0x	9. Generate C2 parity over main data track A and write to SRAM 117 while C1 is done on the fly.
0111.0x1x	10. Generate C2 parity over main data track B and write to SRAM 117 while C1 is done on the fly.
0111.1x0x	11. Generate C2 syndrome over main data track A and write to SRAM 117 while C1 is done on the fly.
0111.1x1x	12. Generate C2 syndrome over main data track B and write to SRAM 117 while C1 is done on the fly.
00xx.xxxx	13. When writing, send data to serdes without generating C2 parity, instead of skipping over parity locations in SRAM 117, read the values in those locations and send them as the parity bytes.
00xx.xxxx	14. When reading, read data from serdes sending the parity bytes read from the tape to memory.

#### Serdes Control Register 0 457

This is a read/write register. It is cleared upon reset. The value read back is controlled by the Read Select bit in Serdes Control Register 1 458. No bits are auto cleared.

**Bit 7—Read Enable.** In general, this bit, when set, enables reading. This bit along with the Control Mode bits in Serdes Control Register 1 458 determine exactly what action occurs. This bit is latched on the synchronized changing edge of the TRKCLK signal. It can be

set up for the next operation after the TRKCLK signal has transitioned.

**Bit 6—Write Enable.** In general, this bit, when set, enables writing. This bit along with the Control Mode bits in Serdes Control Register 1 458 determine exactly what action occurs. This bit is latched on the synchronized changing edge of the TRKCLK signal. It can be set up for the next operation after the TRKCLK signal has transitioned.

**Bit 5—Enable Write to Main Data Only.** When set, writing to the Pre-amble 2 and Main Data areas are enabled. When cleared, a whole track is written. This bit is latched on the synchronized changing edge of the TRKCLK signal. It can be set up for the next operation after the TRKCLK signal has transitioned.

**Bit 4—Test ATF Area 2 Select.** This bit is used for hardware testing only. The Write ATF Text bit in the Serdes Control Register 1 458 must be set when using this bit. When set, the ATF pattern for ATF Area 2 is generated. When clear, the ATF pattern for ATF Area 1 is generated.

**Bit 3—Odd Frame.** When clear, an even numbered (absolute number) frame is being accessed. When set, an odd numbered frame is being accessed. This is needed so the appropriate ATF signals will be generated on write. This bit is latched at the same time as bits 1 and 0. This bit is not used during reading or read after write.

**Bit 2 Track A/B.** When clear, use track A of the buffer pointed to by bits 1 and 0 of this register. When set, use track B. This bit is latched on the synchronized changing edge of the TRKCLK signal. It can be set up for the next operation after the TRKCLK signal has transitioned.

**Bits 1–0—Buffer selection.** These bits select which frame, subdata, ID, and Bad C1 pointer buffers are used in the transfer to/from the serdes. The bits are used as the upper two bits of the address when accessing SRAM 117. Note that the fourth quarter, both bits set, should not be used because it contains areas reserved for special information used during reading and writing to the tape. During normal write, diagnostic write, and raw code write, these bits are latched 13 block times (at the start of the subdata area) after the synchronized changing edge of the TRKCLK signal. During raw code write and read, these bits are really a "don't care" since data comes from/goes to the DMA. During normal read and diagnostic read, these bits are latched on the changing edge of the TRKCLK signal. NOTE: Bit 0 is used with bit 2 of the Serdes Control Register 1 458 in a special test mode.

Further clarification of what bits 7 and 6 enable follows:

#### NORMAL MODE:

**76**  
**00 Sequencer is turned off.**  
**01 Write:** Enable writing date to tape from the buffers pointed to by bits 2–0.

**10 Read:** Enable reading data from tape to the buffers pointed to by bits 2–0.

**11 RAW:** Enable Read-After-Write from tape to the buffers pointed to by bits 2–0.

During Write, subdata is taken from the "Subdata write data" buffer. IDs are taken from the "W1 eight block cycle" and four byte "SW1 SW2 for even and odd blocks" buffers. And main data from the half of frame buffer 117 for the selected track.

During Read, subdata is written to the "Subdata read data" buffer for the selected track. The IDs (sets of 3

bytes) are written to the "ID Entry for Subdata blocks" and ID Entry for Main data blocks" buffers for the selected track. And main data is written to the halo of the frame buffer 117 for the selected track. And Bad C1 pointers are written to the "Bad C1 Block Pointer" buffer.

During RAW, subdata is written to the "Subdata read data" buffer for the selected track. The IDs (sets of 3 bytes) are written to the "ID Entry for Subdata blocks" and ID Entry for Main data blocks" buffers for the selected track. And Bad C1 pointers and syndromes are written to the "Bad C1 Block Pointer" buffer. Main data is not written anywhere.

**DIAGNOSTIC MODE:** Through the use of both Diagnostic Write and Diagnostic Read, complete SERDES loop-back testing can be accomplished. Data transfers to/from main data buffer 104 only while the FMTT1 signal is active.

76

00 Sequencer 406 is turned off.

01 Diagnostic Write: Enable reading data from the buffers pointed to by bits 2-0 and writing code to buffer manager 405. During diagnostic write, data is read from the selected buffer in the SRAM 117 and processed by sequencer 406 just like it would be written to the tape, but instead of writing it to the tape, the raw code is serialized and sent over DMA channel 132 to main data buffer 104.

10 Diagnostic Read: Enable reading code from the buffer manager and writing data to the buffers pointed to by bits 2-0. During diagnostic read, raw code is read from DMA channel 132 and is sent to the serdes where it is processed by sequence 406 just as it had been read from the tape.

11 Diagnostic Read After Write: Enable reading code from buffer manager 405 and writing data to the buffers pointed to by bits 2-0. During diagnostic read, raw code is read from DMA channel 132 and is sent to the serdes where it is processed by sequencer 406 just as it had been read from the tape.

**RAW CODE MODE:** This mode provides raw code transfers to/from the tape. Data transfer to/from the buffer manager is gated by the FMTT1 signal allowing microcontroller 105 to select what data it wants to read or write.

76

00 Sequencer 406 is turned off.

01 Raw Code Write: Enable writing code from the buffer manager directly to the tape. During raw code write, raw code is read over DMA channel 132 from the main data buffer 104 serialized, and written out to tape.

10 Raw Code Read: Enable reading code from the tape and writing it to buffer manager 405. During raw code read, raw data is read from the tape, serialized and sent over DMA channel 132 to main data buffer 104.

11 Invalid combination. Sequencer 406 is turned off.

**SEARCH MODE:** Searching for the sync mark to indicate start of a block is gated by the FMT-T1 signal allowing microcontroller 105 to search for only subdata blocks. Blocks will always be stored in SRAM 117 on 32 byte boundaries. Once a sync is detected, the ID bytes are skipped and the next 32 bytes of data will be sent to the buffer in SRAM 117. This data may not be good if a track crossing occurs.

76

00 Sequencer is turned off.

01 Invalid combination. Sequencer 406 is turned off.

5 10 Enable reading data blocks to the track buffers pointed to by bits 2-0.

11 Invalid combination. Sequencer 406 is turned off. Serdes Control Register 1 458

This is a read/write register. It is cleared upon reset. Bits 7-14 are active on any read cycle but typically should only be used during a diagnostic read.

Bit 7—Disable Serdes Clock. When clear, the Serdes runs off of the ReadClock signal from data separator 118 during all reading (except diagnostic read) and the write clock (crystal) during all writing and during diagnostic read. When set, the Serdes can only be clocked by writing to the bit pointer register 462.

Bit 6—Stop on Code Violation. When set, sequencer/serdes 406 will stop when the 10 bit to 8 bit demodulator ROM detects a code violation.

Bit 5—Stop on Sync Detect. When set, Sequencer/serdes 406 will stop when the sync symbol (10 bits) is detected.

20 20 Bit 4—Read Select. This bit, when set, selects the latched data when reading from Serdes Control Register 0 457. This bit, when clear, selects the unlatched data when reading from Serdes Control Register 0 457.

Bit 3—Test Out. When set, the SL0, SL1, SL2, NTRIS, and NACTI become debug outputs as described below.

Bit 2—Write ATF Test. This bit is used for hardware testing only. It should always be zero for normal operation. When set, data for the ATF area selected by Bit 4 in the Serdes Control Register 0 457 will be output on the WDAT pin.

Bits 1-0—Control Mode. These two bits determine how serdes handles the data and control where the data paths are selected.

10 00 Normal Mode.  
01 Diagnostic Mode.  
10 Raw Code Mode.  
11 Search Mode.  
40 Status Register 459

This register is a read/clear register. Writing any value to this register will clear bits 7-5. This register is cleared by reset.

45 Bit 7—NonZero C1 Syndrome Detected. When set, this bit areas were detected since the last time this bit was cleared.

Bit 6—Started Main Data on Block 1. When set, this bit indicates when the first sync mark was detected, the parity for the ID indicated the ID was good and the ID indicated that it was for block 1 (the second block) of the track. This bit stays set until cleared by microcontroller 105.

50 Bit 5—Sync Missed. When set, this bit indicates that after detecting sync, a subsequent sync mark was not detected for one of the main data blocks. This bit stays set until cleared by microcontroller 105.

Bit 4—Sync Detected. When set, this bit indicates that a sync mark was detected in the data stream. It is cleared by reset and by decoder 454 when the sync mark has passed. During normal reads, it is set for only one code bit time.

60 Bit 3—Code Violation Detected. When set, this bit indicates that a code violation was detected in decoder 454 when data was expected. It is cleared by reset and by decoder 454 when the violation passes. It could be set only for 10 code bit times (1 code byte/symbol time).

65 Bits 2-0—Unused. Reads back as 0.

Bit Pointer 462

## 5,359,468

**21**

Through this register, microcontroller 105 has access to the current bit number of the current byte the serdes is processing during diagnostic read. This is a read register. Writing any value to this register will pulse the bit clock. This is useful as a diagnostic and test feature.

Bits 7-4—Bit Number in code symbol. One of ten possible combinations to indicate the bit number. It counts in the following sequence: 0,1,3,2,4,8,9,B,A,C.

Bit 3—Not Used. Reads back as 0.

Bits 2-0—Bit Number in byte from DMA. Count from 0 to 7 to indicate which bit of the byte from DMA was just shifted in.

**Byte Counter 465**

Byte counter 465 counts the bytes/symbols in a block in 36 combinations. The counter is cleared by sequencer 406 at the start of each block. It is a read only register.

Bits 7-6—Not Used. Read back as 0.

Bits 5-0—Byte/Symbol Number. It counts in the following sequence: 00,03,04,01,02,05,18,1B,1C,19,1A,1D,20,23,24,21,22,25,08,0B,0C,09,0A,0D,10,13,14,11,12,15,28,2B,2C,29,2A,2D.

**Block Counter 465**

Block counter 465 counts the number of blocks in a state from 0 to a maximum of 127. The counter is cleared by sequencer 406 at the start of a state in normal operation. It is a read register. It is used to fill in the block address field (bits 6-0) of W2 of the MAIN ID. In search mode, microcontroller 105 will use this register to indicate how many blocks have been read in since the count was reset. When in search mode, microcontroller 105 can clear this counter by writing any value to this register. When not in search mode, only sequencer 406 or reset can clear it. When not in search mode, writing any value to this register will increment the block count. This feature was added to make it easier to test the part with a small number of test vectors. This feature is not used in normal operation.

Bit 7—Not Used. Read back as 0.

Bits 6-0—Block Number.

**State Sequence**

There is a state sequence for sequencer 406 which counts for each unique area in a track. The counter is cleared by the sequencer at the start of a track. It is read only register.

Bit 7—Amble Area. Set when in an amble area of the track.

Bit 6—Subdata Area. Set when in a subdata area of the track.

Bit 5—IBG Area. Set when in an IBG area of the track.

Bit 4—ATF Area. Set when in an ATF area of the track.

Bit 3—Main Data Area. Set when in the main data area of the track.

Bits 2-0—Section Number. The “sections” start counting at 0. Example: There are four IBG areas in a track. If the IBG Area bit is set and the section number is 2 then we are in the third IBG area of the track.

During Write operation, sequencer 406 will go through all of the areas in the proper order. During Read operations, sequencer 406 will never go through the ATF area, and it will mainly be in either the Subdata or Main Data Area.

**Interrupt Register 413**

The bits in Interrupt Register 413 are unique and described below. All bits are readable. Some are writable.

**22**

Bit 7—SCSI Controller Interrupt Request. This is a level active interrupt.

This is an external interrupt bit, reading a 1 corresponds to an interrupt condition. Writing will do nothing. The interrupt must be cleared or masked at the interrupt source.

Bit 6—DRAM Buffer Manager Interrupt Request. This is a level active interrupt.

This is an external interrupt bit, reading a 1 corresponds to an interrupt condition. Writing will do nothing. The interrupt must be cleared or masked at the interrupt source.

Bit 5—Code. When this read/write bit is set and the Special Address Space Decode Enable bit is set, all addresses are decoded as if in the CODE space. When clear and the Special Address Space Decode Enable bit is set, all addresses are decoded as if in the DATA space.

Bit 4—Special Address Space Decode Enable. When this read/write bit is set, the address decode ignores the INST pin and instead uses the CODE bit in place of the INST pin. When clear, the INST pin determines whether CODE or DATA space is decoded.

Bit 3—Arbitration Error Interrupt Mask. When this read/write bit is clear, the Arbitration Error interrupt is masked. When set, the Arbitration Error interrupt is enabled.

Bit 2—Arbitration Error Interrupt. When clear, the Arbitration Error interrupt is not active. When set, the Arbitration Error interrupt is active. The Arbitration Error interrupt indicates that two grants were issued by the SRAM 117. Arbitor 235 logic at the same time. This error will occur only for a bad chip. This interrupt can be cleared by writing any value to the Clear Arbitration Error Register.

Bit 1—Non-Zero C2 Syndromes Detected/C2 Syndrome Generation Complete. When set, non-zero ECC syndromes where detected and just written to memory. The ECC Block Pointer Register 444 contains the address of the block immediately following the block with non-zero syndromes. When this bit is clear and bit 0 is set, the ECC Syndrome Generation sequence has completed for the track specified. This is a status only bit. No interrupts occur when set. It is read only and is cleared on reset and when the ECC Start Register is written to.

Bit 0—C2 Status. This status bit is set when the C2 parity or syndrome generation is complete, or immediately after non-zero C2 syndromes were detected and written to memory and sequence 405 stopped. See bit 1 for status as to which event caused this. Both events will not occur at the same time. It is read only and cleared by reset and reading from the C2 Block Address Register 446.

55 Serial Port Address/Status Register 416

This register contains status, control and address bits that relate to the state of the Serial Part which is used to access registers in the ATF as described in ATF application Ser. No. 07/741,088, which is hereby incorporated by reference.

The address portion of this register specifies the address of the register in the ATF which is the source or destination for access via the serial port. Microcontroller 105 can safely read and write this register whenever it wants (as long as the serial port is READY); serial port transfers are actually initialized by accesses to the Serial Port Data Register 417. Writing to this register or the Serial Port Data Register 417 while the

READY bit is clear is internally inhibited and has no effect. This is a read/write register but has some read only bits in it.

Bit 7—READY. This bit is 1 when the Serial Port is ready to accept a new request. This bit is 0 when the Serial Port is busy transferring data. The Serial Port Address Register 416 and the Serial Port Data Register 417 should only be accessed when READY is 1. For each transfer, READY goes low for 70 clocks (2.8 microseconds).

Bit 6—Serial Port Parity Error. This bit is 1 if the Serial Port detects a parity error during a read or write transfer. During a write transfer, parity is generated for the data on the serial data line. (This should never happen unless there is a hardware problem in the system.) It is automatically cleared to 0 any time microcontroller 105 writes to the Serial Port Address/Status 416. Note that this bit is only valid when READY is 1 and will toggle during the transfer.

Bit 5—Diagnostic Loopback. Bits 0-7 of the Serial Port Data Register 417, bits 0-4 of the Serial Port Address/Status Register 416 and several control bits (R/W, start bit, stop bit, and parity bit) all form one long shift register. When a serial access occurs, the bits are shifted out and other bits are shifted in. If the Diagnostic Loopback bit is clear, then the state of the Serial Data I/O pin "SDAT" is shifted in. This should match what was shifted out unless there is a hardware error or conflicts on driving the pin. Also, on a read access, the data portion will come from the other chip. If the Diagnostic Loopback bit is set, the data being shifted out is used as the data shifted in. This is done internally and the "SDAT" pin is forced high throughout the access. With the bit set, the hardware can be tested independent of the state of the "SDAT" pin and signal line.

Bits 4-0—Serial port access address. See ATF related application Ser. No. 07/741,088 for details.

#### Serial Port Data Register 417

This 8-bit read/write register is used to access registers in the ATF via the serial port. Note that the serial nature of the access is invisible from a processor viewpoint; microcontroller 105 simply does normal accesses to this register with certain timing and protocol constraints. This is a read/write register.

When microcontroller 105 wants to write to a register in the ATF, it sets up the appropriate address in formatter 110 Serial Port Address register 416 and then writes the data to the Serial Port Data Register. When microcontroller 105 writes to Serial Port Data Register 417, formatter 110 hardware performs the following sequence:

1. The data from microcontroller 105 is latched into the Serial Port Data Register 417 and the bus cycle is terminated. (i.e., this is a no-wait state cycle for microcontroller 105.)

2. The READY bit in the Serial Port Status Register 417 is cleared.

3. The contents of the Serial Port Address Register 416 are transferred to the ATF over the serial link, along with a "Write" bit. 6 bits of information are transferred.

4. The contents of Serial Port Data Register 417 are transferred to the ATF over the serial link, along with parity covering the entire 14 bit transfer. (Odd parity is used, i.e., all 0's is a parity error.) After the ATF confirms that the proper parity was received, it moves the data into the addressed register.

5. Formatter 110 ASIC checks parity on the transfer; if there is an error, the Serial Port Parity Error bit in the Serial Port Status Register 416 is set.

6. When the transfer is complete, the READY bit in the Serial Port Control Register is set. The entire write transfer takes 70 clocks (2.8 microseconds).

7. Bits 0-4 in the Address register and bits 0-7 in the Data register are updated during the transfer with the data that actually appears on the pin.

10 When microcontroller 105 wants to read from a register in the ATF, it sets up the appropriate address in formatter 110 Serial Port Address register 416 and then reads the Serial Port Data Register 417 TWICE. Microcontroller 105 must do two reads because the first read cycle requests the serial port to transfer the data up from the ATF and deposit it in the Serial Port Data Register 417; the second read gets the actual data. Successive read accesses may be pipelined, since the second read will also initiate a new transfer of data up from ATF.

When Microcontroller 105 reads from the Serial Port Data Register 417 formatter 110 hardware performs the following sequence:

1. The data currently in the Serial Port Data Register 417 is returned to microcontroller 105 and the bus cycle is terminated. (i.e., this is a no-wait state cycle for microcontroller 105.)

2. The READY bit in the Serial Port Status Register 416 is cleared.

3. The contents of the Serial Port Address Register 416 are transferred to the ATF over the serial link, along with a "Read" bit. 6 bits of information are transferred.

4. The contents of the addressed register in the ATF are transferred over the serial link to the Serial Port Data Register 417. The ATF also sends a parity bit covering the entire 14 bit transfer. (Odd parity is used, i.e., all 0's is a parity error.)

5. Formatter 110 checks parity on the transfer; if there is an error, the Serial Port Parity Error bit in the Serial Port Status Register 416 is set.

6. When the transfer is complete, the READY bit in the Serial Port Control Register is set (regardless of parity errors). The entire read transfer takes 70 clocks (3.7 microseconds).

7. Bits 0-4 in the Address register are updated during the transfer with the data that actually appears on the pin.

50 420 Galois Multiplier Value 1, Value 2, Result 418, 419,

These registers provide microcontroller 105 with a quick and simple way to do Galois Field multiplication. Having this capability can speed up the ECC correction algorithms. The Galois Field matches the field defined for C1, C2 and C3. Value 1 and Value 2 are multiplied to produce the Result. Value 1 and Value 2 are read/write registers. Result is a read only register. All three registers are a full eight bits. The contents of the two Value registers are combined logically to produce the result. The result is available immediately after the values are written into the Value registers.

The Galois Field is GF(2<sup>8</sup>) with generator polynomial;

$$g(x)=x^8+x^4+x^3+x^2+1$$

A primitive element "a" in GF(2<sup>8</sup>) is defined as follows:

$$\begin{array}{l} "a" = (0\ 0\ 0\ 0\ 0\ 0\ 1\ 0) \\ \quad a^7a^6a^5a^4a^3a^2a^1a^0 \end{array}$$

#### Galois Square Root

This read only register provide microcontroller 105 with a quick and simple way to do Galois Field Square Roots. Having this capability can speed up the ECC correction algorithms. The Galois Field matches the field defined for C1, C2 and C3. The Galois Square Root is taken of the value in the Galois Multiplier Value 1 Register. The contents of the Value register is combined logically to produce the result. The result is available immediately after the value is written into the 15 Value register.

#### RESOURCES

Aside from the above-described registers, other available resources in this embodiment of formatter 110 are described following. These resources include a system memory map, wait state generation, frame buffer 117 (SRAM in the preferred embodiment) access, frame buffer 117 memory map, arbitrator 235 priorities, interleave/deinterleaving, randomizing/derandomizing, ECC resources, and serdes.

#### System Memory Map

Formatter 110 decodes microcontroller 105's address space and produces chip selects for ROM 107, DRAM buffer manager 103, SCSI controller 102, data compression chip, formatter registers, SRAM 117, and ATF 30 serial registers. Microcontroller 105 memory map is shown in Appendix IV.

In one embodiment of the present invention, microcontroller 105 is an 80C198 microprocessor which has an address space of 64K bytes and a signal pin "INST" which signals when an instruction byte is being fetched. In the present invention, the address space is doubled by using the INST pin and the "Special Address Space Decode Enable" and "Code" bits in the interrupt register to create a 64K byte CODE space and 40 a 64K byte DATA space. This allows the system to include 63.5K bytes of firmware in CODE space as well as providing direct address mapping of the 32K byte frame buffer in DATA space. The "Special Address Space Decode Enable" and "Code" bits in the interrupt 45 register provide flexible addressing modes so that special access cases can be handled, such as writing to CODE space to support flash EPROM and executing from DATA space to support downloaded diagnostic firmware routines. Addresses are decoded into CODE 50 and DATA space in the following manner:

Special Address Space Decode			
Enable bit	Code bit	INST pin	Decoded Space
0	X	0	DATA
0	X	1	CODE
1	0	X	DATA
1	1	X	CODE

#### Wait state generation

Formatter 110 produces the READY signal going to microcontroller 105. The READY signal controls whether microcontroller 105 inserts wait states to extend its bus cycle. No wait states will be inserted for 65 ROM cycles or Formatter register cycles. Access to the main buffer and SCSI controller 102 is controlled by the DRAM buffer manager 103. DRAM buffer manager

103 produces the NSWAIT signal when the bus cycle must be extended for accesses to SCSI controller 102. Formatter 110 must extend the NSWAIT signal an additional 4 clocks for SCSI Controller 102 accesses.

5 Formatter 110 must force the ready line low for 5 clocks for Data Compression chip accesses to allow the chip time to produce its own wait signal. Processor accesses to SRAM 117 will cause wait states to be added to the bus cycles. Chip selects for DRAM buffer manager 103 and SCSI controller 102 are not ALE qualified in order to allow the DRAM buffer manager 103 time to produce its wait signal. The Data Compression chip select signal also is not ALE qualified.

#### SRAM Access

For each access, Formatter 110 will hold off microcontroller 105 until SRAM 117 can be accessed. Because formatter 110 must arbitrate between many requests for access to SRAM 117, the number of clocks RDY would be low and could vary from 8 to 35 with 11 being the typical case.

#### SRAM 117 Memory Map

In the preferred embodiment, 32K byte SRAM 117 is partitioned into 4 8K byte sections. The first, second, and third sections are reserved (in normal use) for the triple buffering of read and write data. Each section can store one frame's worth of user data and C1/C2 parity/-syndromes. The fourth section is used as a storage area for both microcontroller 105 and formatter 110. All unspecified areas are free for use by microcontroller 105. See Appendix V for SRAM 117 addresses.

#### Definitions of entries

Bad C1 Block Pointer entries are 1 byte each as shown below.

Bit 7—Subdata Block Pair. This bit when set indicates the block number is from the subdata area. When clear, the block number is from the main data area.

Bit 6—Track B. This bit when set indicates the block number is from track B. When clear, the block number is from track A.

Bit 5-0—Block Pair Number. There are 64 block pairs in the main data area of a track. These six bits give the block pair number 0-63 of the pair with non-zero C1 syndromes. Note that the 6 bits of the block pair number are the six most significant bits of the block number for the track. For Subdata block pairs, these bits go from 0-3 for the first subdata area on a track and from 0-3 again for the second subdata area on a track.

ID Entries are three bytes each as shown below.

0 W1 or SW1

1 W2 or SW2

2 Parity

SW2 Entries for writing are 1 byte. Bits 0-3 are filled in by the hardware with the block address. Bits 0-3 should have zeros written to them in SRAM 117. Bits 4-7 are used as read from the SRAM 117.

Note that W2 entries for writing are not stored in SRAM 117 because the whole byte is "filled in" by the hardware with the block address (bits 0-6) and bit 7 is always zero.

Byte usage of Scratch Area by the Formatter 110 are:

192 byte for subdata (write) info (triple buffered)

3 byte of SW1 for even blocks (triple buffered)

3 byte of SW1 for odd blocks (triple buffered)

3 byte of SW2 for even blocks (triple buffered)

3 byte of SW2 for odd blocks (triple buffered)

24 bytes of W1 for the cycle it goes through (triple buffered)

180 bytes of ID (3/blk, 2 blk subdata, 8 blk main data, triple buffered)  
 384 2 blocks/track of subdata (triple buffered)  
 384 byte of bad C1 block information (triple buffered)  
 This results in 1176 total used (14.3%) out of 8192 total available, leaving 7016 bytes available for microcontroller 105 usage.

#### Arbitration priorities

Formatter 110 arbitrates access to SRAM 117 by looking at access requests and generating grants/acknowledgements. The hard wired priority is shown below.

Highest: SERDES/"on the fly" C1 syndrome. The SERDES has a maximum transfer rate of 1236K byte/second (for 3% speed variation on 1200K byte/sec). The "on the fly" C1 syndrome has a burst transfer rate of 8M byte/second (1 byte/3 clocks) over 9 bytes for every 72 bytes read from the tape. By allowing the syndromes to monopolize the SRAM 117 accesses, the syndromes will be written to SRAM 117 before the next data byte is ready for ECC processing.

Next highest: DMA channel 132. Formatter 110 has a maximum transfer rate of 2.4M byte/second (1 byte/10 clocks). It can be throttled to 1.33M byte/sec (1 byte/18 clocks).

#### Next highest: Microcontroller 105.

Lowest: Processor Initiated ECC Parity/Syndrome generation. The ECC circuitry will use up all available cycles granted to it once it is started. Typically, when the circuitry is started, it will only have to share it with the SERDES channel. DMA channel 132 and processor tasks are sequenced to eliminate overlap between them and the ECC tasks.

SRAM 117 has a maximum transfer rate of 8.0M bytes/second. There is an arbiter checker which will be set whenever more than two grants were issued simultaneously. When set, a maskable interrupt is generated.

#### Interleave/uninterleaving

Interleaving occurs via the formulas found in the DDS and DATA/DAT standards documents incorporated by reference. In the DDS format, microcontroller 105 sending the four header bytes itself, interleaving on DMA transfers start with i=1 in the formula. In the DATA/DAT format, interleaving starts at i=0, and upper/lower byte ordering is reversed. See the DMA Control Register section above for interleaving controls.

#### Randomizing/unrandomizing

Randomizing occurs when data is transferred from the main buffer into SRAM 117. A randomizing pattern, specified by the DDS and DATA/DAT standards documents, is XORed with the data received over DMA channel 132. This happens before any interleaving occurs. Unrandomizing occurs when data is transferred from SRAM 117 to main data buffer 104. It occurs after deinterleaving and before driving the data on DMA bus channel 132. Note that since randomizing is XORing a pattern with the data, XORing the randomized data with the same pattern results in the original data. Randomizing can be disabled via DMA control register 438.

#### ECC Resources

The preferred embodiment of formatter 110 includes circuitry to generate parity bytes and syndrome bytes for the C1 and C2 ECC polynomials is described above. No circuitry is included for doing hardware error correction. Galois Multiplier and Galois Square Root functions are included to aid microcontroller 105 in analyz-

ing the syndromes. The bad C1 Block data and RAW syndrome information is stored on SRAM 117 to aid microcontroller 105 in correcting small numbers of C1 errors per frame. For correcting larger amounts of C1 errors and all C2 errors. Microcontroller 105 must examine the syndrome bytes stored in SRAM 117 to determine if any errors occurred over the track. The same memory area is used for both Bad C1 Block data and RAW syndrome information. When normally reading, the Bad C1 Block data for both subdata and main data is stored in 128 bytes. The bytes are block pair numbers of blocks which had non-zero C1 syndromes. Microcontroller 105 must look at the syndromes to determine if the even or odd bytes or both had non-zero syndromes. Under Read-After-Write operation, the syndromes and block numbers of the first fourteen block pairs with non-zero syndromes are stored to help microcontroller 105 determine if the frame failed and should be rewritten. These block pairs come from both the subdata and main data areas. Again, microcontroller 105 must look at the syndromes to determine if the even or odd bytes or both had non-zero syndromes. In SRAM 117, first the block number is stored followed by the 8 syndrome bytes. The eight syndrome bytes are in the following order: S3 Even, S2 Even, S1 Even, S0 Even, S3 Odd, S2 Odd, S1 Odd, S0 Odd. See the Bad C1 Block Counter Register 447 description above for further details. For C1 processor-initiated parity generation, the eight parity bytes are written to SRAM 117 in the following order: S3 Even, S3 Odd, S2 Even, S2 Odd, S1 Even, S1 Odd, S0 Even, S0 Odd. For both C1 on the fly and processor-initiated C1 syndrome generation, the eight syndrome bytes are written to memory in the following order: S3 Even, S2 Even, S1 Even, S0 Even, S3 Odd, S2 Odd, S1 Odd, S0 Odd. Note that the ordering is different from parity generation. When C1 on the Fly is turned off, the C1 parity and syndrome logic is disabled. When writing (Normal Write or Diagnostic Write), the parity is read from the eight bytes at the end of each block pair and written to tape. When reading (Normal Read, Read After Write, or Diagnostic Read), the parity read from tape is written to the eight bytes at the end of each block pair. Syndromes are not generated. In preparing a frame or track for writing with C1 on the fly turned off, processor-initiated C1 can be used to generate correct parity. In the same way, if a frame or track was read in with C1 on the fly turned off, processor-initiated C1 can be used to generate the syndromes to test if any errors occurred and bytes need to be corrected. Note that if Processor-initiated C1 is enabled, then the SERDES can not be doing anything else.

#### SERDES Resources

The serdes during writing runs off of the crystal clock. When reading, the SERDES will run off the RDCLK from the data separator 118. When switching from reading to writing, the SERDES switches the clock over glitchlessly. The SERDES sequencer in sequencer 406, once it has detected a "sync" mark, will expect all following sync marks to be detected within a window which is plus and minus 1 bit time from the ideal time for detection. The SERGES sequencer will resync itself to the new sync mark when it is detected. If the sync is not detected within the windows, SERDES sequencer will "generate its own sync mark" and read in a block and try to detect the next sync mark.

#### SEQUENCES

The sequences of actions performed by formatter 110 are described following. These sequences include writing and reading a track by a read/write sequencer, DMA transfer, writing a frame, RAW, reading a frame, searching subdata blocks, appending, diagnostic read/write, read/write raw code, handling missing sync marks, and a phase lock loop (PLL) algorithm, all by a software sequence. The sequences are defined as follows.

#### Read/Write Sequencer 453: Writing a track

Read/Write Sequencer 453 does everything on a track basis, and runs off of a byte clock. Note that once something is "passed through" the 8-10 encoder logic, it is also serialized and shifted out to write channel 133. A counter is cleared by the synchronized changing edge 15 of the TRKCLK signal. It counts up based on a byte count generated from a crystal write clock.

The Sequence is as follows:

(1) Clear subdata block counter, and main data block counter, (2) Generate MARGIN 1 pattern: Disable and bypass the 8-10 encoder logic, (3) Generate a constant 1 pattern for 11 blocks. (11\*360 bits=3930 bits), (4) Generate PRE-AMBLE 1 pattern. Disable and bypass the 8-10 encoder logic, (5) Generate a constant 1 pattern for 2 blocks. (2\*360 bits=720 bits), (6) Generate Subdata area: Enable data to pass through the 8-10 encoder logic.

At point AAA, (1) Tell the encoder to generate the Sync symbol, (2) select the Subdata ID start address, (3) read the SW1 field from SRAM 117 and send it through 30 the 8-10 and latch it for parity generation, (4) increment the ID byte pointer, (5) read the SW2 field from SRAM 117 and send it through the 8-10 encoder logic and Xor it with the SW1 value, (6) latch the XOR result, (7) increment the ID byte pointer, (8) send the XOR result (parity) through the 8-10 encoder logic, (9) enable the C1 ECC generation and initialize the registers to zero, (10) select the subdata block start address. Then, in sequence, (1) read the subdata data pointed to by the block byte pointer and send it through the 8-10 encoder 40 and, if C1 on the fly generation is enabled, to the C1 generator circuits, (2) increment the block byte pointer, and repeat for 31 more bytes (32 total).

Next, increment the block counter and, in sequence, (1) generate the Sync symbol, (2) select the ID byte 45 pointer, (3) read the SW1 field from SRAM 117 and send it through the 8-10 encoder logic and latch it for parity generation, (4) increment the ID byte pointer, (5) read the SW2 field from SRAM 117 and send it through 50 the 8-10 encode logic and Xor it with the SW1 value, (6) send the XOR result, (7) increment the ID byte pointer, (8) send the XOR result (parity) through the 8-10 encoder logic.

Next, in sequence, (1) Read the subdata data pointed to by the block byte pointer and send it through the 8-10 encoder logic and, if C1 on the fly generation is enabled, to the C1 generator circuits, (2) increment the block byte pointer, (3) repeat for 23 more bytes, (24 total).

Next, in sequence, if C1 on the fly generation is enabled, (1) read the C1 generated parity bytes and send them through the 8-10 encoder logic, read them in the following sequence: Even X3, odd X3, even X2, odd X2, even X1, odd X1, even X0, odd X0, (2) if C1 on the fly generation is disabled, read the subdata data pointed to by the block byte pointer and send it through the 8-10 encoder logic, (3) increment the block byte pointer, and repeat for 7 more bytes (8 total), then (4) increment the block counter, and (5) clear the block byte pointer.

Repeat from point AAA three more times, (four passes total; two blocks per pass making eight blocks total).

Next, in sequence, (1) generate POST-AMBLE 1 pattern, by disabling and bypassing the 8-10 encoder logic to generate a constant 1 pattern for 1 blocks. (1\*360 bits=360 bits), (2) generate Inter Block Gap (IBG) 1 pattern by disabling and bypassing the 8-10 encoder logic, and generating a 100100 pattern for 3 blocks. (3\*360 bits=1080 bits), (3) generate ATF 1 pattern by generating the pattern for whatever azimuth track is selected and based on the odd frame bit (even addresses have a different pattern from odd addresses), by disabling and bypassing the 8-10 encoder logic, (4) generate the varying pattern for the 5 blocks of the ATF area, (5\*360 bits=1800 bits), assert the NPILOT pin when writing the pilot blocks.) (5) generate Inter Block Gap (IBG) 2 pattern by disabling and bypassing the 8-10 encoder logic and generating a 100100 pattern for 3 blocks (3\*360 bits=1080 bits), (6) generate PRE-AMBLE 2 pattern by disabling and bypassing the 8-10 encoder logic to generate a constant 1 pattern for 2 blocks, (2\*360 bits=720 bits), (7) generate MAIN data area by enabling data to pass through the 8-10 encoder logic, (8) select the main data block start address for the positive azimuth track.

At point BBB, select the main data ID address.

At point CCC, perform the following sequence:

(1) generate the Sync symbol, (2) read the W1 field from SRAM 117 and send it through the 8-10 encoder logic and Xor it with the block counter and main data block identifier bit for parity generation, (3) latch the XOR result, (4) increment the ID byte pointer (5) pass the block counter and main data block identifier bit (always zero) through the 8-10 encoder logic, (7) enable the C1 ECC generation and initialize the registers to zero.

Next, in sequence, (1) read the main data pointed to by the block byte pointer and send it through the 8-10 encoder logic and, if C1 on the fly generation is enabled, to the C1 generator circuits, (2) increment the block address pointer, (3) repeat for 31 more bytes, (32 total).

Next, increment the block counter.

Then, in sequence, (1) generate the Sync symbol (2) read the W1 field pointed to by the ID byte pointer from SRAM 117 and send it through the 8-10 encoder logic and Xor it with the block counter and main data block identifier bit for parity generation, (3) latch the XOR result, (4) increment the ID byte pointer, (5) generate the W2 field from the block counter and main data block identifier (always zero) and pass it through the 8-10 encoder logic, (6) send the XOR result (parity) through the 8-10 encoder logic.

Next, in sequence, (1) read the main data pointed to by the block byte pointer and send it through the 8-10 encoder logic and, if C1 on the fly generation is enabled, to the C1 generator circuits and (2) increment the block byte pointer. Repeat for 23 more bytes (24 total).

If C1 on the fly generation is enabled, (1) read the C1 generated parity bytes and send them through the 8-10 encoder logic, read them in the following sequence: Even X3, odd X3, even X2, odd X2, even X1, odd X1, even X0, odd X0.

If C1 on the fly generation is disabled, read the main data pointed to by the block byte pointer and send it through the 8-10 encoder logic and increment the block byte pointer, and repeat for 7 more bytes (8 total).

5,359,468

31

If C1 on the fly generation is enabled, clear the block byte pointer eight times to skip over the C1 parity areas. Then increment the block counter and clear the block byte pointer. Repeat from point CCC three more times (four total passes; two blocks per pass makes eight blocks; which completes the W1 ID cycle). Repeat from point BBB fifteen more times. (sixteen total passes;  $16 \times 8 = 128$  blocks in a track).

Next, in sequence, generate Inter Block Gap (IBG) 3 pattern by disabling and bypassing the 8-10 encoder logic. Generate a 100100 pattern for 3 blocks. ( $3 \times 360$  bits = 1080 bits).

Then generate ATF 2 pattern by generating the pattern for whatever azimuth track is selected and based on the odd frame bit. Disable and bypass the 8-10 encoder logic and generate the varying pattern for the 5 blocks of the ATF area ( $5 \times 360$  bits making 1800 bits) (Assert the Pilot Output signal when writing the pilot blocks.)

Then generate Inter Block Gap (IBG) 4 pattern by disabling and bypassing the 8-10 encoder logic and generating a 100100 pattern for 3 blocks, ( $3 \times 360$  bits = 1080 bits).

Then generate PRE-AMBLE 3 pattern by disabling and bypassing the 8-10 encoder logic, and generating a constant 1 pattern for 2 blocks ( $2 \times 360$  bits = 720 bits).

Then generate second subdata area by enabling data to pass through the 8-10 encoder logic.

At point DDD, (1) generate the Sync symbol, (2) select the Subdata ID start address, (3) read the SW1 field from SRAM 117 and send it through the 8-10 encoder logic and latch it for parity generation, (4) increment the ID byte pointer, (5) read the SW2 field from SRAM 117 and send it through the 8-10 encoder logic and Xor it with the SW1 value, (6) latch the XOR result, (7) increment the ID byte pointer, (8) send the XOR result (parity) through the 8-10 encoder logic, (9) enable the C1 ECC generation and initialize the registers to zero, and select the subdata block start address.

Then, in sequence, read the subdata data pointed to by the block byte pointer and send it through the 8-10 encoder logic and, if C1 on the fly generation is enabled, to the C1 generator circuits and increment the block byte pointer. Repeat for 31 more bytes (32 total).

Next, increment the block counter. Then (1) generate the Sync symbol, (2) select the ID byte pointer, (3) read the SW1 field from SRAM 117 and send it through the 8-10 encoder logic and latch it for parity generation, (4) increment the ID byte pointer, (5) read the SW2 field from SRAM 117 and send it through the 8-10 encoder logic and Xor it with the SW1 value, (6) latch the XOR result, (7) increment the ID byte pointer, (8) send the XOR result (parity) through the 8-10 ROM.

Then, in sequence, read the subdata data pointed to by the block byte pointer and send it through the 8-10 encoder logic and, if C1 on the fly generation is enabled, to the C1 generator circuits. Increment the block address pointer. Repeat for 23 more bytes (24 total).

If C1 on the fly generation is enabled, read the C1 generated parity bytes and send them through the 8-10 encoder logic. Read them in the following sequence: Even X3, odd X3, even X2, odd X2, even X1, odd X1, even X0, odd X0. If C1 on the fly generation is disabled, read the subdata data pointed to by the block address pointer and send it through the 8-10 encoder logic. Increment the block byte pointer. Repeat for 7 more bytes (8 total).

Next, increment the block counter and clear the block byte pointer. Repeat from point DDD three more

32

times (four passes total; two blocks per pass making 8 blocks total).

Next, generate POST-AMBLE 2 pattern by disabling and bypassing the 8-10 encoder logic and generating a constant 1 pattern for 1 block ( $1 \times 360$  bits = 360 bits).

Then, generate MARGIN 2 pattern by disabling and bypassing the 8-10 encoder logic and generating a constant 1 pattern for 11 blocks ( $11 \times 360$  bits = 3930 bits).

Insert the Read-After-Write sequence at this point. This sequence reads what was written on the opposite azimuth track of the previous frame. When finished, microcontroller 105 must decide whether rewriting that frame is necessary.

#### Read/Write Sequencer 453: Reading a track

A counter is cleared by the synchronized changing edge of the TRKCLK signal. It counts up based on a byte count generated from the crystal write clock and the read data clock from data separator 118. If at any time while reading sequencer 453 does not detect a sync mark within its window and must generate its own sync mark, the Sync Missed bit in Status Register 416 must be set. The MARGIN 1 and PRE-AMBLE pattern comes in here. They are the same constant 1s pattern. The pattern continues for at most 13 blocks. ( $13 \times 360$  bits = 4680 bits). Acquire PLL lock to the data (see section on PLL Algorithm for details below). Read the subdata area: Clear a flag indicating "stop writing subdata blocks to SRAM 117", and enable the 10-8 decoder logic.

At point AAA, select the address for the "Read Subdata for the selected azimuth track", and then (1) detect the sync byte, (2) clear the "byte/symbol" counter to synchronize to the data, (3) enable the C1 syndrome generation circuits and initialize to zero, (4) clear the ID byte pointer and (5) clear the block byte pointer.

Next, (1) fetch the decoded SW1 byte from the 10-8 decoder logic. For the first pass only, write the byte to the SRAM 117 location pointed to by the ID address pointer, and increment the ID byte pointer; (2) fetch the decoded SW2 byte from the 10-8 decoder logic (for the first pass only, write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID pointer), (3) fetch the decoded parity byte from the 10-8 decoder logic (for the first pass only, write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer).

Next, fetch the decoded byte from the 10-8 decoder logic. If the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the block byte pointer and send the byte to the C1 syndrome generation circuits, otherwise do nothing. Then increment the block byte pointer. Repeat this for 31 more bytes.

Next, detect the sync byte and clear the "byte/symbol" counter to synchronize to the data.

Then, (1) fetch the decoded SW1 byte from the 10-8 decoder logic (for the first pass only, write the byte to the SRAM 117 location pointed to by the ID address pointer, and increment the ID pointer), (2) fetch the decoded SW2 byte from the 10-8 ROM (for the first pass only, write the byte to the SRAM 117 location pointed to by the ID address pointer and increment the ID pointer), (3) fetch the decoded parity byte from the 10-8 decoder logic (for the first pass only, write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer).

Then, fetch the decoded byte from the 10-8 decoder logic. If the "stop writing subdata blocks to SRAM

5,359,468

33

117" flag is clear, then write the byte to the SRAM 117 location pointed to by the block address pointer and send the byte to the C1 syndrome generation circuits, and increment the block byte pointer. Repeat for 23 more bytes.

Next, fetch the decoded byte from the 10-8 decoder logic. If the "stop writing subdata blocks to SRAM 117" flag is clear, then send the byte to the C1 syndrome generation circuits. If the "Enable on-the-fly C1 Generation" bit in the ECC control register is cleared and if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the block byte pointer and increment the block byte pointer. Repeat for 7 more bytes. If the eight syndrome bytes are all zero, set a flag indicating stop writing subdata block data to SRAM 117.

If the "Enable on-the-fly C1 generation" bit in the ECC control register is cleared, skip this paragraph. If the "stop writing subdata blocks to SRAM 117" flag is clear, then write the syndrome bytes to the SRAM 117 location pointed to by the block byte pointer, incrementing the block byte pointer after each cycle. Write them in the following sequence: Even X3, even X2, even X1, even X0, odd X3, odd X2, odd X1, odd X0.

Repeat from point AAA for 3 more times (2 blocks per pass; 8 blocks total processed).

The POST-AMBLE pattern enters here as a constant 1 pattern for 1 block (360 bits), with the PLL locked to the write clock (crystal).

Next is the ATF timings. The lead sequencer is not involved in this other than to count past it.

Next the PRE-AMBLE 2 pattern comes in as a constant 1s pattern. Continuing for at most 2 blocks with PLL lock to the data. See the section on PLL Algorithm for details below.

Next is reading the main data area comprising the sequence following:

select the main data block start address for whatever azimuth track is selected and clear the "stop writing main data ID to SRAM 117" flag. Then select 40 the "Read Main data IDs for the selected azimuth track" start address.

At point DDD after eight blocks are read in, set the "stop writing main data ID to SRAM" flag. Next, detect the sync byte and clear the "byte/symbol" counter to synchronize to the data. Enable the C1 syndrome generation circuits and initialize to zero. Then (1) fetch the decoded W1 byte from the 10-8 decoder logic (if the "stop writing main data ID to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), (2) latch the byte for parity checking, (3) fetch the decoded W2 byte from the 10-8 decoder logic (if the "stop writing main data ID to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), (4) latch the byte, (5) Xor the byte with the previous byte and latch the result, (6) fetch the decoded parity byte from the 10-8 decoder logic (if the "stop writing main data ID to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), (7) Xor the byte with the previous XOR result. On the first pass through this, if the XOR result is zero, check the block number in W2 and compare it to the expected block number (0). If the read block number is not 0, then set the Missing First Block Detected bit in the Status Register. If the read block number is odd, then incre-

34

ment the block count and go to point DDD1. Next, fetch the decoded byte from the 10-8 decoder logic and write the byte to the SRAM 117 location pointed to by the block byte pointer. Send the byte to the C1 syndrome generation circuits. Increment the block byte pointer and repeat for 31 more bytes.

Then detect the sync byte and clear the "byte/symbol" counter to synchronize to the data. Then (1) fetch the decoded W1 byte from the 10-8 decoder logic (if the "stop writing main data ID to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), (2) latch the byte for parity checking, (3) fetch the decoded W2 byte from the 10-8 ROM (if the "stop writing main data ID to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), (4) latch the byte, (5) Xor the byte with the previous byte and latch the result, (6) fetch the decoded parity byte from the 10-8 decoder logic (if the "stop writing main data ID to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), (7) Xor the byte with the previous XOR result.

At point DDD1, fetch the decoded byte from the 10-8 decoder logic, write the byte to the SRAM 117 location pointed to by the block byte pointer, send the byte to the C1 syndrome generation circuits, and increment the block byte pointer. Repeat for 23 more bytes. Then fetch the decoded byte from the 10-8 decoder logic and send the byte to the C1 syndrome generation circuits. If the "Enable on-the-fly C1 Generation" bit in the ECC control register is cleared, write the byte to the SRAM 117 location pointed to by the block address pointer and increment the block byte pointer. Repeat for 7 more bytes. If the eight syndrome bytes are not all zero, set the NonZero C1 Syndrome detected bit in the Status Register. If the "Enable on-the-fly C1 Generation" bit in the ECC control register is cleared, skip the following sequence. Otherwise, write the syndrome bytes to the SRAM 117 location pointed to by the generated address, incrementing the block byte pointer after each cycle. Write them in the following sequence: Even X3, even X2, even X1, even X0, odd X3, odd X2, odd X1, odd X0.

Repeat from point DDD for 63 more times (64 total passes; 2 blocks per pass; 128 blocks total processed).

ATF timings enter at this point. The read sequencer does not get involved in this other than to count past it. 50 Lock the PLL to the write clock (crystal).

The PRE-AMBLE pattern enters here as a constant 1s pattern which continues for at most 2 blocks. Acquire PLL lock to the data (see section on PLL Algorithm for details below)

55 Next operation is to read the Subdata Area 2.

At point EEE, (1) select the "Read Subdata for the selected azimuth tracks" start address, (2) detect the sync byte, (3) clear the "byte/symbol" counter to synchronize to the data, (4) enable the C1 syndrome generation circuits and initialize to zero, (5) select the "Read Subdata IDs for the selected azimuth track" start address, (6) select the "Read Subdata IDs for the selected azimuth track" start address. Then fetch the decoded SW1 byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), fetch the decoded SW2 byte from the 10-8 decoder

logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), fetch the decoded parity byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer).

Next, fetch the decoded byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the block byte pointer and send the byte to the C1 syndrome generation circuits), increment the block byte pointer, and repeat for 31 more times.

Next, detect the sync byte and clear the "byte/symbol" counter to synchronize to the data. Then, fetch the decoded SW1 byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), fetch the decoded SW2 byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), fetch the decoded parity byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer), fetch the decoded byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the ID byte pointer, and increment the ID byte pointer).

Then, fetch the decoded byte from the 10-8 decoder logic (if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the block byte pointer and send the byte to the C1 syndrome generation circuits), increment the block byte pointer and repeat for 23 more bytes.

Then fetch the decoded byte from the 10-8 decoder logic, send the byte to the C1 syndrome generation circuits (if the "Enable on-the-fly C1 Generation" bit in the ECC control register is cleared and if the "stop writing subdata blocks to SRAM 117" flag is clear, then write the byte to the SRAM 117 location pointed to by the block byte pointer and increment the block byte pointer), repeat for 7 more bytes, if the eight syndrome bytes are not all zero. If the "Enable on-the-fly C1 Generation" bit in the ECC control register is cleared, do not perform the steps in this paragraph. If the "stop writing subdata blocks to SRAM 117" flag is clear, then write the syndrome bytes to the SRAM 117 location pointed to by the block byte pointer, incrementing the block byte pointer after each cycle. Write them in the following sequence: Even X3, even X2, even X1, even X0, odd X3, odd X2, odd X1, odd X0.

Repeat from point EEE three more times (2 blocks per pass; 8 blocks total precessed).

At this point, for the whole frame (both tracks), C1 ECC correction by microcontroller 105 must begin and C2 syndrome generation and correction must begin. Single byte C1 ECC errors should be corrected before C2 syndrome generation is started. The POST-AMBLE and MARGIN pattern comes in here using constant 1s pattern.

#### Software Sequence: DMA Transfer

When writing data to the tape, the data must first be read from the DRAM buffer manager's main buffer 104 and to one of the 8K byte from buffers in the SRAM 117. Data is always handled one frame at a time. When

reading data from the tape, a frame of data is read into one of the frame buffers, ECC correction is applied to the data, and then the data is transferred to main data buffer 104.

5 Note that formatter 110 has no byte count so it does not automatically know when the transfer is complete. It will request transfers until disabled so the DRAM buffer manager 103 should only be programmed to transfer the exact number of data required in a frame. The typical sequence of commands for transferring the frame of data from main data buffer 104 to frame buffer 117 follows:

1. Set up the count in buffer manager 103 to transfer the 5756 bytes of user data. Note that the header is not included in this. Set up the address pointer in buffer manager 103 to point to the desired frame. Set up buffer manager 103 to send data. Enable buffer manager 103 to generate an interrupt when the transfer has finished.

15 2. Allow DRAM buffer manager 103 to respond to NFDREQ requests.

3. Write E4h to the DMA Control Register (0106h)

1xxx xxxx: enable interleaving

x1xx xxxx: enable randomizing

xx1x xxxx: slow down DMA requests

xxx0 xxxx: select DDS interleave mode, no header transferred

xxxx 0xxx: disable DMA request generation

xxxx x1xx: read from the DMA channel and write to SRAM 117

xxxx xxYY: YY=frame buffer to write to: 00 selects Frame Buffer 0

4. Write any value to the Initialize DMA Register at address 0103h to initialize the interleave counters, track checksum logic, and randomizing logic.

5. Write ECh to DMA Control Register 438 (0106h) to enable DMA xxxx 1xxx: enable DMA request generation

6. The data transfers now. This takes 4.3 milliseconds (103,610 clocks). Wait for an interrupt from DRAM buffer manager 103.

7. Read Interrupt Register (0113h). If DRAM buffer manager 103 request bit is set, then the transfer has completed and go to step 8. Otherwise, service the other interrupt source and go back to step 6.

8. Write E4h to DMA Control Register 438 (0106h) to disable DMA request generation.

9. Service DRAM buffer manager 103 to stop the interrupt request.

The typical sequence of commands for transferring the frame of data from frame buffer 117 to the main data buffer follows:

1. Set up the count in buffer manager 103 to transfer the 5756 bytes of user data. Note that the header is not included in this. Set up the address pointer in DRAM buffer manager 103 to point to the desired frame. Set up the buffer manager to receive data. Enable DRAM buffer manager 103 to generate an interrupt when the transfer has finished.

2. Allow DRAM buffer manager 103 to respond to NFDREQ requests.

3. Write E0h to the DMA Control Register 438 (0106h).

1xxx xxxx: enable interleaving

x1xx xxxx: enable randomizing

xx1x xxxx: slow down DMA requests

xxx0 xxxx: select DDS interleave mode, no header transferred

xxxx 0xxx: disable DMA request generation

## 5,359,468

37

- xxxx x0xx: read from the DMA channel and write to SRAM 117
- xxxx xxYY: YY=frame buffer to write to: 00 selects Frame Buffer 0
- 4. Write any value to the Initialize DMA Register at address 0103h to initialize the interleave counters, track checksum logic, and randomizing logic.
- 5. Write E8h to the DMA Control Register 438 (0106h). xxxx 1xxx: enable DMA request generation.
- 6. The data transfers now. This takes 4.3 milliseconds (103,610 clocks). Wait for an interrupt.
- 7. Read Interrupt Register (0113h). If DRAM buffer manager 103 request bit is set, then the transfer has completed and go to step 8. Otherwise, service the other interrupt source and go back to step 6.
- 8. Write E0h to the DMA Control Register 438 (0106h) to disable DMA request generation.
- 9. Service DRAM buffer manager 103 to stop the interrupt request.

### Software Sequence: Writing a Frame

This is the write sequence for a typical streaming write of frames. Special interrupts and conditions are not accounted for. The steps required to write a frame can be divided into two groups; those needed to process a frame, and those needed to send the frame through the serdes. When streaming writing, one frame will be going through the serdes while another is being processed.

The steps to process a frame are envisioned to start right after the Write+head starts its sweep of the tape and should complete before the Write+head comes around again. This way the processing time is a consistent 1 head revolution time (i.e. 23.52 milliseconds).

### STEPS TO SEND A FRAME THROUGH THE SERDES

- 1. Bit 6 of the ECC Control Register 444 (010Bh) should already be set to enable C1 parity generation on the fly.
- 2. Wait for the Read+head to contact the tape. Allow time for the head to get into the subdata area.
- 3. Write 00h to Serdes Control Register 1 458 (010Dh) to select normal mode.
- 4. Write 40h to Serdes Control Register 0 457 (010Ch) to enable writing and use Frame Buffer 0 and Subdata Buffer 0 and track A and Even Frame Number.
- 5. Wait for the Write+head to contact the tape and allow time for writing of subcodes to start.
- 6. Set up for Read--head next if needed.
- 7. Wait for the Read--head to contact the tape and allow time for the head to get into the subdata area.
- 8. Write 44h to Serdes Control Register 0 457 (010Ch) to enable writing and use Frame Buffer 0 and Subdata Buffer 0 and track B and Even Frame Number.
- 9. Wait for the Write--head to contact the tape and allow time for writing of subcodes to start.
- 10. Set up for Read+head next if needed.

### STEPS TO PROCESS A FRAME

- 1. Set up the DMA and transfer the data over into From Buffer 0 as described below in the DMA Sequence.
- 2. Put the four header bytes in SRAM 117. DFID and Reserved byte at address "Buffer Start"+2h
- LF-ID byte at address "Buffer Start"+0h
- DFID and Reserved byte at address "Buffer Start"+1002h.
- LF-ID byte at address "Buffer Start"+1000h

38

- 3. Write 50h to the ECC Control Register 444 (010Bh) to select C2 generation for Track A and allow C1 generation on the fly.
- 4. Write 70h to the ECC Control Register 444 (010Bh) to enable C2 generation for Track A and allow C1 generation on the fly.
- 5. Write any value to the Start ECC Register (0102h) to start the C2 generation.
- 6. Wait for 0.55 milliseconds (12720 clocks) or for the C2 Status bit in the ECC Control Register to be set. This allows the operation to complete.
- 7. Write 72h to the ECC Control Register 444 (010Bh) to enable C2 generation for Track B and allow C1 generation on the fly.
- 8. Write any value to the Start ECC Register (0102h) to start the C2 generation.
- 9. Wait for 0.55 milliseconds (12720 clocks) or for the C2 Status bit in the ECC Control Register to be set. This allows the operation to complete.
- 10. Write 50h to the ECC Control Register 444 (010Bh) to disable C2 generation and allow C1 generation on the fly. Leave select of C2 generation.
- 11. Xor the four header bytes into the generated track checksums.
- 25 Read the Track 1 Checksum Low Byte (0103h) and XOR with the DFID and Reserved byte. The result goes in the PC6 byte of Pack Item 3.
- Read the Track 1 Checksum High Byte (0102h) and XOR with the LF-ID byte. The result goes in the PC5 byte of Pack Item 3.
- 30 Read the Track 2 Checksum Low Byte (0105h) and XOR with the DFID and Reserved byte. The result goes in the PC6 byte of Pack Item 4.
- Read the Track 2 Checksum High Byte (0104h) and XOR with the LF-ID byte. The result goes in the PC5 byte of Pack Item 4.
- 35 12. Setup Subdata blocks in Subdata Buffer 0. Update the Subdata Pack items and generate new parity. Pack items 3 and 4 need the new Absolute Frame Count. Pack items 1 & 2 need to be updated when starting a new group.
- Software Sequence: Read After Write
- The hardware function for Read-After-Write is exactly the same as for Normal Read except that no main data is sent to SRAM 117, and, pointers to Bad C1 Block pairs are saved along with the actual syndrome bytes. Refer to the Bad C1 Block Counter Register 447 description above for details.
- 1. Set bit 6 of the ECC Control Register 444 (010Bh) to enable C1 syndrome generation on the fly. This should already be set up from the write.
- 2. Wait for the Write--head to contact the tape and wait until after it is in the first subdata area.
- 3. Write any value to the Read Track Checksum Register High Byte (0108h) to clear the Read Track Checksum.
- 4. Write any value to the Bad C1 Block Control Register 447 (0107h) to clear it.
- 5. Write 00h to Serdes Control Register 1 458 (010Dh) to select normal mode.
- 6. Write C2h to Serdes Control Register 0 457 (010Ch) to enable RAW on track A of buffer 2.
- 7. Wait sufficient time for the Read+head to complete its sweep of the tape.
- 8. Read Status Register 459 (010Eh) and test the Missing First Block Detected, bad Track Detected, and Non-Zero C1 Syndromes Detected bits.
- 9. Do checking of subdata.

10. Check the recalculated track checksum. Read the Read Track Checksum Register Low Byte (0109h) and the Read Track Checksum Register High Byte (0108h) and compare to the expected value. If there are any data errors, it will effect the checksums.

11. Read the Bad C1 Block Counter Register 447 (0107h). Save if needed for later. Clear if needed by writing any value to the Bad C1 Block Counter Register 447 (0107h). If it is 00h and the NonZero C1 Syndrome detected bit in Status Register 459 is clear, then no C1 ECC errors were detected. Otherwise, the syndromes must be checked to determine if single byte or multiple byte errors occurred.

12. Based on the above information, decide whether to rewrite the frame. If the decision is to not rewrite, then repeat steps 2-12 for the second track of the frame (write C6h to the Serdes Control Register while the Write+head is on the tape and after it has passed into the first subdata area).

13. If the decision is to rewrite, then rewriting with no frames wasted can be achieved if there is sufficient time to do the following before entering into the subdata area of the Write+head sweep of the next frame. If insufficient time, then complete the writing of the next frame and then rewrite the frame just checked.

The tasks to be done are:

1. Update the subdata pack items (absolute frame counts) and parities. This could be anytime after the frame is completely written out. It could be done while the first track is being read in.

2. Repoint the buffer in the Serdes Control Register 0 457.

#### Software Sequence: Reading a Frame

This is the read sequence for a typical read frame. It is assumed that the heads are tracking the frames on the tape. Special interrupts or exceptions are not accounted for. The steps required to read a frame can be divided into two groups: those needed to process a frame, and those needed to read the frame from the serdes. When streaming reading, one frame will be going through the serdes while another is being processed. The steps to process a frame are envisioned to take start right after the Read-head finishes its sweep of the tape and should complete before the next Read-head comes around. In this way the processing time is a consistent 1 head revolution time (i.e. 23.52 milliseconds).

#### STEPS TO READ A FRAME FROM THE SERDES:

1. Set bit 6 of the ECC Control Register 444 (010Bh) to enable C1 syndrome generation on the fly.

2. Wait for the non used+head to contact the tape and wait until after it has passed into the first subdata area.

3. Write any value to the Read Track Checksum Register High Byte (0108h) to clear the Read Track Checksum Register 460.

4. Write any value to the Bad C1 block Counter Register 447 (0107h) to clear it.

5. Write 00h to Serdes Control Register 1 457 (010Ch) to enable Read on track A into buffer 0.

7. While the Read+head is on the tape and after it is in the first subdata area, write 00h to Serdes Control Register 0 457 (010Ch) to disable any activity on the next head sweep.

8. Wait a set amount of time for the track to be completely read while processing the previous frame.

9. Read the Read Track Checksum Low Byte (0109h) and the Read Track Checksum Low Byte (0108h) and save for checking later.

10. At some point while the non used-head is on the tape and after it is in the first subdata area, write 84h to Serdes Control Register 0 457 (010Ch) to enable Read on track B into buffer 0.

11. Write any value to the Read Track Checksum Register High Byte (0108h) to clear Read Track Checksum Register 460.

12. While the Read-head is on the tape and after it is in the first subdata area, write 00h to Serdes Control Register 0 457 (010Ch) to disable any activity on the next head sweep.

15. Wait a predetermined amount of time for the track to be completely read. While continuing to process the previous frame.

14. Read the Read Track Checksum Low Byte (0109h) and the Read Track Checksum Low Byte (0108h) and save for checking later.

#### STEPS TO PROCESS A FRAME

1. Look at the syndromes on the Subdata blocks to see if the data is reliable. If syndromes are bad, C1 correction could be done.

2. Check any Subdata Pack Item values that are necessary.

3. Do C1 ECC Correction on main data. Look at the C1 Bad Block Count stored in SRAM 117 to find out how many pairs of blocks had at least one set of non-zero C1 syndromes. Pointers to these blocks are in SRAM 117. Microcontroller 105 has to actually look at the 8 syndromes generated for each pair of blocks to determine which "block" had the error or if both "blocks" had errors. Single byte errors in a block are corrected. Blocks with multiple byte errors are called erasures and stored in a table for use when doing C2 ECC correction.

4. Select C2 syndrome generation over Track A of buffer 0 by writing 58h to ECC Control Register 444 (010Bh).

5. Enable the C2 syndrome generation over Track A of buffer 0 by writing 78h to ECC Control Register 444 (010Bh).

6. Write any value to the Start ECC Register (0102h) to start the C2 generation.

7. Poll for completion by reading Interrupt Register 415 (0113h) waiting for the C2 Status bit to be set.

8. If the C2 Status bit is set and the Non-Zero ECC Syndromes bit is set, then go to step 9. Otherwise, if the ECC Interrupt is set and the non-zero ECC Syndromes bit is clear, then go to step 11.

9. Clear the C2 Status bit by reading C2 Block Address 446 (010Ah). It points to the starting byte address of the block which had the non-zero syndromes. Look at the syndromes.

10. Calculate the correction using the syndromes and any erasures from C1. XOR the correction into the appropriate bytes of the block. Restart by going to step 6.

60 11. Clear the C2 Status bit by reading the ECC Block Address (010Ah).

12. Enable the C2 syndrome generation over Track B of buffer 0 by writing 7Ah to ECC Control Register 444 (010Bh).

13. Write any value to the Start ECC Register (0102h) to start the C2 generation.

14. Poll for completion by reading Interrupt Register 413 (0113h) waiting for the C2 Status bit to be set.

15. If the C2 Status bit is set and the Non-Zero ECC Syndromes bit is set, then to step 16. Otherwise, if the ECC Interrupt is set and the Non-Zero ECC Syndromes bit is clear, then go to step 18.

16. Clear the C2 Status bit by reading the ECC Block Address (010Ah). It points to the starting byte address of the block which had the non-zero syndromes. Look at the syndromes.

17. Calculate the correction using the syndromes and any erasures from C1. XOR the correction into the appropriate bytes of the block. Restart by going to step 13.

18. Clear the C2 Status bit by reading the ECC Block Address (010Ah).

19. Do any checking of the four header bytes.

20. Set up the DMA and transfer the data over from Frame Buffer 0 as described below in the DMA Sequence.

21. Recheck track checksums if needed.

Note that the Read Track Checksums are also available for additional checking. However, to be useful, all ECC corrections which do not cover the C2 parity blocks must be applied to them.

#### Search

During SEARCH, subdata blocks are attempted to be read. Microcontroller 105 will generate windows during which blocks will be searched. When the FMTT1 signal is high, the window is considered open. When low, the window is closed. These windows will cover only the subdata areas so distinguishing subdata blocks from main data blocks is not a problem. Whenever the window is open, blocks will be searched for. Nothing prevents microcontroller 105 from opening the windows while all four heads sweep the tape, thereby obtaining the maximum block sample rate possible. Sync detect will determine the start of a block. Once a block starts transferring to SRAM 117, the whole 32 bytes of the block will be transferred independent of the state of the FMTT1 signal.

All blocks will be written to the selected track portion of frame buffer 117 pointed to by the Buffer Selection and Track A/B bits in the Serdes Control Register 0 457 starting at offset 0 in the buffer. Data will be transferred into the buffer as long as the window defined by the FMTT1 signal is active. If a block transfer is in progress when the FMTT1 signal goes inactive, the block will finish transferring. The Block Counter Register (0111h) will point to the next block in SRAM 117 to fill. In other words, it will be equal to the number of blocks read. The Block Counter Register will roll over to 0 after 128 blocks are read and will increment up from there (overwriting previously read blocks) provided microcontroller 105 does not reset it. Microcontroller 105 will be able to reset the Block Counter when in search mode by writing any value to it. No C1 syndrome generation will be attempted on the blocks read in. The C1 parity bytes will be written to memory as they were read from tape.

#### Appending

During appending, the system is in search mode at 1X speed with ATF working and frames are being read to keep track of location and to be able to switch over to writing frames. Two frames of data should be in SRAM 117 ready to write out. The third frame buffer should be used for the subdata blocks read while in search mode. Microcontroller 105 must know what frame it is looking for, or sequence of frames (if frames have been rewritten) in order to know when to switch off search mode

and go into write mode. Also, when switching to write mode, the ATF tracking must be turned off and we run open loop. Note that the absolute frame count must be contiguous (last data to save=n, next previously written frame=n+1, new frame =n+2); the new frame is an amble frame. According to the spec, a minimum of 2 amble frames will be written before starting to write data for a seamless append.

#### Software Sequence: Diagnostic Read

Diagnostic Read works just like regular read from tape except the raw code comes from the main buffer instead of from tape. When in diagnostic read mode, no DMA needs to be enabled in the formatter 110. DMA will be enabled automatically. The Read Clock from data separator 118 is not used, the write clock (crystal) is used if the Enable SERDES Clock bit of the SERDES Control Register 1 458 is clear. If the Enable SERDES Clock bit of the SERDES Control Register 1 458 is set, microcontroller 105 must clock each bit by writing any value to the Bit Pointer Register. Raw code is read from the main buffer, serialized, deserialized, decoded, and sent to SRAM 117. If C1 on the fly is enabled, then syndrome bytes will be calculated and written to SRAM 117 also.

It is assumed that a whole track of data will be read from main data buffer 104. No special control logic is added to sequence 406 in order to be able to send only part of a track. Sequence 406 does not look for the ATF areas, so those areas do not need to be sent. If the serdes is idle, the read will start when Serdes Control Register 0 457 is set up for Diagnostic Read and the synchronized changing edge of the TRKCLK signal is detected. IF the serdes is not idle, the read will start when Serdes Control Register 0 457 is written to set it up for Diagnostic Read (8820 bytes are in a raw track).

There are some special track status bits which are useful during diagnostic read. Sync Detected and Code Violation Detected in the Status register can be used along with the Stop on Code Violation and Stop on Sync Detect in Serdes Control Register 1 458 to help microcontroller 105 determine where structures or errors are in the raw code. By enabling the Stop on Sync Detect, the Sync Detected bit can be polled to determine when a sync was detected. Then microcontroller 105 can look at how many bytes were transferred by buffer manager 103 and the bit counter to determine exactly where the sync is located in the raw code in the main buffer.

#### Software Sequence: Diagnostic Write

Diagnostic Write works just like regular write to tape except the raw code goes to main data buffer 104 instead of to the tape. When in diagnostic write mode, no DMA needs to be enabled in formatter 110. DMA will be enabled automatically. A track of data is read from SRAM 117, encoded, ATF and subcode generated, serialized, deserialized into 8 bit bytes, and DMAed over to the main buffer. If C1 on the fly is enabled, then parity bytes will be generated also. A whole track of data is always sent to main data buffer 104. The write will start when the Serdes Control Register 0 457 is set up for Diagnostic Write and the synchronized changing edge of the TRKCLK signal is detected (8820 bytes are in a raw track).

#### Software Sequence: Read Raw Code

Read data from the tape and deserialize and DMA over to DRAM buffer manager 103. Set up is similar to appending. When the track before the track of interest is read, wait for the right head to come around and turn

on read raw code and DMA each track of data to DRAM buffer manager 103. DMA is turned on automatically when the mode is set to Read Raw Code. After microcontroller 105 has "fixed" the data, use diagnostic read to decode the data.

Note that at 12.0M bits/sec, this is 1.5M bytes/sec over DMA channel 132 to main data buffer 104. With +3% tape speed, the data rate is 1.54M bytes/sec.

#### Software Sequence: Write Raw Code

Set up the frame's worth of data to write in the main buffer. Set up the formatter 110 just as in appending. When the track before the track of interest is read, wait for the right head to come around and turn on write raw code and DMA each track of data from the buffer manager to tape. DMA is turned on automatically when the mode is set to Write Raw Code.

Note that at 12.0M bits/sec, this is 1.5M bytes/sec over DMA channel 132 to main data buffer 104. With +3% tape speed, the data rate is 1.545M bytes/sec.

#### Handling Missing Syncs

On a tape, it is likely that a defect will occur such that it affects a sync mark at the start of a block. Formatter 110 has several features which help in the detection of and recovery from such an event. Information is stored in several locations to aid the software in its recovery algorithm.

The ID bytes (W1, W2, Parity) from the first eight main data blocks read in from a track are stored in SRAM 117 for later analysis. By looking at the block number in W2, it can be determined whether any blocks were missed at the start of the main data area.

Status Register 416 has two bits which provide additional information. The "Started Main Data on Block 1" bit is set when the hardware detects what it thinks is the first sync mark of the main data area and the ID bytes indicate that it is block 1 (the ID parity check must be good to set to set this bit). When the hardware detects this specific condition, it will not store the data from the block at the start of the track buffer but will increment the address pointer to point to where block 1 should go in SRAM 117 and store it there. If all the other blocks come in as expected, then C2 ECC can be used to correct block 0 being missed.

The second Status Register bit is "Sync Missed". When a sync mark is detected, the hardware will read in that block and expect the next sync mark to occur at its ideal position  $\pm 1$  bit time. The hardware resyncs its bit counter when it detects the sync mark. If the sync is not detected within this window, then the "Sync Missed" bit is set, and the hardware pretends a sync occurred and transfers the next block to SRAM 117. The hardware will then detect the sync mark of the next block and get back in sync with the data.

#### PLL Algorithm

The circuit which controls RDGATE and FLOCK generation is reset when not in Normal Read or Read After Write mode, when the synchronized changing edge of the TRKCLK signal occurs, at the end of a data or subdata area, and by NRESET being low. The reset state is with RDGATE low and FLOCK low and the "ONES COUNTER" cleared.

In Search Mode, FLOCK and RDGATE normally are low. They go high when FMTT1 goes high (the search window is open). FLOCK and RDGATE remain high as long as FMTT1 is high. They also will remain high for up to one block after FMTT1 goes low. This allows the last block started to finish transferring to SRAM 117.

In Normal Read, Read After Write, after reset, the circuitry counts the number of falling edges on the RAWCOD signal which occur in a 2 "byte" period (20 clocks of the RDCLK signal). If 17 or more transitions occur in this period, then RDGATE goes high and four "byte" periods later FLOCK goes high. Once high, RDGATE and FLOCK remain high until the circuitry is reset. If, during the 2 "byte" period, fewer than 17 transitions occur, then the counter is cleared and counts for another 2 byte period and repeats the test.

#### TEST MODES

Formatter 110 provides test modes and features including ATE tri-state, ATE activity, and processor diagnostics hooks.

##### ATE Tri-state

Grounding the/TRIS pin causes formatter 110 to tri-state all of its pins. This facilitates in-circuit ATE test of other components on the board with formatter 110, since the ATE equipment does not need to overdrive any signals from formatter 110.

The/TRIS pin has a small internal pullup. It should be connected to a test point on the pcb for easy ATE access.

##### ATE Activity

Grounding the/ACTI pin, when the/TRIS pin is high, causes formatter 110 to connect all its output pins directly to input pins. This allows ATE equipment to easily verify that the correct part is on the board in the proper orientation, and that all the pins make contact with their pads. The ATE equipment does this by applying stimulus to the input pins (overdriving the normal input sources, if necessary) and looking for the appropriate changes on each of the output pins. The/ACTI pin has a small internal pullup. It should be connected to a test point on the pcb for easy ATE access. The pin mapping for the Activity test mode is shown in Appendix VI.

##### Processor diagnostics hooks

To facilitate diagnostics of formatter 110 by firmware running in microcontroller 105 processor, the following diagnostics hooks are provided:

1. All of the registers which are writable may also be read back. In most cases, microcontroller 105 should be able to read back the same data that was written.

2. Readback is provided for some buried registers (for which reading is not required for normal operation).

3. A loop-back port is provided for diagnosing the serial prot.

4. Diagnostic Read and Diagnostic Write together provide a loop-back capability for testing the SERDES.

##### Signal Descriptions

All bidirectional and input pins have pullup resistors.

##### Processor/System Signals

AD[0..7] BI-Multiplexed Address and Data bus from microcontroller 105.

A[8 .. 15] IN-High order Address lines from microcontroller 105.

INST IN-Signal from microcontroller 105 indicating code (instruction) space access.

LA[0..7] OUT-Low order address lines latched from the AD[0..7] lines

ALE IN-Address Latch Enable. This signal indicates when valid data is present on the multiplexed address and data bus.

NRD IN-Read signal from microcontroller 105.

NWR IN-Write signal from microcontroller 105.

RDY OUT-Ready signal to microcontroller 105 indicating that the active bus cycle can complete.

5,359,468

45

When inactive, microcontroller 105 adds wait states to the current bus cycle.

FMTT1 IN-A high speed output from microcontroller 105 which during Raw Code and Diagnostics is high when formatter 110 should transfer data 5 to buffer manager 103. During Search Mode is high when subdata block should be looked for.

TRKCLK IN-An active high, high speed output from microcontroller 105 which indicates when head switching should occur. The changing edge 10 of the signal indicates the start of a head sweep across the tape. The signal will have a nominal 50% duty cycle.

NRESET IN-The reset signal from the system.

FINT OUT-An active high interrupt signal to microcontroller 105. It can be activated by not only select formatter 110 conditions but also the external interrupt signals.

NROMOE OUT-An active low signal enabling ROM 107 to drive the multiplexed address and data 20 bus.

SL0 BI-Inputs for SCSI address selection. When selected as outputs by setting the TEST OUT bit in the SERDES Control Register 1 458; SLO supplies the SERDES RAM cycle signal for debug purposes, SL1, supplies the ECC RAM cycle signal for debut purposes, SL2 supplies the DMA RAM cycle signal for debug purposes.

SL1

SL2

SCLOCK IN-The clock signal which runs formatter 110. This input can be used with the XTAL OUT pin and a crystal to run formatter 110 if need be. For DDS Compliance, the clock should run at a frequency of 18.815855 MHz.

XOUT OUT-The output signal which drives a crystal. The signal at this output should be the inversion of the SCLOCK signal.

SDAT BI-Serial Port data bus.

NTRIS IN-Input telling all output pins to go into 40 high impedance mode.

NACTI IN-Input telling all inputs to connect to outputs.

Buffer Manager/SCSI Controller Signals

FD[0..7] BI-DMA channel data bus.

NFDREQ OUT-An active low signal requesting a DMA transfer.

NFDACK INPU-An active low signal indicating a transfer is in progress.

NSTAC OUT-An active low signal indicating the 50 STAC DRAM buffer manager's address space has been decoded from microcontroller 105 memory space.

NSCSI OUT-An active low signal indicating the 55 SCSI controller's address space has been decoded from microcontroller 105 memory space.

NCSDC OUT-An active low signal indicating the data compression chip's address space has been decoded from microcontroller 105's memory space.

NIRQ INPU-An active low input indicating an interrupt request from the STAC buffer manager 103.

SCSINT INPU-An active high input indicating an interrupt request from the SCSI controller 102.

NWAIT IN-An active low input indicating 1) buffer manager 103 is not ready to complete the current cycle to SCSI Controller 102 or itself or 2) indicating the data compression chip is not ready to com-

46

plete the current cycle to itself. During buffer manager 103 and SCSI and data compression chip accesses, if this signal is active, the READY signal goes inactive.

SRAM 117 signals

RA[0..4] OUT Address lines for SRAM 117.

RD[0..7] BI -Data bus for SRAM 117

NRWR OUT-Write pulse for SRAM 117

NRRD OUT-Read pulse for SRAM 117

Read/Write Channel 133 Signals

WDAT OUT-Serialized encoded data to write channel 133. When a 1 is to be written, a high going pulse is generated. The pulse width is equal to half of the codebit timing.

REFCLK OUT-Reference clock for WRDAT and data separator 118. It runs at the data bit clock (half of the SCLOCK rate) and is referenced to SCLOCK.

RAWCOD IN-Raw code from read channel 133. Used for Pre-Amble detect (all ones pattern).

RDAT IN-Serial data from data separator 118.

RDCLK IN-Data clock from data separator 118.

RDGATE OUT-Signal to the read channel 133 and data separator 118 which is active during reading. It tells data separator 118 to track the data stream instead of the reference clock 118

FLOCK OUT-Signal to data separator 118 telling it whether to lock to the frequency or lock to the phase. When locking to the frequency, data separator 118 can change quickly, when locking to phase, it changes slowly.

NWRGAT OUT-An active low signal to write channel 133 indicating writing is currently in progress.

NPILOT OUT-An active low signal indicating the pilot signal is currently being written.

The preferred embodiment of formatter 110 is presented in Microfiche Appendix A containing a pin out diagram and a series of 90 gate level logic schematic diagrams representing approximately 12,000 logic gates. The figures in Microfiche Appendix A present an example of how the above-described functions, registers, resources, sequences, and test modes may be implemented in hardware.

In conventional systems, functions performed by the 45 two-chip system comprising formatter 110 and frame buffer 117 were performed by 4 to 10 chips. The present invention reduces parts count, costs, complexity, and increases reliability. It will be appreciated by those skilled in the electronic arts that an alternate embodiment integrating frame buffer 117 with formatter 110 produces a single chip formatter, even further reducing parts, costs, complexity, and increasing reliability.

In summary, the advantages of the present invention include a significant reduction in parts count, costs, and electronic complexity and subsequent increased reliability over the prior art. Further, the invention achieves a reduction in RAM requirements, higher data transfer rates using an independent oscillator, on-the-fly error correction, a faster parallel functioning randomizer/-

60 derandomizer, provision for reading tracks with missing sync marks at the beginning of its main data field using only simple C2 error correction. Still further, in RAW operation, a bad frame may be re-written with only one intervening frames as contrasted with prior art systems requiring two to five intervening frames which wasted tape space, an independent track checksum unit acts on RAW data which allows more stringent RAW criteria resulting in higher data integrity.

5,359,468

47

An embodiment of this invention can be found in Digital Data Tape Storage System Model RB100 manufactured by R-Byte of San Jose, Calif.

While the above description provides a full and complete description of the preferred embodiments of the present invention, various modifications, alternate constructions, and equivalents may be employed while still remaining within the scope of the invention. For example, although the preferred embodiment described was a 1.0 micron CMOS gate array ASIC, other technologies may be advantageously implemented to reach the objectives of the present invention. These include standard cells, full custom integrated circuits, high-density programmable logic, and other implementation technologies known to those skilled in the electronic arts. Other functions may also be included on the formatter chip including a frame buffer, a main data buffer manager, a microcontroller, ATF logic, and an interface unit. Therefore, the above description and illustrations should not be construed as limiting the scope of the invention which is defined by the following claims.

## APPENDIX I

Register MAP			
	R=Read Only, RW=Read/Write	25	
00	Not used		
01	SCSI Select Input Register	R	
02	Track 1 Checksum High Byte	R	
02	Start ECC	W	
03	Track 1 Checksum Low Byte	R	
03	Initialize DMA	W	
04	Track 2 Checksum High Byte	R	
05	Track 2 Checksum Low Byte	R	
05	Clear Arbitration Error	W	
06	DMA Control Register	RW	
07	Bad C1 Block Counter Register	RW	
08	Read Track Checksum High Byte	RW	
09	Read Track Checksum Low Byte	RW	
0A	C2 Block Address	R	
0B	ECC Control Register	RW	
0C	Serdess Control Register 0	RW	
0D	Serdess Control Register 1	RW	
0E	Status Register	RW	
0F	Bit Pointer	RW	
10	Byte Counter	R	
11	Block Counter	RW	
12	State Sequence	R	
13	Interrupt Register	RW	
14	Serial Port Address/Status Register	RW	
15	Serial Port Data Register	RW	
16	Galois Multiplier Value 1	RW	
17	Galois Multiplier Value 2	RW	
18	Galois Multiplier Result	R	
19	Galois Square Root	R	

## APPENDIX II

Symbol	Description	MIN	MAX	Units
t1	Address Valid to NRRD Active	tcyc-3.3	ns	55
t2	NRRD Inactive to Address Change	2.2	ns	
t3	Data Valid to NRRD Inactive	32.1	ns	
t4	Data hold from NRRD Inactive	0	ns	60
t5	Address Valid to NRWR Active	1/2tcyc-3.3	ns	
t6	NRWR Inactive to Address Change	1/2tcyc+4.4	ns	
t7	Data Valid to NRWR Inactive	1/2tcyc-30.7	ns	65
t8	Data Hold from NRWR Inactive	1/2tcyc+3.3	ns	
t9	NRRD pulse width	2tcyc-1.7	ns	

48

## APPENDIX II-continued

Symbol	Description	MIN	MAX	Units
t10	NRWR pulse width	2tcyc-1.5	ns	

## APPENDIX III

Symbol	Description	Min	Max	Units
t1	Address Valid to NSTAC Active		27.5	ns
t2	Address change to NSTAC inactive	8.2		ns
t3	Address Valid to NSCSI Active		26.3	ns
t4	Address Change to NSCSI Inactive	10.0		ns
t5	Address Valid to NSCDC Active		29.9	ns
t6	Address Change to NCSDC Inactive	8.4		ns
t7	NWAIT Active to RDY Inactive		18.1	ns
t8	NWAIT Inactive to RDY Active		16.2	ns
t9	Address Valid to RDY (NSCSI cycle)		29.6	ns
t10	NWAIT Inactive to RDY (NSCSI cycle)	4tcyc	5tcyc+25.2	ns
t11	RDY Inactive (NSCSI cycle)	4tcyc		ns
t12	Address Valid to RDY (NCSDC cycle)		33.0	ns
t13	RDY Inactive (NCSDC cycle)	5tcyc		ns
t14	ALE Inactive to NROMOE Active		18.0	ns
t15	ALE active to NROMOE InActive		16.1	ns

## APPENDIX IV

CODE SPACE	
0000	Processor internal registers
00FF	
0100	Formatter ASIC
013F	25 registers
0140	SCSI controller
017F	8 registers
0180	DRAM Buffer Manager
01BF	32 registers
01C0	Data Compression Chip
01FF	32 registers
0200	ROM
FFFF	63.5K bytes
DATA SPACE	
0000	Processor internal registers
00FF	
0100	Formatter ASIC
013F	25 registers
0140	SCSI controller
017F	8 registers
0180	Buffer Manager
01FF	64 registers
0200	ROM
7FFF	31.5K bytes
8000	SRAM
FFFF	32K bytes

## APPENDIX V

8000	Frame Buffer 0 - Track A(+)
8FFF	
9000	Frame Buffer 0 - Track B(-)
9FFF	
A000	Frame Buffer 1 - Track A(+)
AFFF	
B000	Frame Buffer 1 - Track B(-)
BFFF	
C000	Frame Buffer 2 - Track A(+)
CF00	

## APPENDIX V-continued

D000	Frame Buffer 2 - Track B(-)		
FFFF	Processor and FORMATTER ASIC Scratch		
E000	Processor and FORMATTER ASIC Scratch		
FFFF	Area		
<u>SRAM address E000h</u>			
E000	Buffer 0 - Subdata write data	W	
E03F			2 Blocks
E040			
E07F			
E080	Buffer 0 - W1 eight block cycle	W	
E087			
E088			
E097			
E098	Buffer 0 - SW1 for even blocks	W	
E099	Buffer 0 - SW2 for even blocks	W	
E09A	Buffer 0 - SW1 for odd blocks	W	
E09B	Buffer 0 - SW2 for odd blocks	W	
<u>E09C-E0BF</u>			
E0C0	Buffer 0 - ID Entry for Main	R	
E0D7	data blocks - Track A (+)	8 entries	
E0D8	Buffer 0 - ID Entry for Subdata	R	
E0DD	blocks - Track A (+)	2 entries	
<u>E0DE-E0DF</u>			
E0E0	Buffer 0 - ID Entry for Main	R	
E0F7	data blocks - Track B (-)	8 entries	
E0F8	Buffer 0 - ID Entry for Subdata	R	
E0FD	blocks - Track B (-)	2 entries	
<u>E0FE-E0FF</u>			
E100	Buffer 0 - Subdata read data -	R	
E13F	Track A(+)	2 blocks	
E140	Buffer 0 - Subdata read data -	R	
E17F	Track B(-)	2 blocks	
E180	Buffer 0 - Bad C1 Block	R	
E1FF	Pointers/RAW Syndromes		
<u>SRAM address E200h</u>			
E200	Buffer 1 - Subdata write data	W	
E23F			2 Blocks
E240			
E27F			
E280	Buffer 1 - W1 eight block cycle	W	
E287			
E288			
E297			
E298	Buffer 1 - SW1 for even blocks	W	
E299	Buffer 1 - SW2 for even blocks	W	
E29A	Buffer 1 - SW1 for odd blocks	W	
E29B	Buffer 1 - SW2 for odd blocks	W	
<u>E29C-E2BF</u>			
E2C0	Buffer 1 - ID Entry for Main	R	
E2D7	data blocks - Track A (+)	8 entries	
E2D8	Buffer 1 - ID Entry for Subdata	R	
E0DD	blocks - Track A (+)	2 entries	
<u>E2DE-E2DF</u>			
E2E0	Buffer 1 - ID Entry for Main	R	
E0F7	data blocks - Track B (-)	8 entries	
E2F8	Buffer 1 - ID Entry for Subdata	R	
E2FD	blocks - Track B (-)	2 entries	
<u>E2FE-E2FF</u>			
E300	Buffer 1 - Subdata read data -	R	
E33F	Track A(+)	2 blocks	
E340	Buffer 1 - Subdata read data -	R	
E37F	Track B(-)	2 blocks	
E380	Buffer 1 - Bad C1 Block	R	
E1FF	Pointers/RAW Syndromes		
<u>SRAM address E400h</u>			
E400	Buffer 2 - Subdata write data	W	
E43F			2 Blocks
E440			
E47F			
E480	Buffer 2 - W1 eight block cycle	W	
E487			
E488			
E497			
E498	Buffer 2 - SW1 for even blocks	W	
E499	Buffer 2 - SW2 for even blocks	W	
E49A	Buffer 2 - SW1 for odd blocks	W	
E49B	Buffer 2 - SW2 for odd blocks	W	
<u>E49C-E4BF</u>			
E4C0	Buffer 2 - ID Entry for Main	R	
E0D7	data blocks - Track A (+)	8 entries	
E4D8	Buffer 2 - ID Entry for Subdata	R	

## APPENDIX V-continued

E4DD	blocks - Track A(+)	2 entries
E4DE-E4DF		
E4E0	Buffer 2 - ID Entry for Main	R
5 E4F7	data blocks - Track B(-)	8 entries
E4F8	Buffer 2 - ID Entry for Subdata	R
E4FD	blocks - Track B(-)	2 entries
E4FE-E4FF		
E500	Buffer 2 - Subdata read data -	R
E53F	Track A(+)	2 blocks
10 E540	Buffer 2 - Subdata read data -	R
E57F	Track B(-)	2 blocks
E580	Buffer 2 - Bad C1 Block	R
E5FF	Pointers/RAW Syndromes	

## APPENDIX VI

	INPUT PIN tied to OUTPUT PIN	INPUT PIN tied to OUTPUT PIN
R	A15(91)	AD0(7)
R	A14(92)	AD1(6)
20	A13(93)	AD2(5)
	A12(94)	AD3(4)
	A11(95)	AD4(2)
	A10(96)	AD5(1)
	A9(97)	AD7(99)
	A8(98)	AD6(100)
25	NRESET(72)	LA0(74)
	SCSINT(73)	LA1(75)
	NRESET(72)	LA2(76)
	INST(89)	LA3(77)
	NRD*(87)	LA4(79)
	NWR*(85)	LA5(80)
30	FMTT1(83)	LA6(81)
	TRKCLK(84)	LA7(82)
	NIRQ(67)	NSTAC(61)
	NFDACK(64)	NSCSI(62)
	NWAIT(66)	NFDREQ(63)
	SCSINT(73)	NRESET(72)
	FINT(71)	NROMOE(70)
35	RAWCOD(12)	NWRGAT(14)
	SL2(8)	SL1(9)
	REFCLK(17)	RDAT(20)
	RDCLK(19)	RDGATE(21)
	RDCLK(19)	SL0(10)
	SDAT(23)	WDAT(13)

\*These signals are inverted before being driven out.

W	40	What is claimed is:
R		1. A formatter write device for a digital data storage
R		system, said digital data storage system having a data
		bus, a main buffer, a frame buffer, a frame buffer bus, a
R		microcontroller, and a head drum in communication
45		with magnetic tape comprising:
R		randomizer means for randomizing data, said ran-
		domizer means coupled to the data bus;
R		track checksum means for computing a running track
		checksum, said track checksum means coupled to
		said randomizer means;
R	50	data interleaving means for interleaving data, said
		data interleaving means coupled to said track
		checksum means;
R		arbiter means for arbiting access to the frame buffer,
		said arbiter means coupled to frame buffer;
55		C2 parity byte computing means for generating C2
		parity bytes, said C2 parity byte computing means
		coupled to said interleaving means and said arbiter
		means;
W	60	C1 parity byte computing means for generating C1
		parity bytes, said C1 parity byte computing means
		coupled to the frame buffer bus;
W		8-10 conversion means for performing 8-10 coding
		conversion of data, said 8-10 conversion means
		coupled to the frame buffer bus;
W	65	track format data generating means for generating
		track format data and for synchronizing data writ-
		ing with the head drum means, said track format

data generating means coupled to the frame buffer bus and to the head drum; and  
status register means for registering the status of a write operation, said status register means coupled to the microcontroller and to the frame buffer bus. 5  
2. The formatter write device of claim 1 wherein said randomizer means is parallel-functional.  
3. The formatter write device of claim 1 wherein said randomizer means performs randomization on the fly during data transfers from the main buffer to the frame 10 buffer.  
4. The formatter write device of claim 1 wherein said interleaving means performs interleaving on the fly during data transfers from the main buffer to the frame 15 buffer.  
5. The formatter write devices of claim 1 wherein said C1 parity byte computing means computes C1 parity bytes on the fly during data transfers from the frame buffer to the tape.  
6. The formatter write device of claim 1 wherein said data is in DDS standard format. 20  
7. A formatter read device for a digital data storage system (DDS), said DDS system having a frame buffer, a microcontroller, and read/write channel coupled to a data separator comprising:  
deserializer means for converting serial data from magnetic tape to parallel, said deserializer means coupled to the data separator;  
8-10 decoder means for decoding said data, said 8-10 30 decoder means coupled to said deserializer means; track checksum means for performing track checksum on said data, said track checksum means coupled to said decoder 8-10 means;  
C1 syndrome generator means for computing C1 syndromes on the fly and for writing both data and syndromes into the frame buffer, said C1 syndrome generator means coupled to said track checksum means;  
frame buffer manager means for managing data transmitted to the frame buffer, said frame buffer manager means coupled to said C1 syndrome generator 35 means;  
arbiter means for arbiting access to the frame buffer, said arbiter means coupled to said frame buffer manager means; and  
C2 error correction coding means for generating C2 error correction code, said C2 error correction coding means coupled to said frame buffer manager means and to said arbiter means. 40  
8. The formatter read device of claim 7 wherein said track checksum means is independent from any write operation track checksum.  
9. In a digital data storage audio tape system having a host device interface unit that is coupled to a main data buffer manager, the main data buffer manager coupled to a main data buffer means, the digital data storage audio tape system further including a processor means, a tape drive unit, a read/ write channel, and a DMA channel bus, a formatter comprising:

formatter interface means for interfacing the processor means, said formatter interface means coupled between said processor means and an internal processor bus; 60  
formatter sequencer means for sequencing data and data operations, said formatter sequencer means coupled to said internal processor bus and to the read/write channel;

formatter buffer manager means for buffering data between said formatter interface means and said formatter sequencer means, said formatter buffer manager means coupled to said formatter sequencer means by an internal read/write bus and to said internal processor bus, and further coupled to the DMA channel bus;  
said formatter interface means including an address map decoder, ready logic, interrupt control, serial port handshake logic, and a plurality of registers including an interrupt, a serial port address/status, a serial port data, a Galois field value 1, a Galois field value 2, a Galois field result, a Galois square root, and a jumper input;  
said formatter sequencer means including a C1 parity generator, a C1 syndrome generator, a read/write sequencer, an encoder/decoder, an automatic track finding (ATF) pattern generator, and an ID parity, and a plurality of registers including a serial/deserializer (serdes) control 0, a serdes control 1, a status, two read track checksum, and a bit pointer, a state counter, a block counter, and a byte counter; and  
said formatter buffer manager means including a C2 parity generator, a C2 syndrome generator, a data path control, a buffer address generator, a direct memory access (DMA) arbitrator, a DMA handshake, and a randomizer, and a plurality of registers including a DMA controller control, SRAM access latch, two track 1 checksum, two track 2 checksum, an ECC control, a control pulse, a C2 block address, and a bad C1 block count.  
10. The formatter of claim 9 further comprising tri-state and activity modes means for automatic test equipment testing.  
11. In a digital data storage magnetic tape system for transmitting data between a host device and a tape drive unit, the system including a formatter coupled to a frame buffer having at least one buffer pointer, and a processor, a method for read-after-write checking to determine whether a track was successful written, comprising the steps of:  
(a) multiple buffering the frame buffer wherein the processor stores a copy of the frame which is being read-after-write checked while simultaneously processing new frames for writing; and  
(b) immediately rewriting an improperly written frame by changing the buffer pointer in the frame buffer and updating relevant sub data areas under instruction control of said processor.  
50 12. In a digital data storage magnetic tape system for transmitting data between a host device and a tape drive unit having a read channel and a frame buffer, the data being accessed and encoded and decoded on a per frame and per track basis, and the data being in a main data area and subject to C1 error correction code, read track checksum, a method for read-after-write (RAW) checking comprising the steps of:  
re-writing a frame when more than a predetermined number of blocks in any area of either track of the frame have C1 syndromes indicating an error; calculating read track checksums as the data is transferred from the read channel to the frame buffer; and  
re-writing the frame when a track checksum discrepancy is found, thereby achieving re-writing of a bad frame after only one intervening frame.  
13. The RAW checking method of claim 12 further comprising the steps of:

5,359,468

53

re-computing the track checksums during the main data area read;  
comparing the track checksums with a stored value having been written for that track; and  
comparing the track checksum to the value read back from one good subdata block pair, thereby avoiding drop-ins in the main data area of the frame.

14. The RAW checking method of claim 13 wherein said track checksum checking acts on RAW data, thereby allowing more stringent RAW criteria resulting in higher data integrity.

15. A method of formatting data comprising the operational steps of:

interleaving;  
randomizing;  
C1/C2 ECC generation;  
subdata block processing;  
sync generation;  
automatic track finding signal generation;  
deinterleaving; and  
derandomizing.

16. A method of formatting data comprising the operational steps of:

- (a) assembling data in a main data buffer on a data group basis under the instruction control of a formatter and a processor;
- (b) generating relevant system data including group number, filemark, save set marks for the data group in the main data buffer means by the processor;
- (c) transferring the data group responsive to instruction control by the processor to a frame buffer means;
- (d) moving system data for the single frame to the frame buffer means under the instruction control of the processor;
- (e) computing C2 parity bytes with the formatter;
- (f) writing to the frame buffer on a frame basis with the formatter;
- (g) reading data from the frame buffer, and before outputting the serial data stream:
  - (i) computing C1 parity bytes;
  - (ii) performing a data code conversion;
  - (iii) generating track format data including automatic track finding signals, margins, IBGs, preambles, sub data; and

(h) determining if a proper operation was completed on a frame of data at the end of the frame with the processor, and if proper, writing the next data frame from the beginning of the generation of system data with the formatter under the instruction control of the processor.

17. The method of claim 16 wherein the transferring of the data group responsive to instruction control step comprises the steps of:

initializing a DMA controller and a main data buffer manager and moving one frame of data from the main data buffer to the formatter through the main data buffer manager;  
randomizing each byte of the data;  
computing a running track checksum of transferred bytes of the data;  
interleaving each bytes of the data; and  
storing each byte of the data in the frame buffer.

18. The method of claim 17 wherein said randomizing step is performed in parallel.

19. The method of claim 17 wherein said randomizing step is performed on the fly before that data is written to frame buffer.

54

20. The method of claim 17 wherein said running track checksum step is computed on the fly before that data is written to said frame buffer.

21. The method of claim 17 wherein said interleaving step is performed on the fly before that data is written to the frame buffer.

22. The method of claim 17 wherein said C1 parity byte computation step is performed on the fly before that data is written to magnetic tape.

23. The method of claim 17 wherein said data code conversion step is performed on the fly as frames are transferred between said frame buffer and magnetic tape.

24. The method of claim 17 wherein said data code conversion step is 8-10 GCR encoding and decoding.

25. The method of claim 17 wherein track format data is generated in said generating track format data step in synchronization with signals from a head drum in said digital storage magnetic tape system.

26. The method of claim 17 wherein the data is randomized compatible with the DDS standard.

27. The method of claim 17 wherein the data is interleaved compatible with the DDS standard.

28. In a digital data storage magnetic tape system for storing data from a host device to magnetic tape having a head drum, a main data buffer coupled to a main data buffer manager coupled to a host device interface, a formatter including a direct memory access controller, the formatter coupled to a frame buffer, and a processor coupled to the main data buffer manager and the formatter, a method of formatting data for accessing and encoding and decoding the data to and from magnetic tape, the data being organized into blocks and including system area data and track format data comprising the operational steps of:

- (a) transmitting synchronized serial data and clock pulses to the formatter;
- (b) deserializing the serial data to parallel;
- (c) computing C1 syndromes from the data;
- (d) buffer managing including branching on C1 syndrome value being no read errors for zero value and read errors for non-zero value;
- (e) storing pointers to block with non-zero syndrome values;
- (f) writing both the data and the syndromes into the frame buffer, thereby constituting a read;
- (g) reading a complete frame and thereafter reading the non-zero C1 syndrome block pointers for error correction;
- (h) generating C2 syndromes and pausing when a non-zero syndrome is generated, thereby performing error correction;
- (i) assembling a frame of data into the frame buffer and correcting all C1 and C2 errors, thereafter moving the data into the main data buffer; and
- (j) moving the data to the host device interface for communication with the host device.

29. The method of claim 28 wherein said C1 syndrome computation step (c) is performed on the fly for main data and sub data as data is moved to and from the tape.

30. In a processor having an address space, an interrupt register having an address space enable bit and a code bit, and a signal pin for signalling when an instruction byte is fetched, a method of increasing the address space comprising the steps of:

creating a first address space utilizing the signal pin and the addressing modes of the address space

## 5,359,468

**55**

enable and code bits to designate said first address space; and  
 creating a second address space utilizing the signal pin and the addressing modes the address space  
 enable and code bits to designate said second address space.

**31.** The method of claim 30 wherein addresses are decoded into said first (CODE) and second (DATA) decoded address spaces as:

Address Space	Enable bit	Code bit	Signal pin	Decoded Space
0	X		0	DATA
0	X		1	CODE
1	0		X	DATA
1	1		X	CODE

**32.** The method of claim 31 further comprising the step of writing to CODE space to support flash EPROM.

**33.** The method of claim 31 further comprising the step of executing from DATA space to support diagnostic firmware routines.

**34.** The method of claim 30 wherein said processor is an 80C198 microprocessor.

**35.** A formatter for a digital data storage tape system that encodes and decodes data comprising:

**56**

an interface to a processing unit, said interface coupled to a processor bus;  
 a sequencer, said sequencer coupled to said processor bus and to a read/write channel; and  
 a buffer manager coupled to said interface and to said sequencer by an read/write bus.

**36.** The formatter of claim 35 wherein said interface comprises an address map decoder, ready logic, interrupt control, serial port handshake logic, and a plurality of registers including an interrupt, a serial port address/status, a serial port data, a Galois field value 1, a Galois field value 2, a Galois field result, a Galois square root and a jumper input.

**37.** The formatter of claim 35 wherein said sequencer comprises a C1 parity generator, a C1 syndrome generator, a read/write sequencer, an encoder/decoder, an track finding pattern generator, an ID parity and a plurality of registers including a serial/deserializer control 0, a serial/deserializer control 1, a status, two read track checksum, and a bit pointer, a state counter, a block counter and a byte counter.

**38.** The formatter of claim 35 wherein said buffer manager comprises a C2 parity generator, a C2 syndrome generator, a data path control, a buffer address generator, a direct memory access arbitrator, a direct memory access handshake, and a randomizer, and a plurality of registers including a direct memory access controller control, SRAM access latch, two track 1 checksum, two track 2 checksum, an ECC control, a control pulse, a C2 block address and a bad C1 block count.

\* \* \* \* \*

35

40

45

50

55

60

65



**United States Patent** [19]  
Beachy

[11] Patent Number: **5,502,807**  
[45] Date of Patent: **Mar. 26, 1996**

[54] CONFIGURABLE VIDEO SEQUENCE  
VIEWING AND RECORDING SYSTEM

[75] Inventor: Jeffrey L. Beachy, Wilsonville, Oreg.

[73] Assignee: Tektronix, Inc., Wilsonville, Oreg.

[21] Appl. No.: 947,620

[22] Filed: Sep. 21, 1992

[51] Int. Cl.<sup>6</sup> ..... G06T 13/00

[52] U.S. Cl. ..... 395/152

[58] Field of Search ..... 395/128, 139,  
395/152, 164, 165; 345/122; 348/715

[56] References Cited

U.S. PATENT DOCUMENTS

Re. 33,316	8/1990	Katsuta et al.	.....	395/128
4,454,593	6/1984	Fleming et al.	.....	395/128
4,649,380	3/1987	Penna	.....	340/721
4,688,095	8/1987	Beg et al.	.....	358/160
4,703,356	10/1987	Herzog et al.	.....	358/160
4,703,439	10/1987	Lutz	.....	395/128
4,751,507	6/1988	Hama et al.	.....	340/721
4,774,582	9/1988	Hakamada et al.	.....	358/183
4,808,987	2/1989	Takeda et al.	.....	340/721
4,868,764	9/1989	Richards	.....	395/125
4,870,479	9/1989	Dubner	.....	358/18
4,980,765	12/1990	Kudo et al.	.....	358/160
5,018,014	5/1991	Hashimoto	.....	358/183

5,021,977	6/1991	Sakaguchi	.....	395/164
5,025,394	6/1991	Parke	.....	395/152
5,047,857	9/1991	Duffield et al.	.....	358/183
5,068,805	11/1991	Tsuzuki	.....	395/164
5,093,907	3/1992	Hwong et al.	.....	395/152
5,109,438	4/1992	Alves et al.	.....	358/433
5,119,081	6/1992	Ikehira	.....	395/128
5,319,460	6/1994	Kubo	.....	348/715

FOREIGN PATENT DOCUMENTS

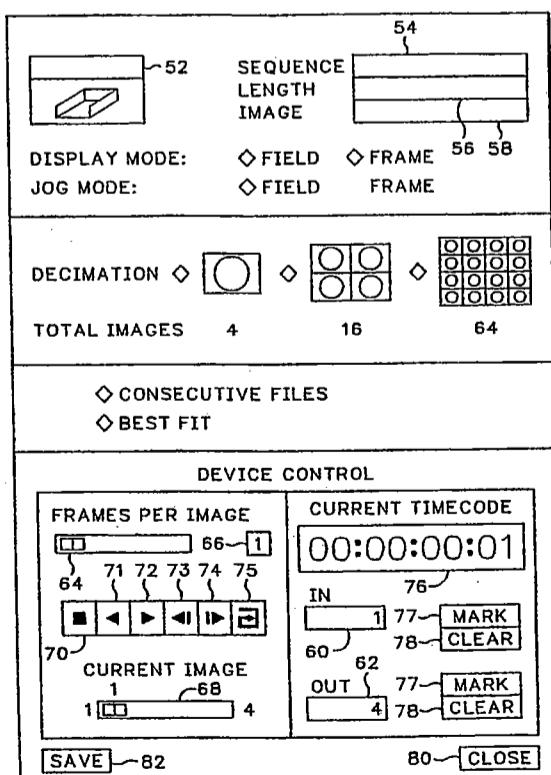
55-34706 3/1980 Japan

Primary Examiner—Heather R. Herndon  
Assistant Examiner—Anton Fetting  
Attorney, Agent, or Firm—Francis I. Gray

[57] ABSTRACT

A configurable video sequence viewing and recording system stores multiple computer generated images of a sequence in a frame buffer and plays back the images for animation motion study. The images are stored either in a full size format or in a decimated format according to a decimation factor. The images are read out in real time for display from the frame buffer, with pixels/lines/frames being replicated to provide full size images for a desired number of frames each. The number of images of a sequence to be displayed/stored are determined by an operator, and a loop function is provided so that the sequence may be continuously displayed.

5 Claims, 5 Drawing Sheets

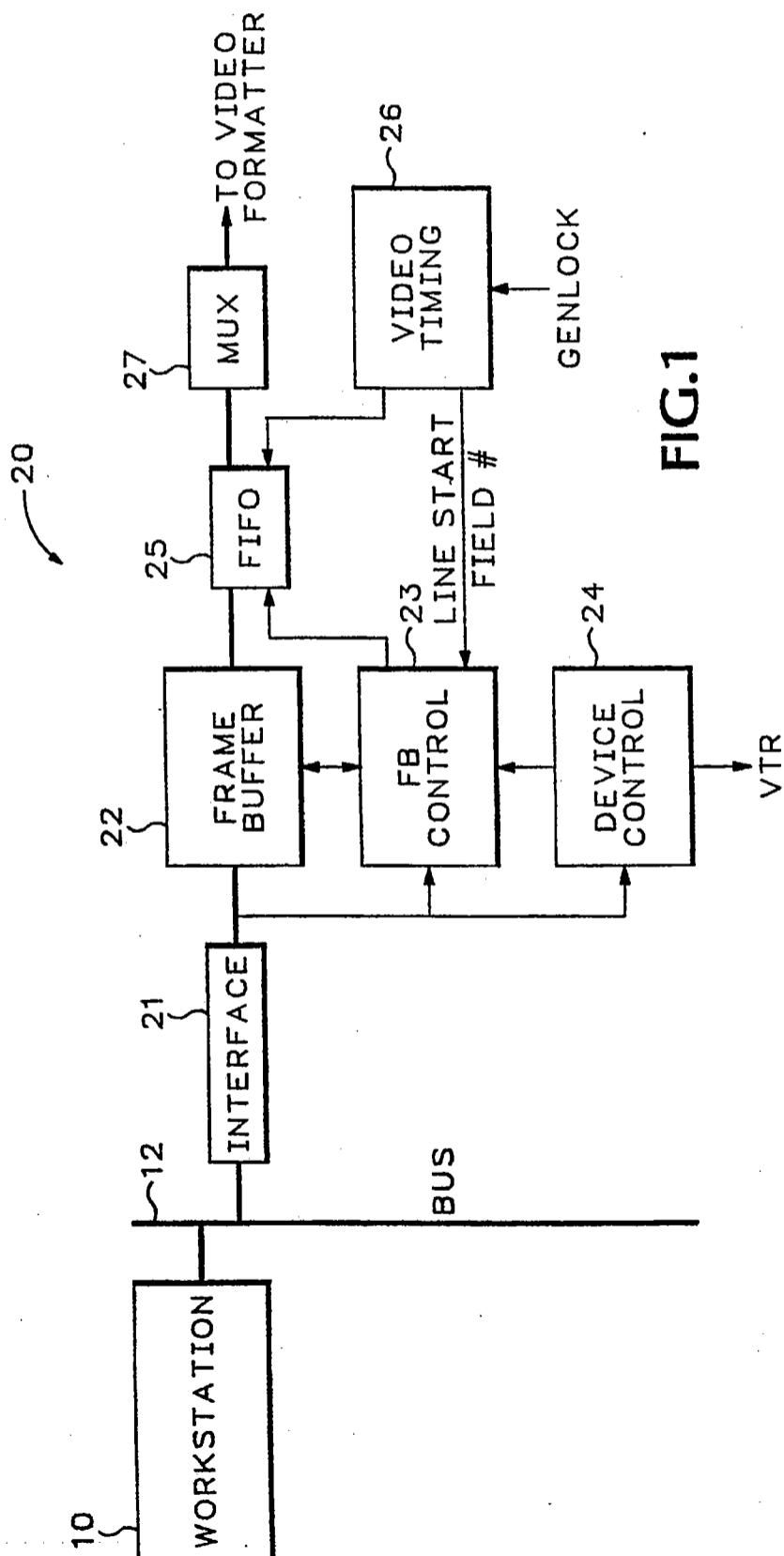


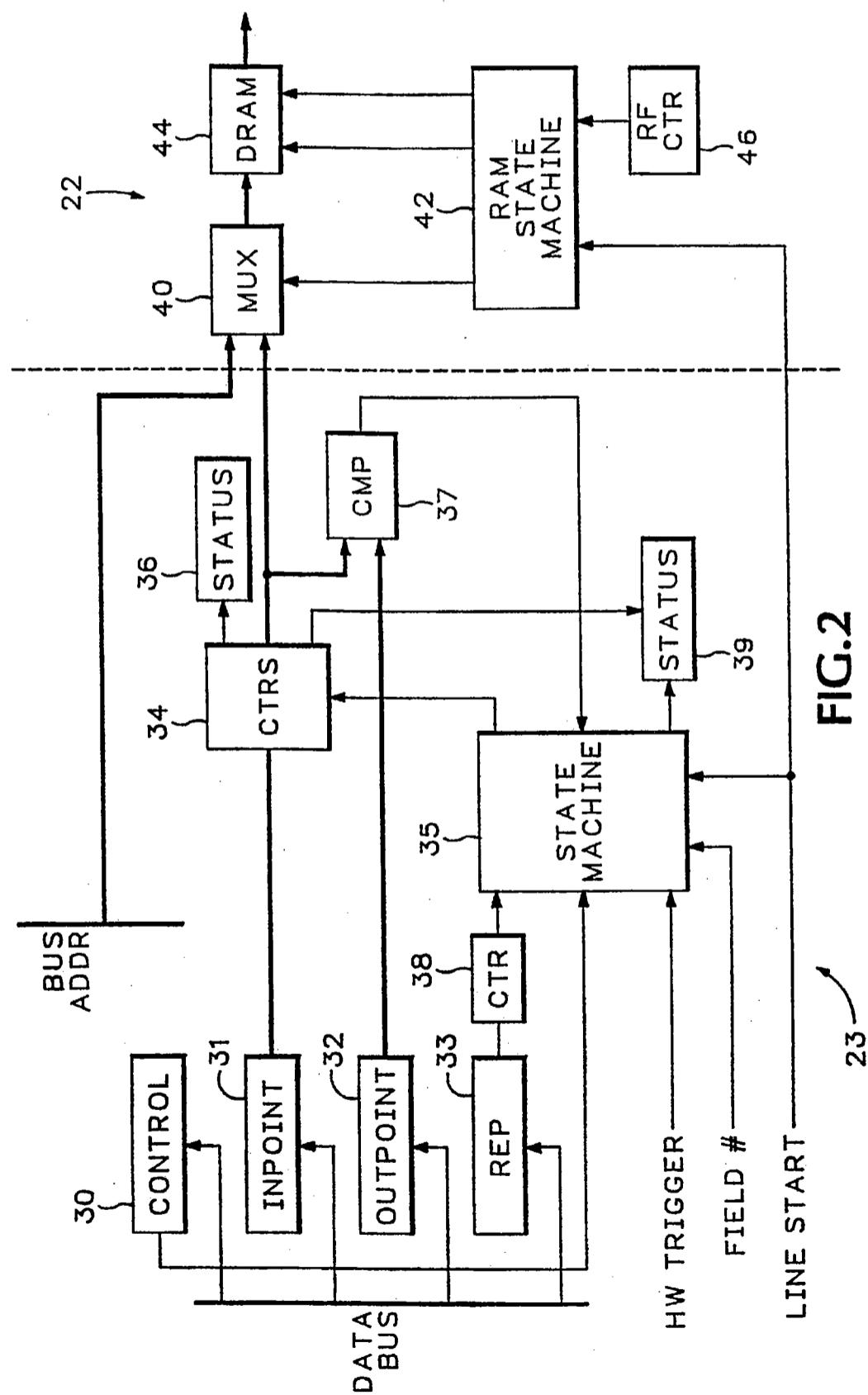
U.S. Patent

Mar. 26, 1996

Sheet 1 of 5

5,502,807





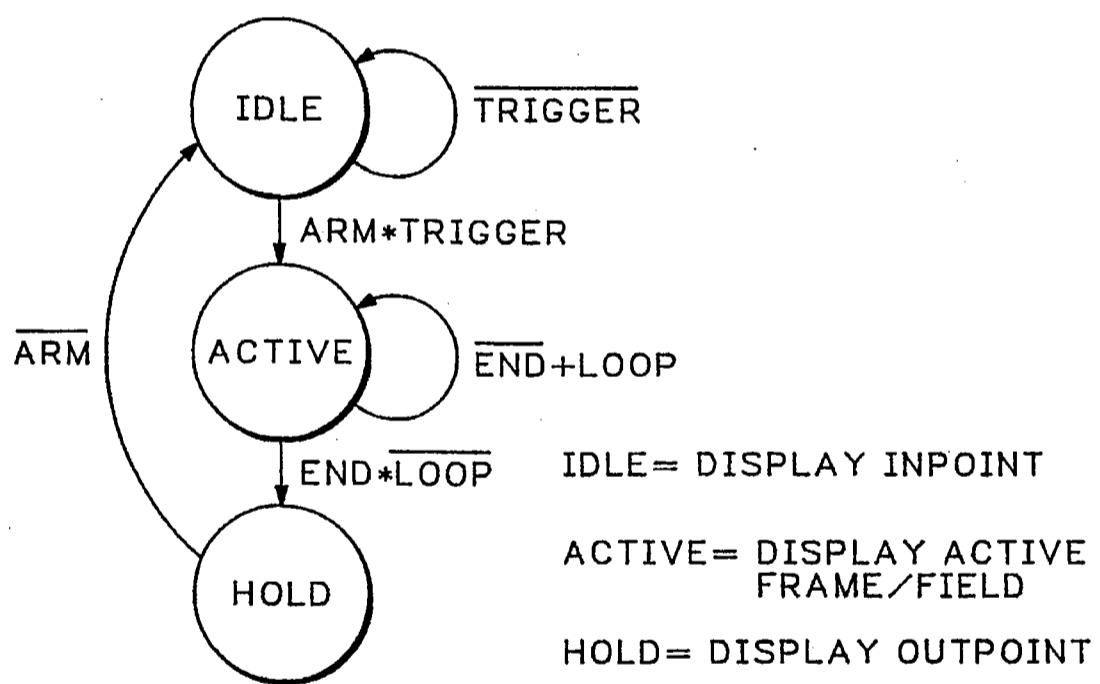
23

U.S. Patent

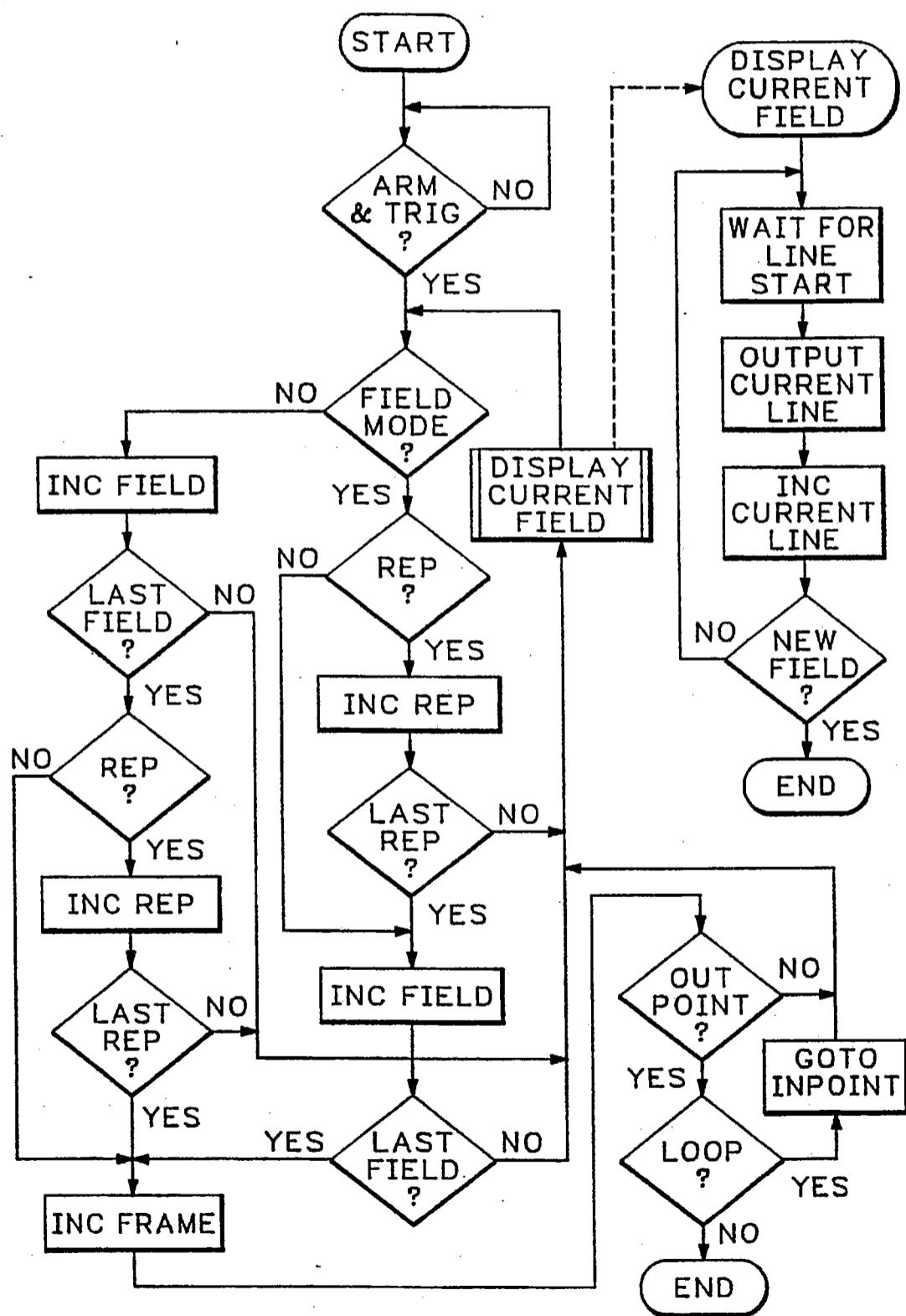
Mar. 26, 1996

Sheet 3 of 5

5,502,807



**FIG.3**



**FIG.4**

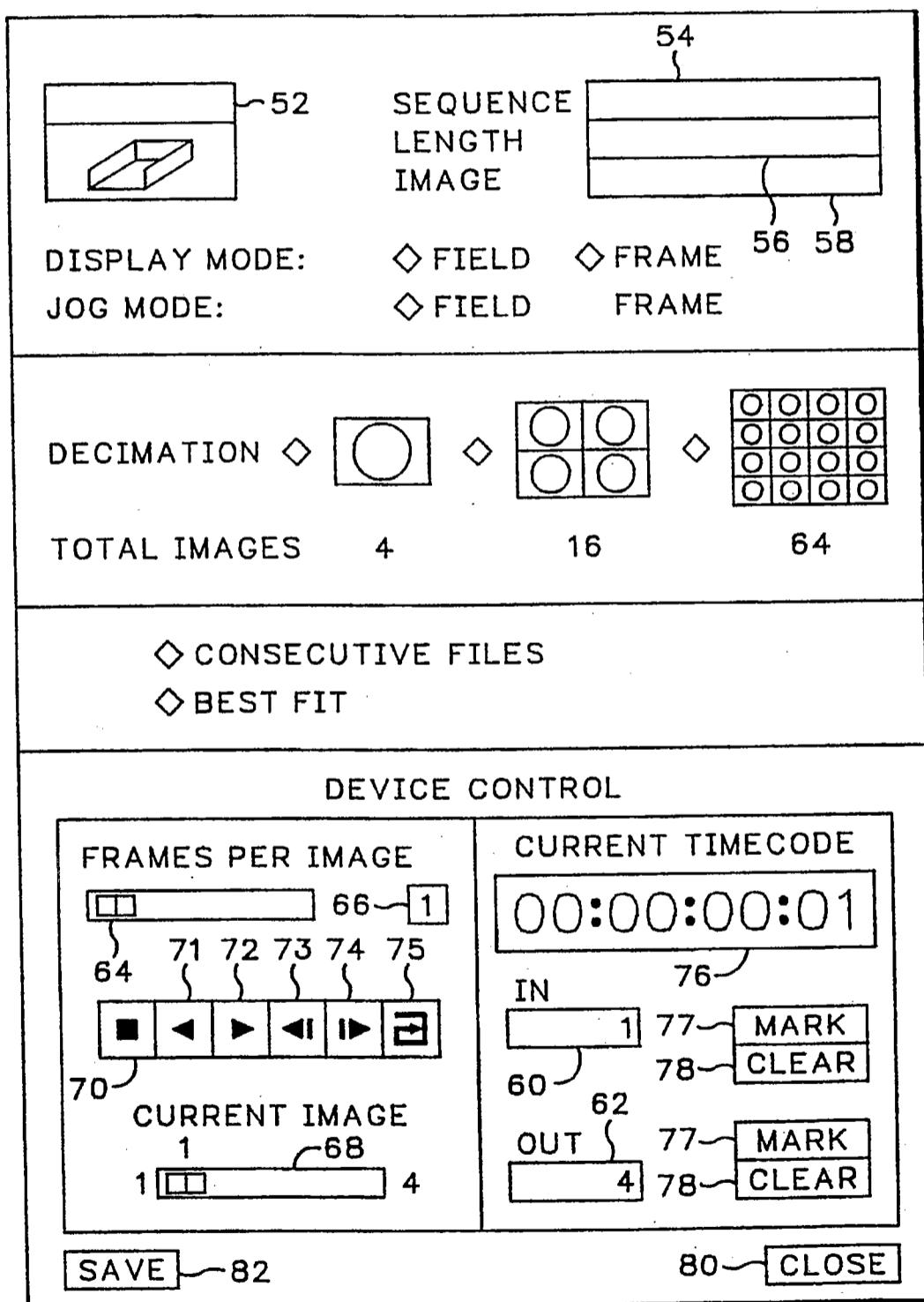


FIG.5

**1****CONFIGURABLE VIDEO SEQUENCE  
VIEWING AND RECORDING SYSTEM****BACKGROUND OF THE INVENTION**

The present invention relates to computer generated graphics systems, and more particularly to a configurable video sequence viewing and recording system that provides the ability to view video sequences, regardless of the size at which the image is rendered, in full size and in realtime on a video monitor.

In computer generated graphics systems an operator renders a sequence of images to create an animation sequence. When the operator completes the sequence, he/she performs a motion study to verify the sequence. The operator may perform the motion study by stepping through the sequence on the computer display. However it is preferable to convert the sequence to a video sequence and view it on a video monitor. The video memory generally is limited to a four frame buffer, i.e., only four frames of video may be viewed at one time. Since one frame of video is displayed in 1/30th of a second, this allows only a very small portion of the sequence to be displayed and run for motion study at a time. Since animation sequences may be 15-30 seconds in duration, motion studies become a very time consuming process.

What is desired is a video viewing and recording system that allows an operator to conduct a motion study on animation sequences that significantly reduces the amount of time required to complete the study.

**SUMMARY OF THE INVENTION**

Accordingly the present invention provides a configurable video sequence viewing and storing system that provides the ability to view video sequences in full size and in realtime on a video monitor. An animation sequence is generated on a computer workstation by an operator and then stored in a frame buffer under control of a video timing and frame buffer controller. The animation sequence may be decimated under control of the operator as it is input to the frame buffer so that, for example, a four frame buffer memory may contain up to sixty-four image frames of the sequence. The effective length of the sequence further may be extended by decreasing the viewing rate to less than thirty frames per second. Frames may further be decimated in the vertical dimension and displayed as fields to double the effective sequence length. Programmable logic arrays are used to implement the controllers so that appropriate sub-sets of the features may be provided with reduced hardware since all features need not be required at once.

The objects, advantages and novel features of the present invention are apparent from the following detailed description when read in light of the appended claims and attached drawing.

**BRIEF DESCRIPTION OF THE DRAWING**

FIG. 1 is a block diagram of a configurable video sequence viewing and recording system according to the present invention.

FIG. 2 is a block diagram of a programmable logic array configured to implement the configurable video sequence viewing and recording system according to the present invention.

FIG. 3 is a state diagram for a state machine in the programmable logic array of FIG. 2.

**2**

FIG. 4 is a flow chart for the display output of the configurable video sequence viewing and recording system according to the present invention.

FIG. 5 is a plan view of a control panel for a configurable video sequence viewing and recording system according to the present invention.

**DESCRIPTION OF THE PREFERRED  
EMBODIMENT**

Referring now to FIG. 1 a computer graphics workstation 10, having a central processing unit, memory and mass storage, is coupled to a computer bus 12, such as a VMEbus. Also coupled to the VMEbus 12 is a video systems module 20, such as the AVANZAR Video Systems card manufactured by Tektronix, Inc. of Wilsonville, Oreg., United States of America. A bus interface circuit 21 couples the video systems card 20 to the VMEbus 12. A frame buffer 22 receives data from the workstation 10 and stores it under control of a frame buffer controller 23. The frame buffer controller 23 receives commands from the workstation 10 to store and readout data to and from the frame buffer 22 according to a feature selected by an operator at the workstation. Commands from the workstation 10 also are received by a device controller 24 that controls external video storage devices, such as video tape recorders (VTR). The device controller 24 also provides information to the frame buffer controller 23. Data from the frame buffer 22 is input to a buffer register 25, such as a first-in, first-out (FIFO) buffer, under control of the frame buffer controller 23. The video timing circuit 26 generates all necessary timing signals from a genlock input clock signal to read the data from the buffer register 25. The video timing circuit 26 provides line start and field number data to the frame buffer controller 23. The output of the buffer register 25 is coupled to a register/multiplexer 27 which outputs the digital data for subsequent conversion into a desired video format for display on a video monitor and/or recording on an appropriate video storage device.

The frame buffer controller 23 is shown in greater detail in FIG. 2. A plurality of registers 30-33 store data values from the workstation 10 via the VMEbus 12. The registers include a CONTROL register 30, an INPOINT register 31, an OUTPOINT register 32 and a REP register 33. The output of the CONTROL register 30 is input to a state machine 35. Also input to the state machine 35 is an external hardware trigger, and the line start and field number data from the video timing circuit 26. The data from the INPOINT register 31 is input to video counters 34. The output of the video counters 34 is input to a comparator 37 together with the data from the OUTPOINT register 32. The output of the comparator 37 is input to the state machine 35. The output of the REP register 33 is input to a repetition counter 38, the output of which also is input to the state machine 35. A status register 36 is coupled to the video counters 34, and another status register 39 is coupled to both the video counters and the state machine 35.

The output from the video counters 34 is input to an address multiplexer 40 of the frame buffer 22 together with an address from the VMEbus 12. A memory state machine 42 selects which address the multiplexer 40 passes on to a memory 44, such as a DRAM, that stores the data from the workstation 10 representing image files of the sequence to be displayed. Also input to the memory state machine 42 is the line start data from the video timing circuit 26 via the frame buffer controller 23 and an input from a refresh

counter 46. The memory state machine 42 controls the DRAM 44.

The display process for the frame buffer controller state machine 23 is represented by the state diagram of FIG. 3. The state machine has three states: IDLE, ACTIVE and HOLD. In the IDLE state the INPOINT is displayed, i.e., the first image of the sequence appears on the video monitor. So long as an ARM signal and a TRIGGER signal are absent, the state machine remains in the IDLE state. The ARM signal acts as an enable signal, and the TRIGGER signal acts as a start signal. When the ARM and TRIGGER signals are present, the state machine transitions to the ACTIVE state and displays the currently active frame or field, depending upon the mode of operation, which changes as the state machine controls the readout of the images of the sequence from the frame buffer 22. The state machine remains in the ACTIVE state so long as there is no END signal or there is a LOOP signal. The END signal acts as a stop signal, and the LOOP signal acts as a continuous activation, or loop, signal. When the END signal occurs and there is no LOOP signal, the state machine transitions to the HOLD state. In the HOLD state the OUTPOINT image of the sequence is displayed on the video monitor.

This process is described in further detail by the flowchart of FIG. 4. The first test is whether both the ARM and TRIGGER signals are present. If the answer is NO, no further action occurs. If the answer is YES, then the mode is tested for FIELD or FRAME mode. If in the FRAME mode, the FIELD number is incremented and tested to determine whether it is the last field of a frame. If NO, then the current field is displayed. If YES, then the REP register is tested to determine whether the frame is to be repeated. If YES, then the REP count is incremented and tested to determine whether it is the last repetition. If NO, then the current field is displayed again. If YES or if no repetition is indicated, then the frame is incremented. If in the FIELD mode the repetition loop is accessed as described above. If there are no repetitions, then the field is incremented and tested. If NO, then the current field is displayed, otherwise if YES, the frame is incremented.

Since each frame represents an image of the sequence stored in the frame buffer 22, after the frame is incremented OUTPOINT is tested to determine whether the last image of the sequence is indicated. If NO, then the current field is displayed. If YES, then a test is made of LOOP. If at OUTPOINT and not in LOOP, then the HOLD state is entered and the last image of the sequence is displayed. If in LOOP, then the process returns to the INPOINT image and displays the current field. Once the current field is displayed, the process returns to the FIELD/FRAME test step and repeats the process.

The step of displaying the current field waits until a LINE START signal from the video timing circuit 26 occurs. Then the current line is output and incremented. The field number data from the video timing circuit 26 is then tested to determine whether a new field is indicated. If NO, then the next line is output when the next LINE START signal occurs. Otherwise the step exits from the display process and returns to the FIELD/FRAME mode test step.

Referring now to FIG. 5 in operation when this system is accessed an interface panel 50 is displayed at the workstation 10. When a sequence is selected, a file folder or other indicia appears in the basket 52 and the sequence name appears in an appropriate window 54. The length of the sequence (in images) appears in another window 56, and a file name for the current image, which is also displayed on

the video monitor, appears in yet another window 58. A LOAD button (not shown) appears. The display mode, FIELD or FRAME, as well as the jog mode, is then selected. Finally the decimation factor is selected. In this particular embodiment either four (no decimation), sixteen (decimation by four) or sixty-four (decimation by sixteen) frames are selected, and that number of image files is transferred as pixel data into the frame buffer. For decimation by four, every other pixel on every other line is stored for each image. For decimation by sixteen, every fourth pixel from every fourth line is stored for each image. The operator may specify that consecutive files of the sequence are to be viewed, indicating either the first image number at which viewing is to begin in an IN box 60 or the last image number at which viewing is to stop in an OUT box 62. Alternatively the operator may specify that a best fit of the sequence be determined and used when all of the image files won't fit into the frame buffer 22, i.e., there are more than sixty-four image files. When the LOAD button is activated, the images from the workstation 10 are transferred for storage into the frame buffer 22 according to the selected decimation factor.

A device control section of the interface panel 50 includes a frames per image slider 64 that allows the operator to vary the number of times each image frame is consecutively displayed during playback. The slider scale indicates a first range, such as 1-10, but a data entry box 66 allows entry of numbers above the first range up to a predetermined maximum, such as thirty-two. Also included are device control buttons 70-75 for providing the following functions: stop, play forward, play reverse, jog forward, jog reverse and loop respectively. A current image slider 68 is used to select the current image file to be displayed. Also a timecode window 76 indicates the timecode of the current image in terms of hours, minutes, seconds and frames. The data in the IN and OUT boxes 62, 64 are controlled by respective MARK/CLEAR buttons 77, 78. Finally a CLOSE button 80 acts to stop the process and close the panel, while a SAVE button 82 opens another panel so that the decimated sequence may be saved.

Thus the present invention provides a configurable video sequence viewing and recording system that allows an operator to motion check an animation sequence by decimating the image files for storage in a frame buffer and repeating frames when playing the sequence from the frame buffer.

What is claimed is:

1. A method of viewing and recording an animation sequence of graphic images comprising the steps of:  
storing a specified number of graphic images of the animation sequence in a frame buffer according to a selected decimation factor; and  
reading out from the frame buffer selected ones of the graphic images in sequence to provide a full size, real time display of the animation sequence represented by the selected graphic images on a display monitor.
2. A method as recited in claim 1 further comprising the step of specifying a first and a last graphic image of the animation sequence for storing and reading out.
3. A method as recited in claim 1 wherein the storing step comprises the steps of:  
dividing the frame buffer into k image regions, where k is the number of graphic images of the animation sequence that are to be stored in the frame buffer according to the selected decimation factor; and  
for each graphic image transferring every 2<sup>n</sup>th pixel of every 2<sup>n</sup>th line of the graphic image of the animation

5,502,807

**5**

sequence to the corresponding image region of the frame buffer, where  $n$  is the selected decimation factor.

4. A method as recited in claim 3 wherein the reading out step comprises the steps of:

for each graphic image reading out each pixel from the frame buffer  $2^n$  times, and repeating each line  $2^n$  times, for each graphic image in real time to produce a full size image for display; and

repeating the reading out step for each graphic image in <sup>10</sup> the animation sequence to be displayed.

5

**6**

5. A method as recited in claim 1 wherein the storing step comprises the steps of:

determining a best fit for the graphic images of the animation sequence to be stored when the number of graphic images in the animation sequence exceeds the capacity of the frame buffer at the selected decimation factor; and

storing the graphic images satisfying the best fit in sequence in the frame buffer.

\* \* \* \* \*



**United States Patent** [19]  
Naughton et al.

[11] Patent Number: **5,519,825**  
[45] Date of Patent: **May 21, 1996**

[54] METHOD AND APPARATUS FOR NTSC  
DISPLAY OF FULL RANGE ANIMATION

5,411,272 5/1995 Naka et al. .... 273/437  
5,416,497 5/1995 Nishi et al. .... 345/122

[75] Inventors: Patrick J. Naughton, Palo Alto; James  
A. Gosling, Woodside, both of Calif.

0284904 3/1988 European Pat. Off.

0497344 1/1992 European Pat. Off.

[73] Assignee: Sun Microsystems, Inc., Mountain  
View, Calif.

Primary Examiner—Heather R. Herndon

Assistant Examiner—Jody R. Wingard

Attorney, Agent, or Firm—Irell & Manella

[21] Appl. No.: 153,219

**ABSTRACT**

[22] Filed: Nov. 16, 1993

Full-motion animation video is displayed in a computer system through use of sprite objects. The sprite objects define the images on the output display, and the locations of the sprite objects are changed to create the animation. The computer system includes three areas of physical memory assigned the status of a front buffer, a back buffer, and a cache buffer. The front buffer stores a frame currently displayed on the output display. The cache buffer is utilized to store a subset of the sprite objects so that all sprite objects need not be rendered for each frame of animation. The contents of the cache buffer are copied to the back buffer during display of the front buffer. To display a subsequent frame, the front and back buffers are switched. A cache buffer permits display of full-motion animation by minimizing use of processor and computer resources.

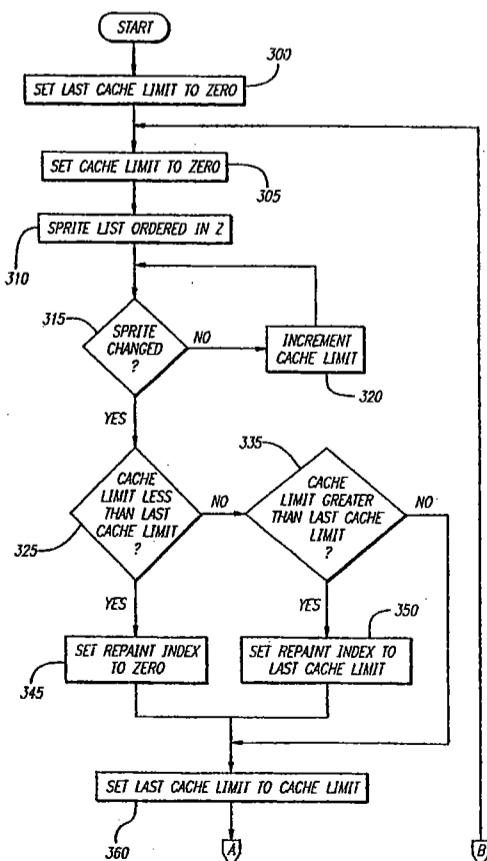
[51] Int. Cl.<sup>6</sup> ..... G06F 13/00  
[52] U.S. Cl. ..... 395/152  
[58] Field of Search ..... 395/152, 153,  
395/154; 434/118; 345/113, 114, 122

[56] References Cited

U.S. PATENT DOCUMENTS

4,414,628	11/1983	Ahuja et al.	395/158
4,951,038	8/1990	Yamamura	345/122
5,001,469	3/1991	Pappas et al.	345/120
5,043,923	8/1991	Joy et al.	395/164
5,111,409	5/1992	Crosby	395/152
5,113,493	5/1992	Gasper et al.	395/152
5,150,312	9/1992	Beitel et al.	395/118
5,164,716	11/1992	Taylor et al.	345/201
5,315,692	5/1994	Hansen et al.	395/119
5,394,523	2/1995	Harris	395/162

10 Claims, 11 Drawing Sheets



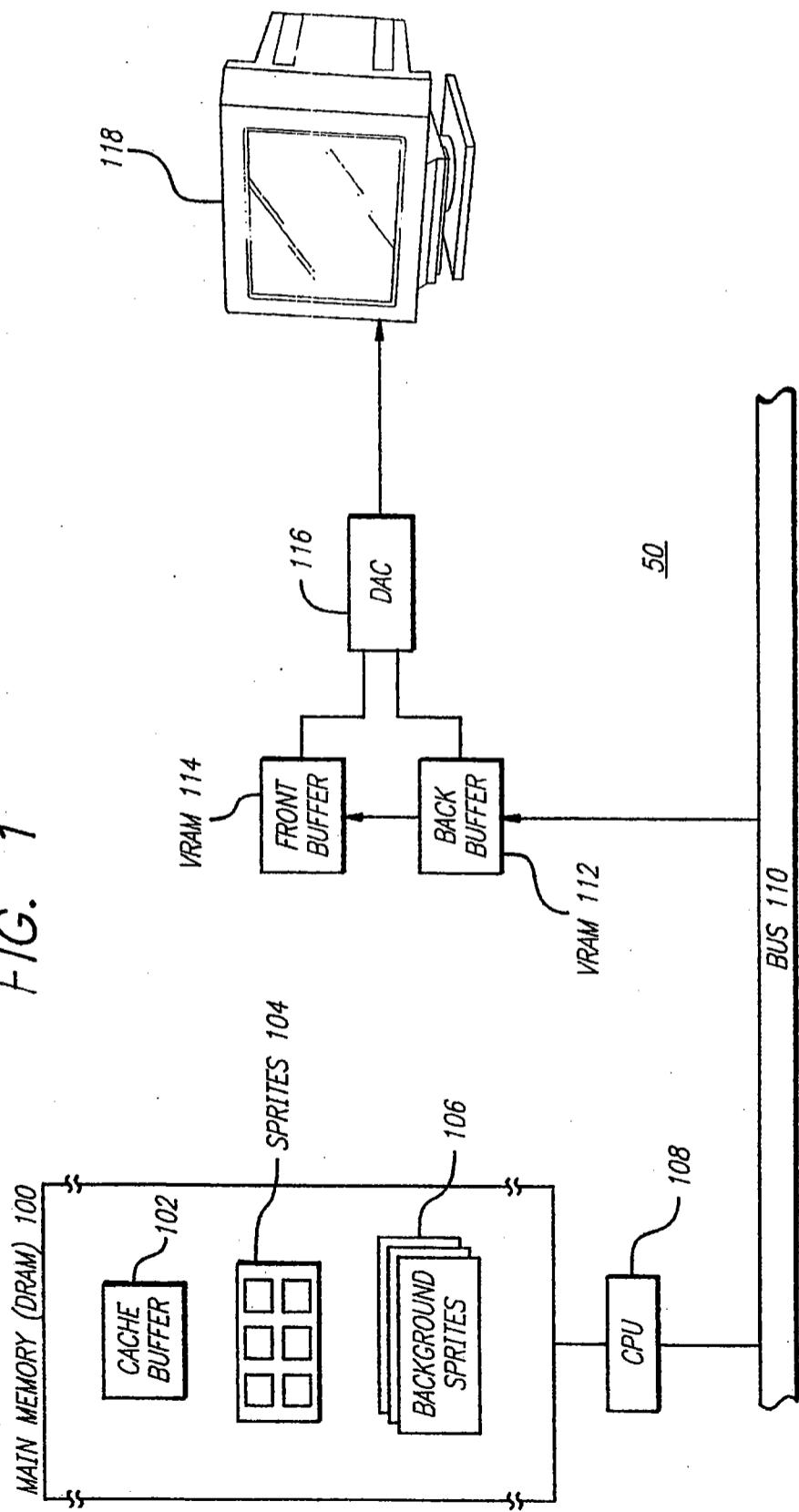
U.S. Patent

May 21, 1996

Sheet 1 of 11

5,519,825

FIG. 1



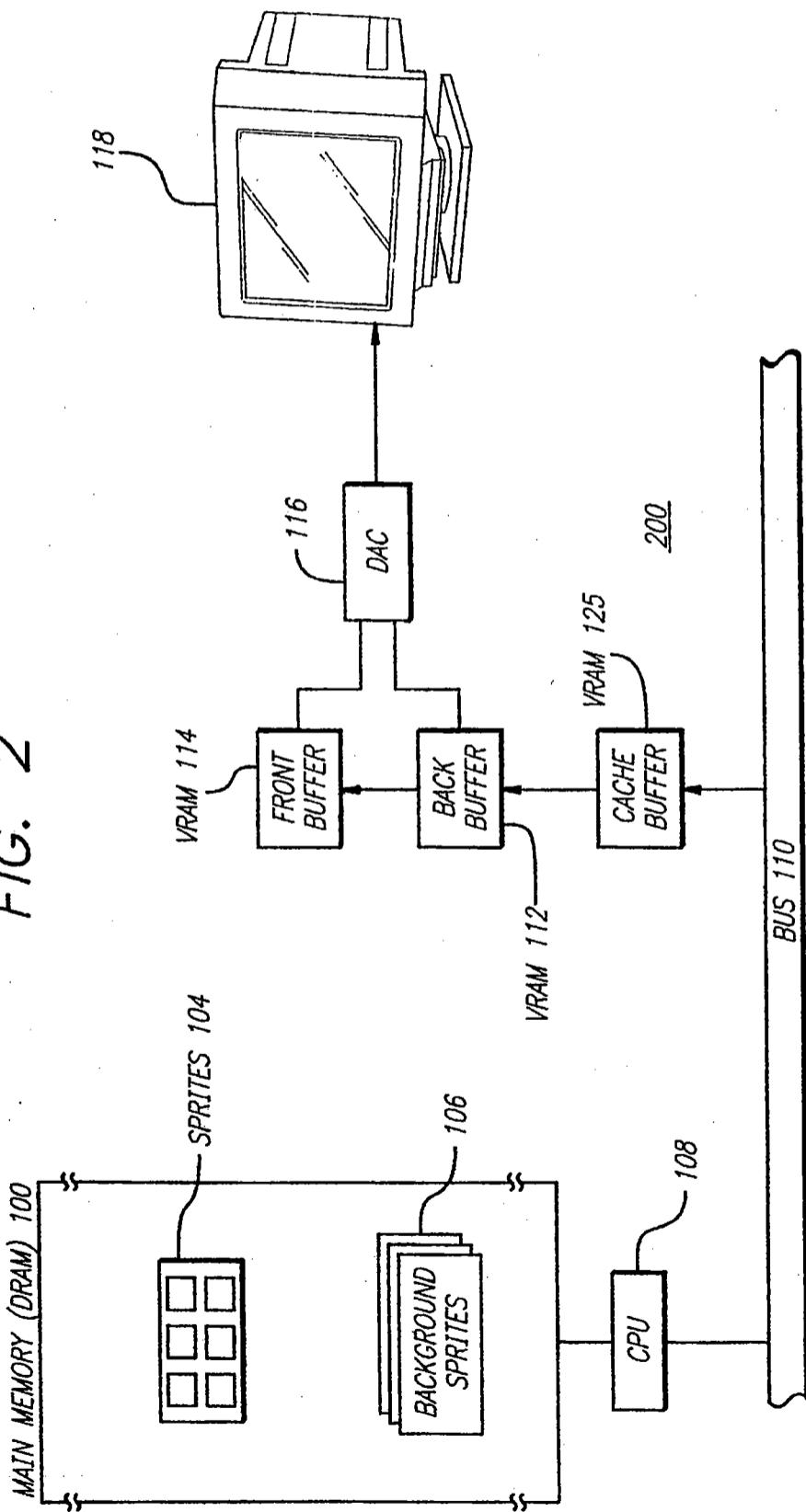
U.S. Patent

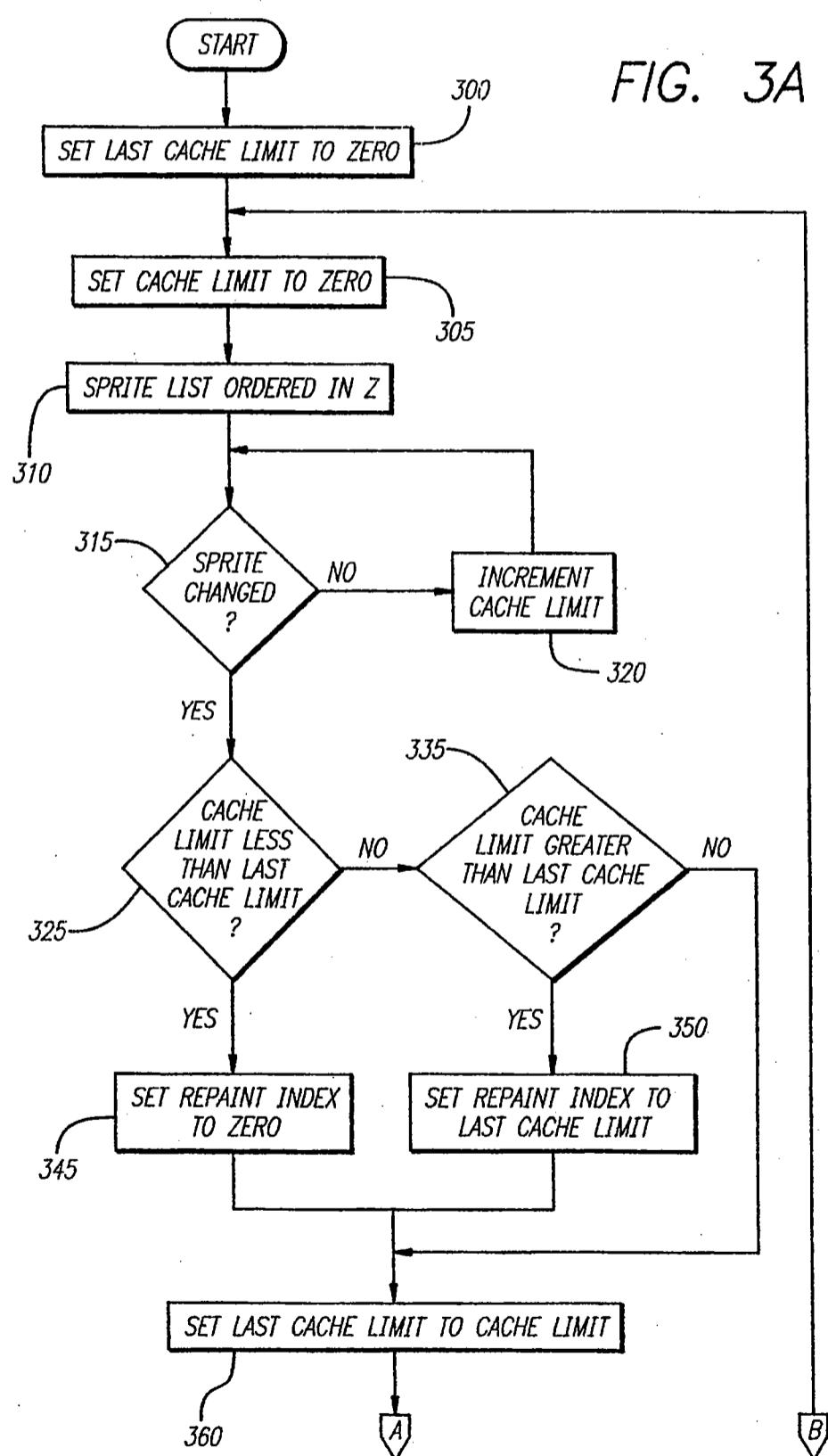
May 21, 1996

Sheet 2 of 11

5,519,825

FIG. 2





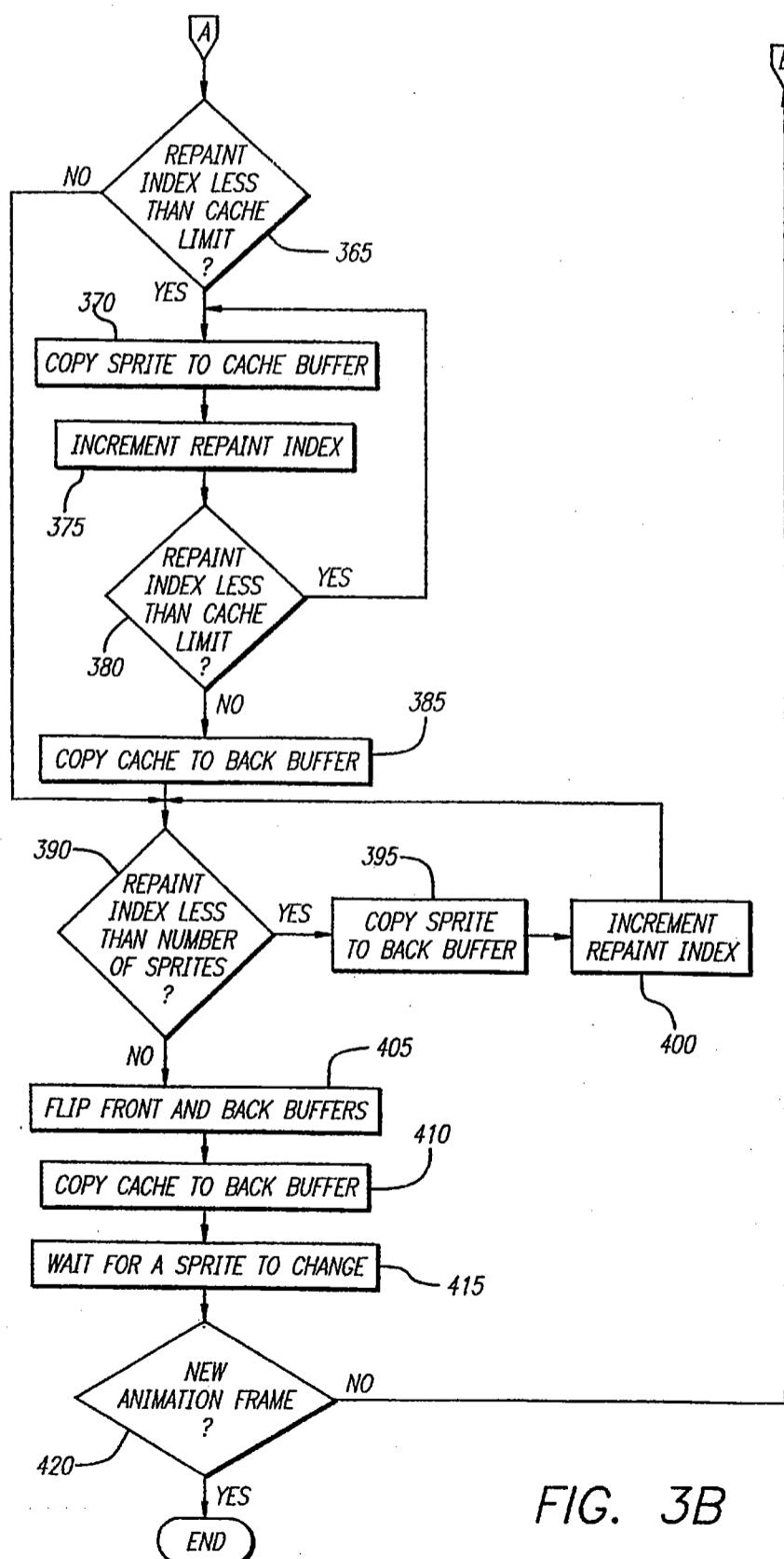


FIG. 3B

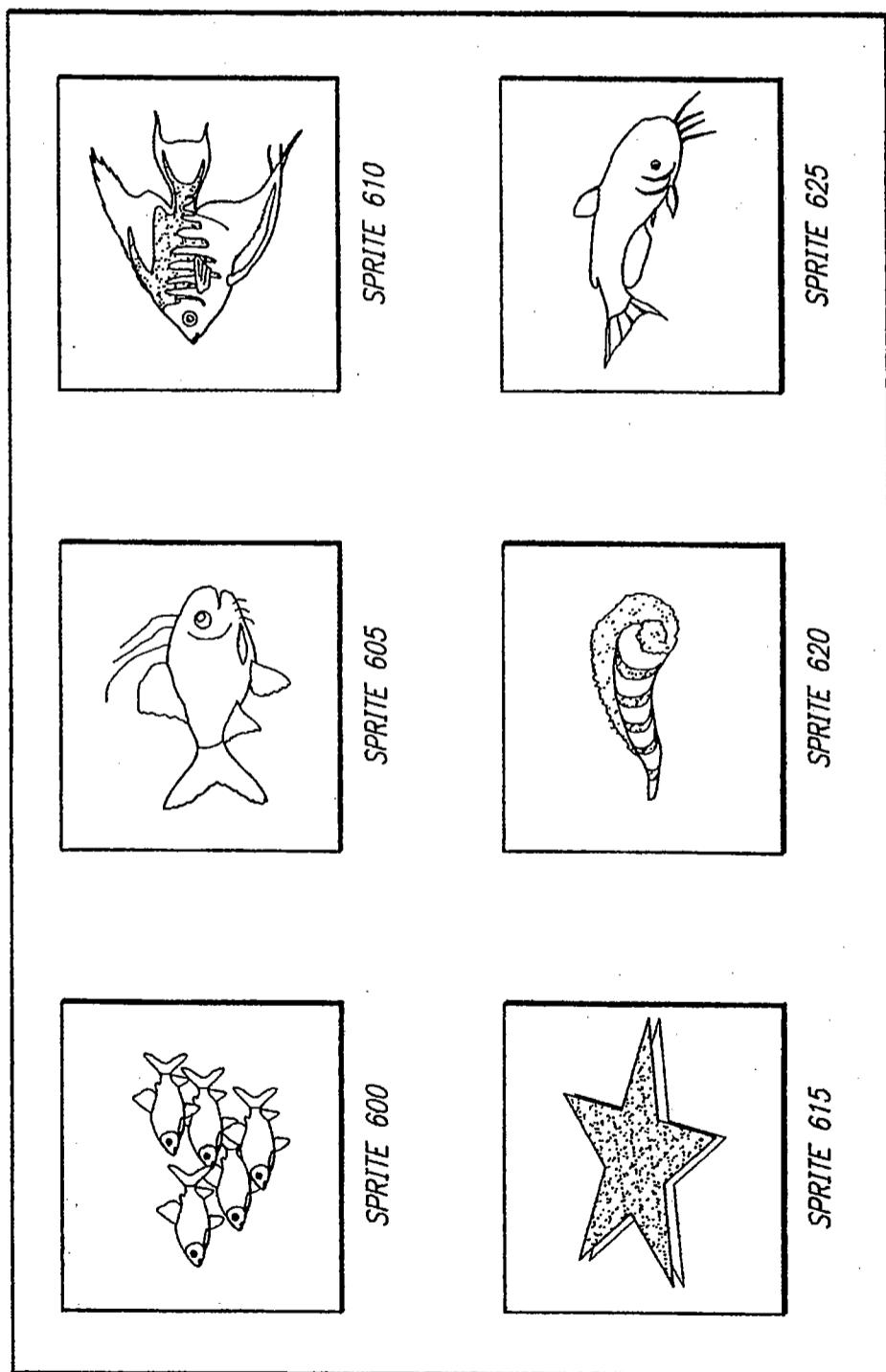
U.S. Patent

May 21, 1996

Sheet 5 of 11

**5,519,825**

*FIG. 4A*

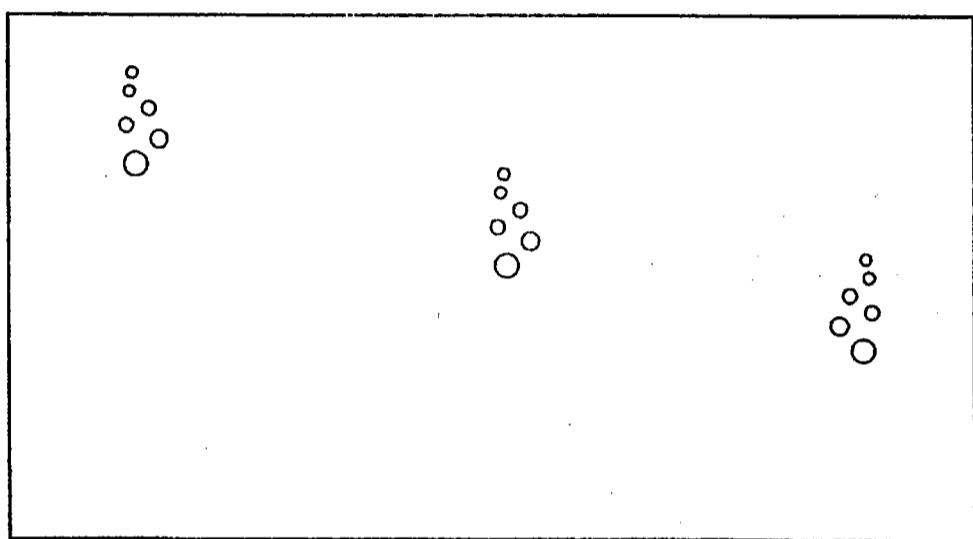


**U.S. Patent**

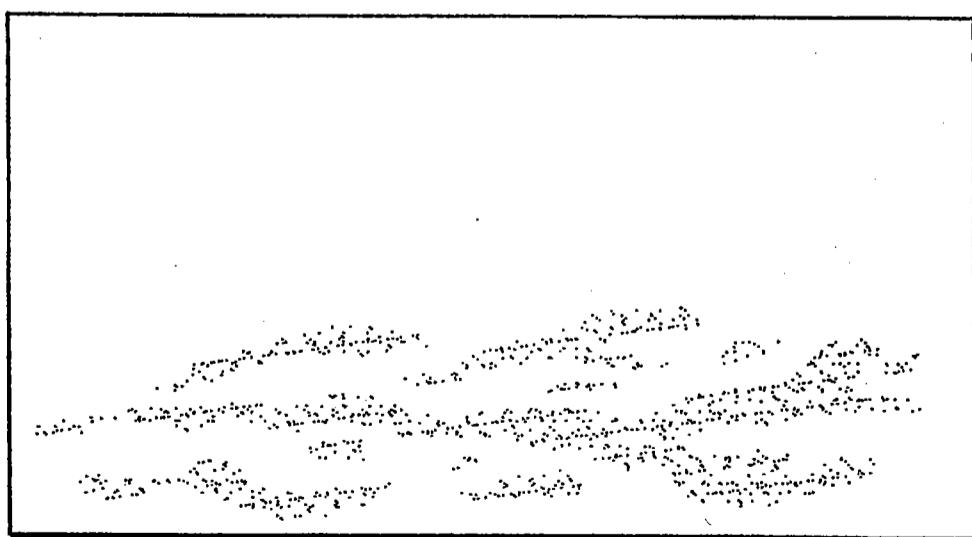
May 21, 1996

Sheet 6 of 11

**5,519,825**



*FIG. 4B*  
630



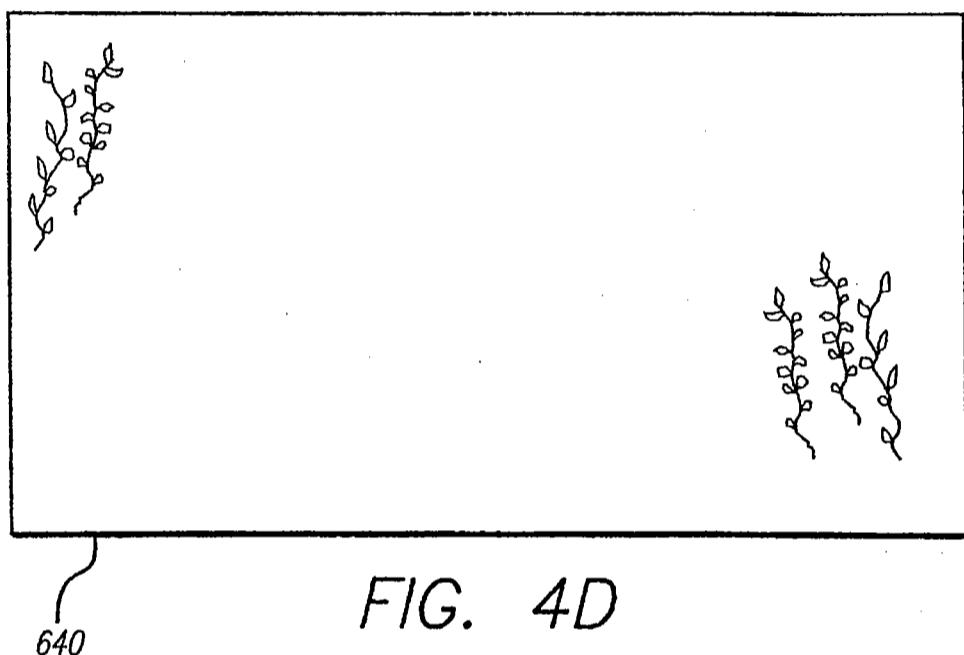
*FIG. 4C*  
635

U.S. Patent

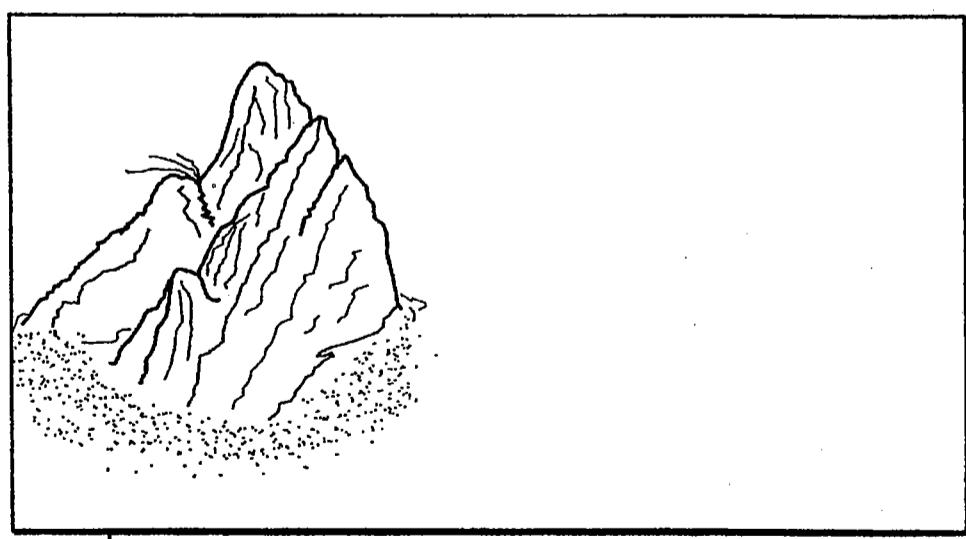
May 21, 1996

Sheet 7 of 11

**5,519,825**



*FIG. 4D*



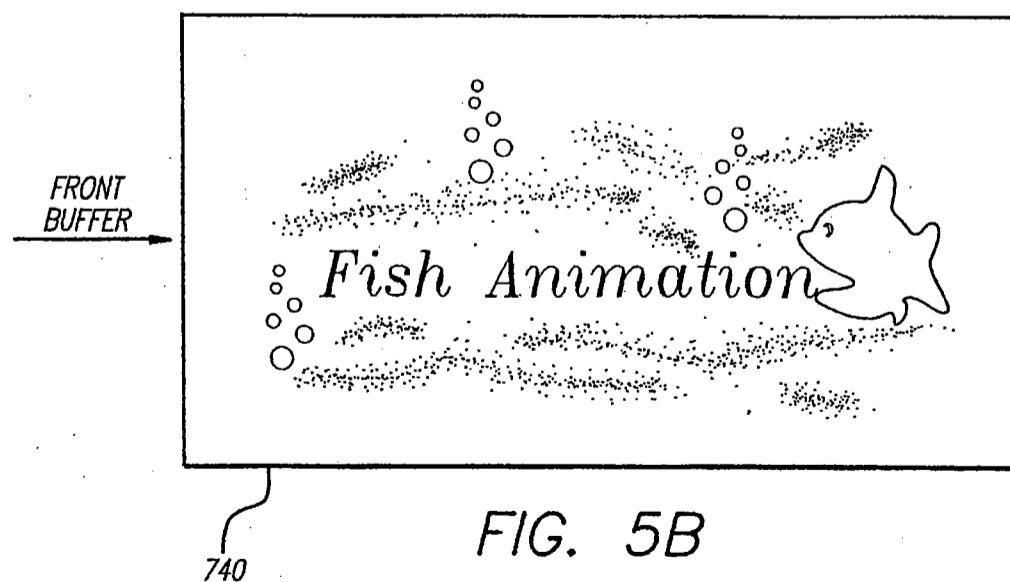
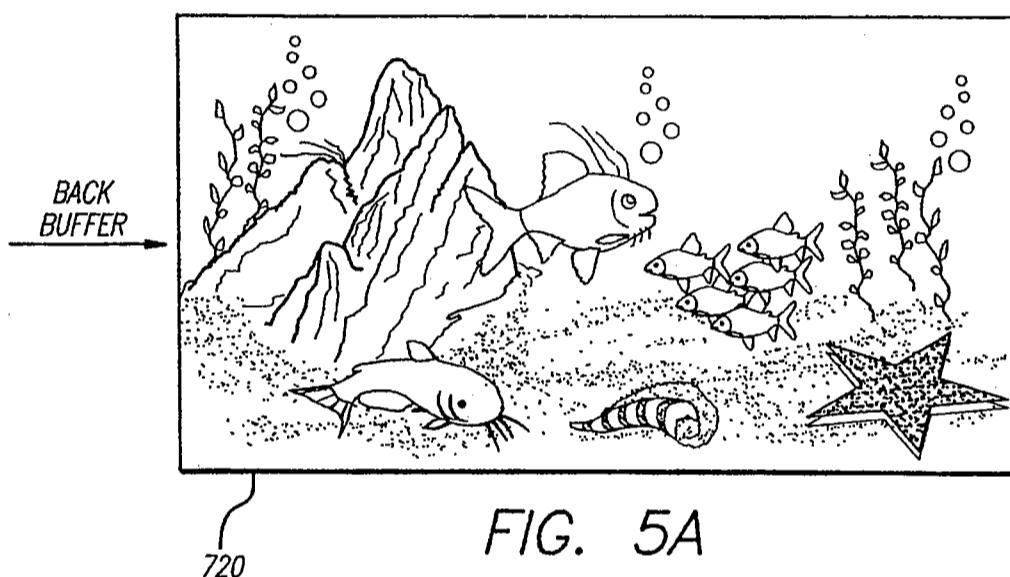
*FIG. 4E*

U.S. Patent

May 21, 1996

Sheet 8 of 11

5,519,825



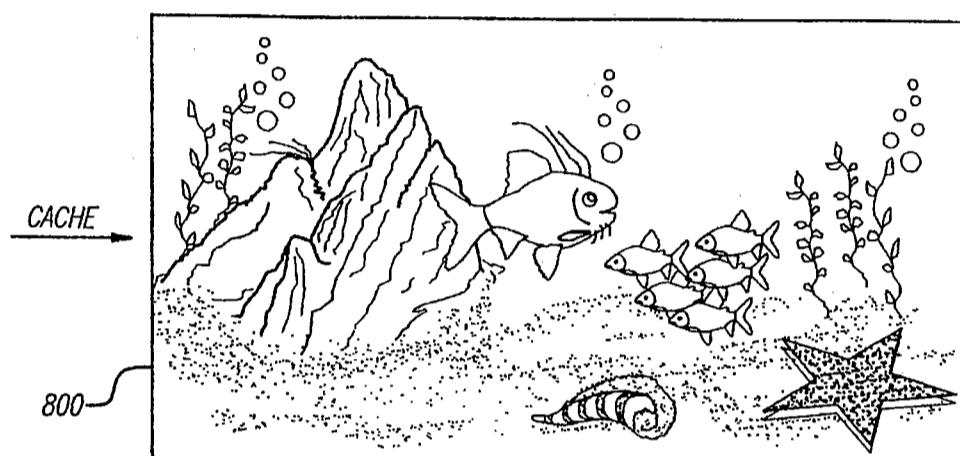


FIG. 6A

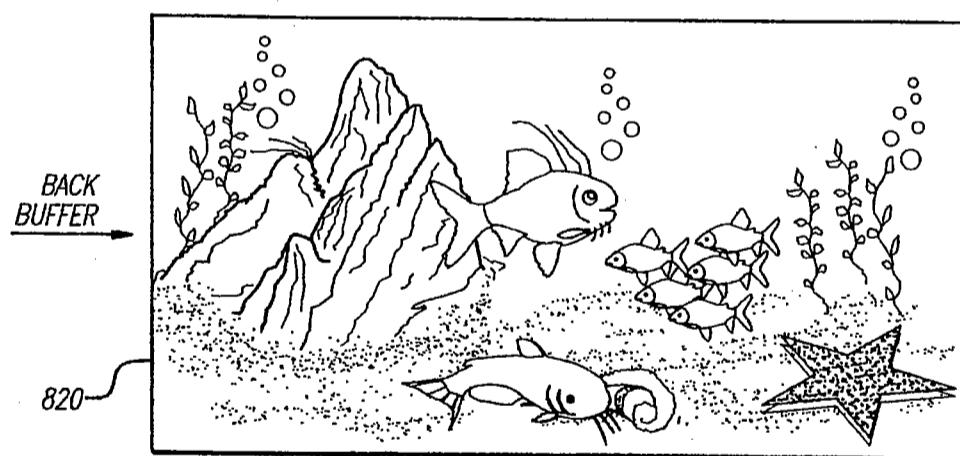


FIG. 6B

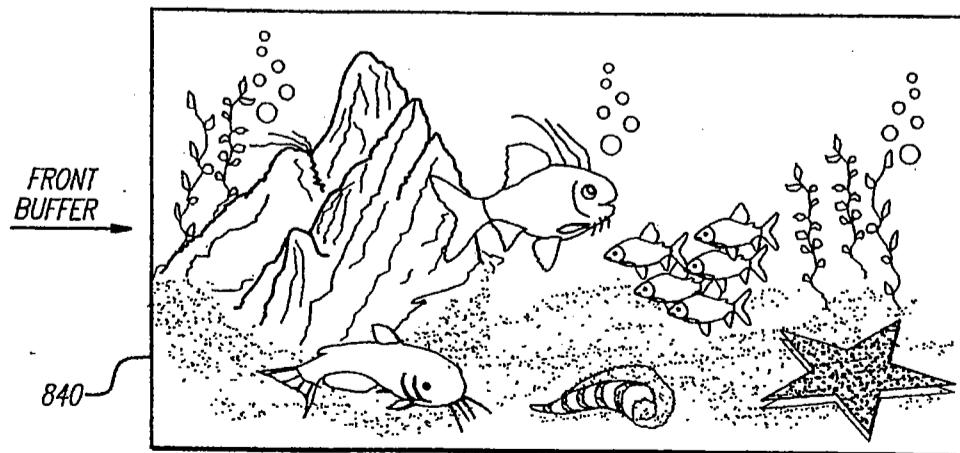


FIG. 6C

U.S. Patent

May 21, 1996

Sheet 10 of 11

5,519,825

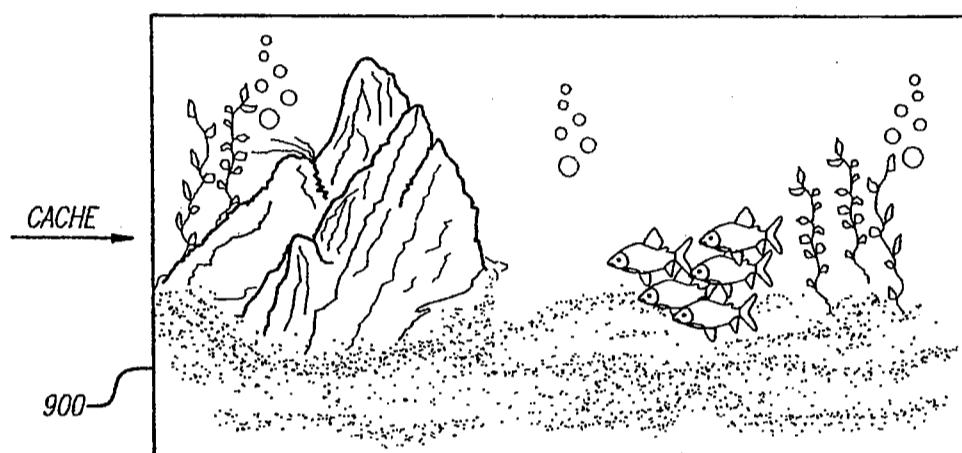


FIG. 7A

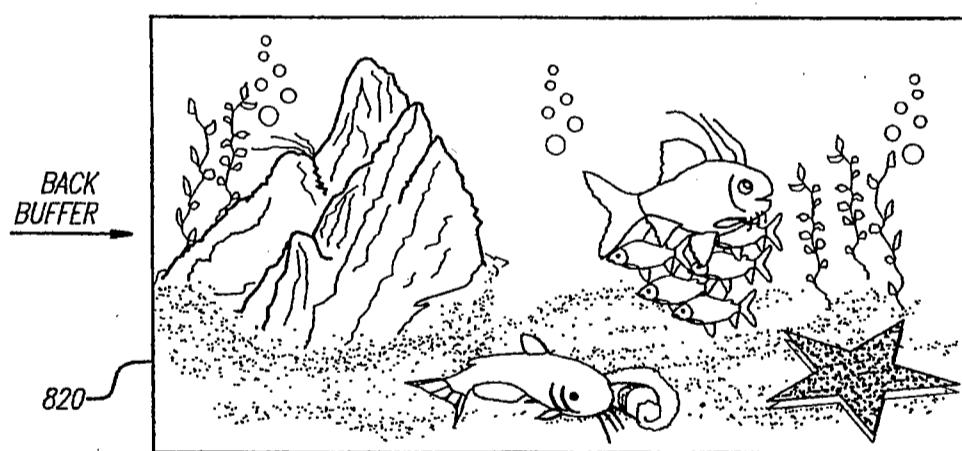


FIG. 7B

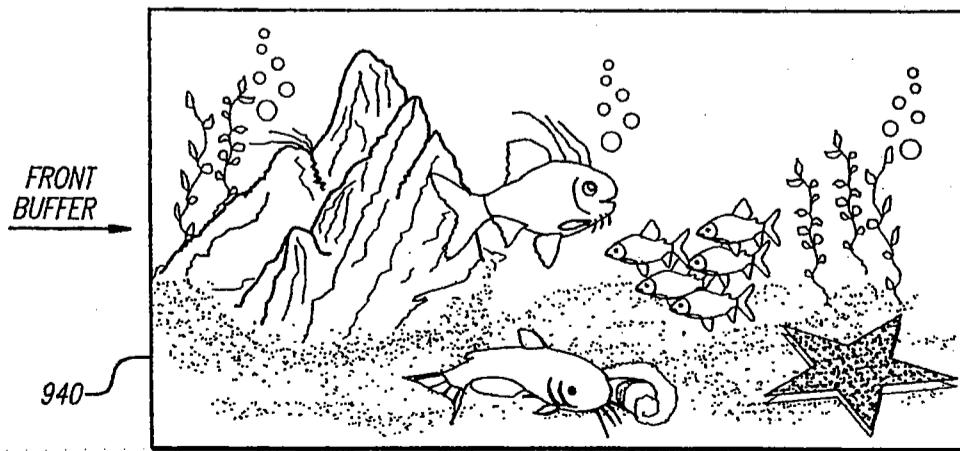


FIG. 7C

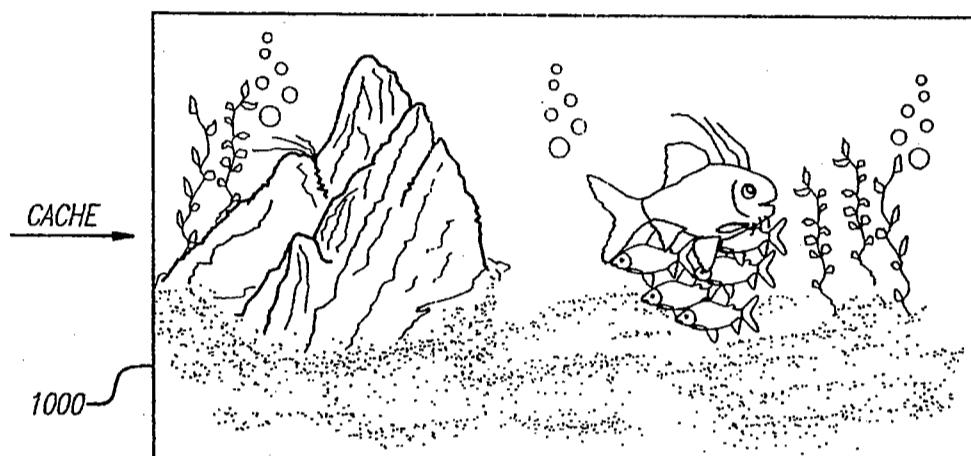


FIG. 8A

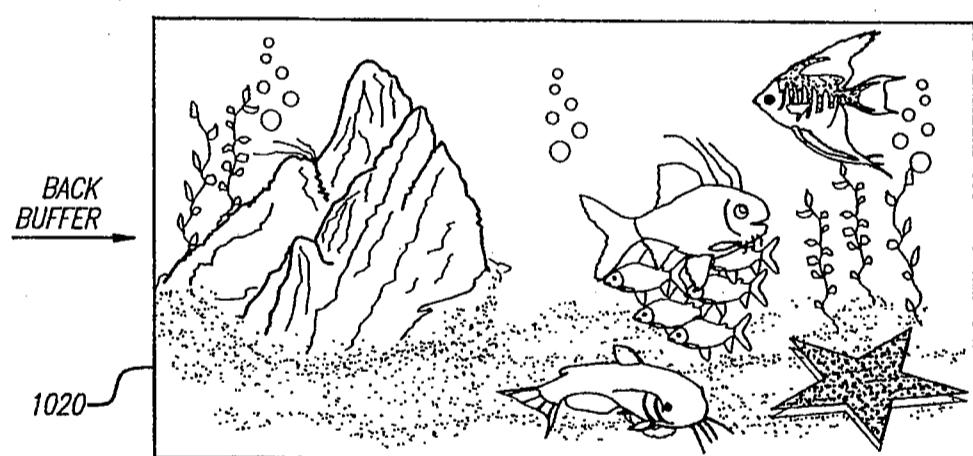


FIG. 8B

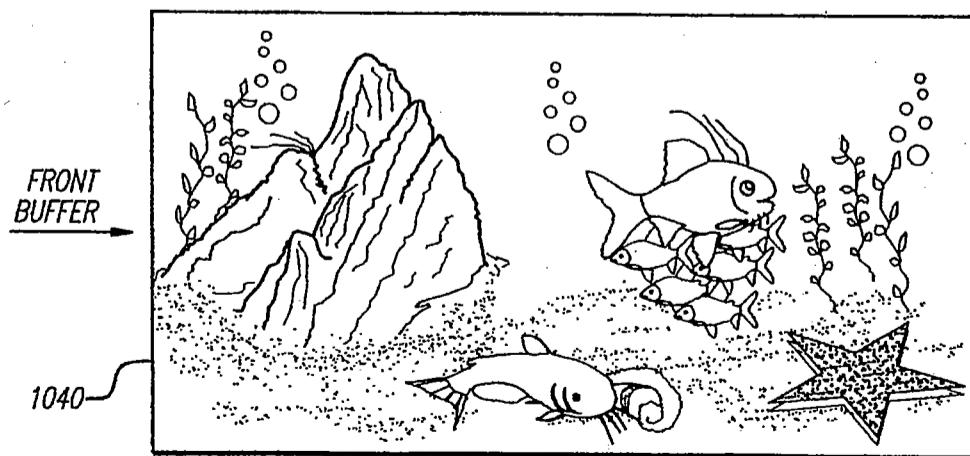


FIG. 8C

## METHOD AND APPARATUS FOR NTSC DISPLAY OF FULL RANGE ANIMATION

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

The present invention relates to data processing in a computer system, and more specifically to methods and apparatus for display of full-motion animation.

#### 2. Art Background

Many computer systems use a region of memory called a frame buffer for storing pixel data for display on a graphics output display device. In order to display the pixel data stored in the frame buffer, a display control system reads the pixel data in the frame buffer line-by-line, converts the data into an analog video signal using a digital to analog converter (DAC), and transmits the analog video signal to the output display device. The line-by-line scanning generally begins at a region in the frame buffer corresponding to the upper left-hand corner of the display screen and continues to the lower right-hand corner.

Typically, a frame buffer is constructed of video random access memory (VRAM) devices. The VRAM devices differ from conventional dynamic random access memory (DRAM) devices because the VRAM devices contain two access ports wherein the DRAM devices typically contain one port. A first access port, called a random access port, provides conventional random access to the VRAM such that a central processing unit (CPU) coupled to the VRAM may read or write to any memory location in the VRAM. A second port, called a serial access port, provides simultaneous serial access to the VRAM such that a device coupled to the serial port can shift data in or out of the VRAM. A display circuit usually accesses the serial port to furnish pixel data to the circuitry controlling the output display. In such a configuration, the CPU can write to the VRAM while a display circuit continually furnishes pixel data to an output display.

Animation sequences are often created in computer systems that couple a display screen to this type of frame buffer based display system. When creating an animation sequence within such a configuration, animation software renders a series of frames in which each frame's image changes slightly. To provide smooth animation, approximately 15 to 30 new frames are displayed each second. As the first frame image changes to the next frame image, the effect of continuous motion is created. Therefore, to create a full-motion animation sequence, the frame buffer is continually updated.

The ability of a frame buffer to both receive pixel data and transfer the pixel data to an output display simultaneously causes certain difficulties. If the animation software writes to the frame buffer memory while the display controller is scanning the image in the frame buffer memory, then the output display may simultaneously display a graphic image from multiple animation frames. The display of improper pixel data from more than one animation frame is referred to as a "frame tear". Frame tears are particularly apparent where motion from one frame to the next causes distortion in the graphic image presented on the display.

To eliminate frame tears, certain computer systems utilize a double buffering display system. The double buffered display system provides two regions of memory in the frame buffer wherein each region of memory stores pixel data to the DAC circuitry. A first region of memory provides a first animation frame to the output display such that the first

region of memory is not updated during scanning for output to the display screen. While the first memory region is displayed on the display screen, animation software renders the next animation frame in the second region of memory.

After the animation software completes the next animation frame, the DAC is switched such that the second region of memory becomes the displayed frame and the first region of memory becomes the "work" region. The animation software renders the next animation frame in the work region of memory. Consequently, frame tears are eliminated in a double buffered display system because pixel data is not written to the region of memory that is currently supplying pixel data to the display screen.

When computer systems utilize a double buffered display system to create animation sequences, the CPU generates every scene in the work region for each new frame of animation in the animation sequence. The animation scenes may comprise both a background scene and animated objects. If the animated objects are being rendered on top of the background scene, the entire background scene must be generated by the CPU before it can render the animated objects. To provide high-quality real time animation, the rendering of the background and the animated objects for an animation frame must be done approximately 15 to 30 times per second.

Full-motion animation on a NTSC-resolution frame buffer requires updating approximately 345,600 pixels per frame based on the dimensions of a full NTSC-resolution frame buffer. Each frame of animation is a single screen, consisting of 345,600 pixels (720x480). Thus, in order to display full-motion animation at 30 frames per second, 10.368 million pixels per second (720x480x30) must be copied to the frame buffer. A modest integer reduced instruction set computer (RISC) CPU executes 10 million instructions per second (MIPS). For such a RISC CPU, approximately 1 instruction is available to paint each pixel (10 MIPS/10m pixels) in order to display full-motion NTSC-resolution animation. Moreover, the calculation of how many instructions are available per pixel does not take into account the multiple layers of image data that may need to be written in order to generate several semi-transparent images together to form the final image for each frame. This type of animation is not possible without special hardware. Therefore, it is desirable to generate full-motion NTSC-resolution animation without the use of specialized computer hardware. The present invention allows for full-motion, NTSC-resolution, 30 frames per second animation without special hardware.

### SUMMARY OF THE INVENTION

The present invention displays full-motion animation video at the rate of 30 frames per second through use of sprite objects and without utilizing special output display hardware. Each sprite object contains a horizontal (X), vertical (Y), and depth (Z) attribute to map the sprite object to a location on the output display. For each frame of an animation sequence, a sprite list, containing a set of sprite objects defining the corresponding animation frame, is generated. To render the animation, the X, Y or Z attributes of the sprite objects in the sprite list are changed.

To implement full-motion animation, a computer system contains a main memory coupled to a central processing unit (CPU). The computer system includes three areas of physical memory arbitrarily and dynamically assigned the status of a front buffer, a back buffer, and a cache buffer. The front

buffer, back buffer, and cache buffer each contain identically sized rasters to match the capacity of the resolution depth color of a corresponding output display. The back and front buffers are coupled to a video digital to analog converter (DAC), and the video DAC is connected to an output display. To display a frame of the full-motion animation video, the CPU programs the DAC to select either the front buffer or the back buffer.

In order to initialize the cache and back buffers, an initial frame for the animation sequence is painted in the cache buffer, and subsequently copied into the back buffer. The front and back buffers are switched to display the first frame. To generate a second frame, the sprite objects in the first frame are compared with the sprite objects on the second frame to determine whether any sprite objects have moved. All sprite objects that have not moved prior to encountering a sprite object that has moved are painted to the cache buffer. A cache limit is set to identify the sprite objects stored in the cache buffer. The cache buffer is then copied to the back buffer, and the back and front buffers are switched to display the second frame.

For each subsequent frame, the sprite objects for the current frame are compared with the sprite objects of the previous frame to determine whether any sprite objects have moved. When a changed sprite object is detected and the contents of the cache buffer are valid, then the cache buffer is copied to the back buffer, and the changed sprite objects are painted into the back buffer. Alternatively, when a changed sprite object is detected and the contents of the cache buffer are not valid, then the cache buffer is repainted and copied to the back buffer. The changed sprite objects are painted into the back buffer, and the front and back buffers are switched.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features, and advantages of the present invention will be apparent from the following detailed description of the preferred embodiment of the invention with references to the following drawings.

FIG. 1 illustrates a block diagram of a computer system configured in accordance with a first embodiment of the present invention.

FIG. 2 illustrates a block diagram of a computer system configured in accordance with a second embodiment of the present invention.

FIGS. 3a-3b illustrate a flow diagram of a method for full-motion animation configured in accordance with the present invention.

FIGS. 4a-4e illustrate graphical depictions of a plurality of sprite objects residing in memory for an animation sequence example configured in accordance with the present invention.

FIGS. 5a-5b illustrate a graphical representation of pixel data for an initial scene of an animation sequence example configured in accordance with the present invention.

FIGS. 6a-6c illustrate a graphical representation of pixel data for a second scene of an animation sequence example configured in accordance with the present invention.

FIGS. 7a-7c illustrate a graphical representation of pixel data for a third scene of an animation sequence example configured in accordance with the present invention.

FIGS. 8a-8c illustrate a graphical representation of pixel data for a fourth scene of an animation sequence example configured in accordance with the present invention.

#### NOTION AND NOMENCLATURE

The detailed descriptions which follow are presented largely in terms of algorithms and symbolic representations of operations within a computer system. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art.

Within the context of this application, and generally, an algorithm is conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Further, the manipulations performed are often referred to in terms, such as adding or comparing, which are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein which form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases, a distinction is maintained between the method operations in operating a computer and the method of computation itself. The present invention relates to method steps for operating a computer in processing electrical or other (e.g., mechanical, chemical) physical signals to generate other desired physical signals.

The present invention also relates to apparatus for performing these operations. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer as selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to a particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these machines will appear from the description given below. Machines which may perform the functions of the present invention include those manufactured by FirstPerson, Inc., as well as other manufacturers of computer systems.

#### DETAILED DESCRIPTION OF THE INVENTION

Methods and apparatus for full-motion real-time animation are disclosed. In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required to practice the present invention. In other instances, well known circuits and devices are shown in block diagram form to avoid obscuring the present invention unnecessarily.

The methods and apparatus of the present invention permit display of real-time full-motion animation. In a preferred embodiment of the present invention, full-motion animation is generated and displayed in the National Television Standard Committee (NTSC) video format. However, the methods and apparatus of the present invention equally apply to animation sequences generated in the red, green, blue (RGB) and monochrome video formats. In addition, the methods and apparatus of the present invention are applicable to animation sequences generated in a high definition television (HDTV) format. However, animation sequences conforming to the NTSC video format permit direct interfacing with existing television formats. As will be described more fully below, the present invention displays full-motion NTSC resolution animated video at the rate of 30 frames per second without utilizing special output display hardware.

The present invention generates sprite objects for use in displaying full-motion animation. In some video game applications, sprites are utilized in hardware based animation techniques. The number of hardware sprites that can be used in hardware animation techniques is limited, and therefore display of animation sequences is also limited. Also, hardware sprites are limited to fixed register sizes such that all sprites contain the same number of pixels. In the present invention, sprite objects are defined as graphical images that are stored in memory and used to generate full-motion animation. The sprite objects of the present invention are not limited to a particular output display size, and each sprite object may comprise a different pixel resolution. For example, a sprite object containing a background image, may comprise a size consisting of an entire screen of pixels.

Each sprite object contains a horizontal (X), vertical (Y), and depth (Z) attribute to map the sprite object to a location on the output display. The X and Y attributes for each sprite object define the horizontal and vertical output display locations, respectively. The Z attribute defines the stacking order or depth dimension for the particular sprite object. For purposes of explanation and convention, sprite objects having a lower Z attribute value are displayed on top of other sprites having a higher Z attribute value. For each frame of an animation sequence, the present invention generates a sprite list. The sprite list contains a set of sprite objects that defines the corresponding animation frame. To generate a typical animation sequence, a number of sprite objects, containing background images, are displayed. In addition to displaying the sprite objects containing the background images, sprite objects defining characters or target objects for the animation sequence are displayed. Typically, the character sprite objects are displayed in the foreground of the animation frame. To generate the animation, the X, Y or Z attributes of the sprite objects are changed. Consequently, the sprite objects move or change location to create the full-motion animation. The use of sprite lists, including ordering of sprite objects based on Z attributes, is described more fully below.

Referring to FIG. 1, a block diagram of a computer system configured in accordance with a first embodiment of the present invention is illustrated. To implement full-motion animation, a computer system 50 contains a main memory 100. Preferably, the main memory 100 consists of dynamic random access memory (DRAM). The main memory 100 is coupled to a central processing unit (CPU) 108 and, in turn, the CPU 108 is coupled to a bus 110. The main memory 100 contains storage locations for general operation of the computer system 50. In addition, the main memory 100 contains storage locations to generate the full-motion animation of the present invention. The main memory 100 contains, in

part, a cache buffer 102, sprite objects 104, and sprite objects containing background sprites 106. The cache buffer, the sprite objects 104, and background sprites 106 may reside at any location within main memory 100. The computer system 50 includes three areas of physical memory arbitrarily and dynamically assigned the status of a front buffer, a back buffer, and a cache buffer. The front buffer, back buffer, and cache buffer each contain identically sized rasters to match the capacity of the resolution depth color of a corresponding output display. The back buffer 112 and front buffer 114 are implemented with video random access memory (VRAM). The back and front buffers 112 and 114 are coupled to a video digital to analog converter (DAC) 116. The video DAC 116 is connected to an output display 118.

The front and back buffers 114 and 112 operate as frame buffers for the computer system 50. The front and back buffers 114 and 112 store a series of scan lines that represent an image for display on output display 118. The video DAC 116 scans either the front buffer 114 or back buffer 112, line by line, converts the pixel scan lines to an analog format, and drives the raster output display 118 to display an associated scan line. The output display 118 may comprise a cathode ray tube (CRT) or a liquid crystal display (LCD) output display. A pixel, or picture element, is a single colored dot illuminated on the output display 118, and is represented in the frame buffer by an arbitrary number. For example, a 24 bit pixel may comprise 8 bits of red, green, and blue. A 16 bit pixel may comprise 5 bits of red, 6 bits of green, and 5 bits of blue. In addition to pixels, the computer system also stores alphas. An alpha is a fixed point fractional number that represents the opacity of each pixel and may be stored in a separate physical memory having a one-to-one correspondence with the associated pixel data. Alternatively, alpha values may be packed into each word in the physical frame buffer that contains the pixel data. Any format for storing pixels can be used in conjunction with the generation of the full-motion animation without deviating from the spirit and scope of the present invention.

To display a frame of the full-motion animation video, the CPU 108 programs the DAC 116 to select either the front buffer 114 or the back buffer 112. The CPU 108 selects the front and back buffers 114 and 112 in an alternative manner. For purposes of explanation, the VRAM currently selected to supply pixel data is entitled the front buffer, and the VRAM not currently selected to supply pixel data is entitled the back buffer. Using this convention, the front buffer alternates from the VRAM 114 and the VRAM 112 for each frame displayed on the output display 118. The operation of the VRAM with a corresponding DAC and output raster display is well-known in the art and will not be described further.

Referring to FIG. 2, a block diagram of a computer system configured in accordance with a second embodiment of the present invention is illustrated. The computer system 200 is configured similar to the computer system 50 illustrated in FIG. 1 except for the arrangement of the cache buffer. In the computer system 200, the cache buffer 125 is implemented with a VRAM. By utilizing a VRAM for the cache buffer as shown in FIG. 2, data transferred from the cache buffer 125 to the back buffer 112 is optimized. An optimized copy operation for the cache buffer 125 to the back buffer VRAM 112 is described more fully below.

Referring to FIG. 3a, a flow diagram illustrating a method for full-motion animation configured in accordance with the present invention is shown. To display full-motion animation video, sprite objects 104 and background sprites 106 are stored in the main memory 100. The sprite objects 104 and

background sprites 106 comprise the source material for generating the full-motion animation. The composite list of sprites for a particular animation scene is entitled a "sprite list". The method for full motion animation of the present invention utilizes a last cache limit, cache limit, and repaint index variables. To initialize the last cache limit variable for an initial scene of full-motion animation, the last cache limit is set to zero as shown in step 300. Also, the cache limit variable is set to zero as shown in step 305. In step 310, the sprite list is ordered based on the Z attributes of the sprite objects. To order the sprite list, the sprite object comprising the largest Z attribute is placed first on the sprite list. The largest Z attribute represents that the sprite object contains the greatest depth dimension. Similarly, all sprite objects are ordered in a list starting with the sprite object having the largest Z attribute and ending with the sprite object having the smallest z attribute. Typically, the sprite objects containing the background sprites reside in the top of the Z ordered sprite list.

After ordering the sprite list, sprite objects for a current scene are compared against sprite objects for a previous scene to determine which sprite objects have changed as shown in block 315. For an initial scene, all sprites objects are marked as unchanged. For subsequent scenes, the ordered sprite list for the previous scene is entitled the "previous sprite list", and the ordered sprite list for the subsequent scene is entitled the "current sprite list". To determine whether a sprite has changed, the X, Y, and Z attributes of the current sprite list are compared against the X, Y and Z attributes of the previous sprite list. If a sprite object contained on the current sprite list does not reside on the previous sprite list, then the sprite object is new, and the sprite object is considered changed. If the sprite object has not changed, then the cache limit is incremented as shown in step 320. For each sprite object that has not changed, the cache limit is incremented as shown in a loop comprising steps 315 and 320. If a sprite object changes, then the cache limit is compared against the last cache limit as shown in step 325. If the cache limit is less than the last cache limit, then the repaint index variable is set to zero as shown in step 345. If the cache limit is greater than the last cache limit, then the repaint index variable is set to equal the last cache limit as shown in steps 335 and 350. Furthermore, if the cache limit is equal to the last cache limit, then the repaint index is not set. As shown in step 360, the last cache limit is set to the value of the cache limit.

Referring to FIG. 3b, a continuation of the flow diagram of FIG. 3a illustrating a method for full-motion animation configured in accordance with the present invention is shown. The repaint index variable is compared with the cache limit variable as shown in step 365. If the repaint index is less than the cache limit, then the sprite object is copied to the cache buffer as shown in step 370. Upon copying the sprite object to the cache buffer, the repaint index is incremented as shown in step 375. In step 380, the repaint index is again compared against the cache limit. If the repaint index is less than the cache limit, the sprite object is copied to the cache buffer, and the repaint index is incremented. The loop, comprising of steps 370, 375, and 380 is executed until the repaint index is equal to the cache limit. When this occurs, the contents of the cache buffer are copied to the back buffer as shown in step 385.

In step 365, if the repaint index is greater than or equal to the cache limit, then the repaint index is compared against the number of sprites contained in the current sprite list as shown in step 390. This comparison is also made after the contents of the cache buffer are copied to the back buffer in

step 385. If the repaint index is less than the number of sprites contained on the current sprite list, the next sprite object on the current sprite list is copied to the back buffer as shown in step 395. In step 400, the repaint index is incremented. Again, the repaint index is compared with the number of sprite objects on the current sprite list, and if the repaint index is less than the number of sprites, the next sprite object on the current sprite list is copied to the back buffer, and the repaint index is incremented. The loop comprising of steps 390, 395, and 400 are executed until the repaint index is equal to the number of sprites. When this occurs, the front and back buffers are reversed so that the current scene is stored in the front buffer for display on the output display as shown in step 405. The contents of the cache buffer are copied to the back buffer as shown in step 410. In step 415, the method waits for a sprite to change, thereby indicating the generation of a new scene for the full-motion animation video. The cache limit variable is set to zero, and the method is executed for a subsequent scene.

The methods and apparatus of the present invention permit display of full-motion animation by taking advantage of the way in which animated is generated. Typically, an animation scene comprises a general background scene. In addition, the animation image contains objects displayed on top of the general background scene. The objects are the characters that move to create the animation. Therefore, to create a typical animation sequence, the objects are moved relative to the background scene. Consequently, for a full-motion animation sequence consisting of a number of frames, the background sprites do not move very often.

The methods and apparatus of the present invention are most effective when conditions occur that do not require painting of a new cache buffer. If the background sprites do not change for several animation frames, then the background sprites stored in the cache buffer remain valid. Therefore, based on this general operation of animation, the present invention stores more elements in the cache buffer that do not typically move. If the sprite objects having large Z attributes for display in the background portion of the frame do not move, then those sprite objects do not require painting to either the cache or back buffers. Instead, the sprite objects stored in the cache buffer are fast copied to the back buffer. In a preferred embodiment, a special VRAM is utilized to facilitate the fast copy between the cache buffer and the back buffer. For a further description of the fast copy between special VRAMs, see, U.S. patent application, Ser. No.08/106,281, filed by Forrest, et al. on Aug. 13, 1993, entitled "Method and Apparatus for Constructing a Frame Buffer with a Fast Copy Means", and assigned to the assignee of the present invention, FirstPerson, Inc., Mountain View, Calif.

The present invention will now be further described by way of an example. The example is an animation program which depicts movement of a number of fish in an under-water environment. Referring to FIG. 4a, a graphical depiction of a plurality of sprite objects residing in memory are illustrated for the fish animation sequence. FIG. 4a illustrates graphical representations of sprite objects stored in memory for purposes of description only, and one skilled in the art will appreciate that a plurality of bits are actually stored in memory for each sprite object. For the fish animation example, six sprite objects, including sprite objects 600, 605, 610, 615, 620, and 625 are stored in memory and used for animation. Each sprite object comprises X, Y and Z coordinates identifying an initial location for display on the output display.

Referring to FIGS. 4b-4e, a plurality of sprite objects containing background scenery images for the fish anima-

tion sequence are illustrated. For the fish animation example, four sprite objects, containing background scenery, including sprite objects 630, 635, 640, and 645 are stored in memory. For the fish animation example, the background sprites merely provide background scenery and do not move. Although the background sprites are stable for the present example, the present invention supports movement of the large sprite objects such as background sprites. The sprite objects containing the background sprites store X, Y and Z coordinates identifying positions for illumination of pixels on the output display. The sprite objects illustrated in FIGS. 4b-4e are ordered according to the Z attribute such that sprite object 630, illustrated in FIG. 4b, contains the largest Z attribute, and sprite object 645, illustrated in FIG. 4e, contains the smallest Z attribute. The sprite object 645 illustrated in FIG. 4e illustrates a shipwreck; the sprite object 640 illustrated in FIG. 4d is sea vegetation background; the sprite object 635 illustrated in FIG. 4c is a sea floor background; the sprite object 630 illustrated in FIG. 4b is a water background.

Referring to FIGS. 7a-7c, a graphical representation of pixel data for an initial scene of the fish animation sequence configured in accordance with the present invention is illustrated. As discussed above in conjunction with the flow diagram of FIG. 3, an initial scene is painted to the cache buffer. In the current example, the sprite object 630 containing the water background is painted first, the sprite object 635 containing the sea floor is painted second, the sprite object 640 containing the sea vegetation is painted third, and the sprite object 645 containing the shipwreck is painted fourth. Similarly, the sprite objects illustrated in FIG. 6a are painted starting with the sprite object comprising the largest Z attribute value.

Referring to Table 1, a sprite list is ordered based on the Z priority for each sprite object for the fish animation example. After the sprite objects 630, 635, 640 and 645 containing background scenes are painted, the sprite object 600 is next on the Z ordered sprite list. Therefore, the sprite object 600 contains the greatest depth dimension for the fish sprite objects. The sprite object 605, being next in the Z ordered sprite list, has a depth dimension less than the sprite object 600. Similarly, sprite objects 615, 620 and 625 are also contained on the sprite list for the initial scene. Because all sprites are considered changed in the initial scene, the sprite objects contained on the initial sprite list are painted to a back buffer 720 in the Z order.

FIG. 5a graphically illustrates the contents of the back buffer 720 after painting the Z ordered sprite list of Table 1. For the initial scene, the cache buffer remains empty. FIG. 5b illustrates a screen logo, initially contained in a front buffer 740, displayed on the output display before initiation of the fish animation sequence. To display the first scene stored in the back buffer 720, the DAC is programmed so as to flip the front buffer 740 and the back buffer 720.

TABLE 1

Sprite List Z Order	Stored in Cache
Sprite 630	
Sprite 635	
Sprite 640	
Sprite 645	
Sprite 600	
Sprite 605	
Sprite 615	
Sprite 620	
Sprite 625	

In order to display a second scene for the fish animation sequence, the sprite list of Table 1 is labeled the previous

sprite list and a current sprite list for the second scene is generated. Referring to Table 2, a Z ordered sprite list for the second scene in the fish animation example is shown. In addition to the Z ordered sprite list for the second scene, Table 2 identifies the sprite objects that moved in the second scene in relationship to the first scene. In the second scene, the sprite object 625 has changed position. As illustrated in the flow diagram of FIG. 3, the current sprite list is ordered based on the Z attributes of the sprite objects. For each sprite object compared, the cache limit is incremented. When sprite object 625 from the current sprite list is compared with the sprite object 625 of the previous sprite list, a determination that the sprite object 625 has moved is made. The background scene sprite objects 630, 635, 640 and 645 did not move, and consequently the sprite objects 630, 635, 640 and 645 are painted to a cache buffer 800. The sprite object 600 has not changed from the previous Z ordered sprite list, and consequently, the sprite object 600 is painted to the cache buffer. Similarly, sprite objects 605, 615 and 620 are also painted to the cache buffer. Based on the comparison, the cache limit, now set to the last cache limit, equals 8. The cache buffer 800 is copied to a back buffer 820, and the sprite object 625 is painted onto the back buffer 820. The back buffer 820 is flipped with a front buffer 840 to display the second scene.

TABLE 2

Sprite List Z Order	Sprite Changed	Stored in Cache
Sprite 630	Not Changed	Sprite 630
Sprite 635	Not Changed	Sprite 635
Sprite 640	Not Changed	Sprite 640
Sprite 645	Not Changed	Sprite 645
Sprite 600	Not Changed	Sprite 600
Sprite 605	Not Changed	Sprite 605
Sprite 615	Not Changed	Sprite 615
Sprite 620	Not Changed	Sprite 620
Sprite 625	Changed	

Referring to FIG. 6a, a graphical depiction of the contents of the cache buffer 800 after generation of the second scene for the fish animation sequence is illustrated. Note that the cache buffer 800 in FIG. 6a does not contain the sprite object 625. Referring to FIG. 6b, a graphical depiction of the contents of the back buffer 820 containing the second scene for the fish animation sequence is illustrated. Note that the sprite object 625 has moved to cover a portion of the sprite object 620. In FIG. 6c, the front buffer 840 containing the initial scene for the fish animation sequence is illustrated. The movement of the sprite object 625 from the first scene to the second scene is larger than generally made in a full-motion animation sequence. However, the larger displacement is made for purposes of illustration and explanation.

In order to generate a third scene for the animation sequence, the sprite list for the second scene is labeled the previous sprite list, and a sprite list for the third scene is generated. The Z ordered sprite list for the third scene is shown in Table 3. For the third scene, the sprite objects 605 and 615 have changed position. To generate the third scene, the cache limit is cleared, and the current sprite list is ordered according to Z attributes. The sprite objects 630, 635, 640, 645 and 600 have not changed position. The sprite objects 630, 635, 640, 645 and 600 are already contained in the cache buffer 800, and therefore the cache limit is incremented for each sprite object. When sprite object 605 in the current sprite list is compared with sprite object 605 in the previous sprite list, it is determined that the sprite

5,519,825

11

object 605 has changed position. The last cache limit, set at 8, is compared with the cache limit which is now set at 5. Because the last cache limit is greater than the value of the cache limit when a sprite object change is encountered, the contents in the cache buffer 800 are no longer valid. Once the contents in the cache buffer 800 are no longer valid, the cache buffer must be refilled. To refill the cache buffer, the background sprite objects 630, 635, 640 and 645 and the fish sprite object 600 are painted to the cache buffer 900. The last cache limit is set to a new value of 5. The new contents of the cache buffer 900 are copied to the back buffer 920. The sprite object 605 is painted to the back buffer 920. Similarly, the sprite objects 615, 620 and 625 are painted to the back buffer.

TABLE 3

Sprite List Z Order	Sprite Changed	Stored in Cache
Sprite 630	Not Changed	Sprite 630
Sprite 635	Not Changed	Sprite 635
Sprite 640	Not Changed	Sprite 640
Sprite 645	Not Changed	Sprite 645
Sprite 600	Not Changed	Sprite 600
Sprite 605	Changed	
Sprite 615	Changed	
Sprite 620	Not Changed	
Sprite 625	Not Changed	

Referring to FIG. 7a, a graphical depiction of the contents of the cache buffer 900 after generation of the third scene is illustrated. The cache buffer 900 contains the background sprite objects 630, 635, 640 and 645, and the fish sprite object 600. In FIG. 7b, a graphical depiction of the contents of the back buffer 920 containing the third scene for the fish animation sequence is illustrated. Note that the back buffer 920 contains sprite objects 605 and 615 in new positions. In FIG. 7c, a graphical depiction of the contents of the front buffer 940 containing the second scene of the fish animation sequence is illustrated. Upon generation of the third scene in the back buffer, the front buffer 940 and the back buffer 920 are flipped so as to display the third scene on the output display. In order to generate a fourth scene for the animation sequence, the sprite list from the third scene is labeled as the previous sprite list, and a sprite list for the fourth scene is generated. Referring to Table 4, a Z ordered sprite list for the fourth scene is shown. In the fourth scene, all sprite objects contained on the previous sprite list are not changed. However, a new sprite object 610 is added. In order to generate the fourth scene, the sprite objects are ordered based on the Z attributes as shown in Table 4. Upon comparison of the sprite objects 630, 635, 640, 645, 600 and 605 from the current sprite list to the previous sprite list, it is determined that sprite objects 630, 635, 640, 645, 600 and 605 have not changed.

Because the sprite objects 630, 635, 640, 645, and 600 are already contained in the cache buffer 900, only the cache limit is incremented for each sprite object. The sprite object 605 has also not changed. However, sprite object 605 is not contained in the cache buffer 900. Therefore, the sprite object 605 is painted into the cache buffer, and the cache limit is incremented. Upon comparing the current sprite list with the previous sprite list, it is determined that the sprite object 610 is a new sprite object. As discussed above, a new sprite object is considered as changed. At this point, the cache limit is set at 6, and the last cache limit is equal to 5. Because the cache limit is greater than the last cache limit, the contents of the cache buffer are copied to the back buffer. Subsequently, the new sprite object 610 and sprite objects 615, 620 and 625 are copied in a back buffer 1020.

12

TABLE 4

Sprite List Z Order	Sprite Changed	Stored in Cache
Sprite 630	Not Changed	Sprite 630
Sprite 635	Not Changed	Sprite 635
Sprite 640	Not Changed	Sprite 640
Sprite 645	Not Changed	Sprite 645
Sprite 600	Not Changed	Sprite 600
Sprite 605	Not Changed	Sprite 605
Sprite 610	Changed	
Sprite 615	Not Changed	
Sprite 620	Not Changed	
Sprite 625	Changed	

Referring to FIG. 8a, a graphical depiction of the contents of the cache buffer after generation of the fourth scene is illustrated. The cache buffer 1000 contains the background sprite objects 630, 635, 640 and 645 and the fish sprite objects 600 and 605. Referring to FIG. 8b, a graphical depiction of the contents of the back buffer 1020 after generation of the fourth scene is illustrated. Note that the back buffer 1020 contains the new sprite object 610, and the sprite object 625 is in a new location. Referring to FIG. 8c, a graphical depiction of the contents of the third scene reside within the front buffer. After generation of the fourth scene of the back buffer, the front and the back buffers are switched such that the contents of the back buffer are displayed on the output display device.

Although the present invention has been described in terms of a preferred embodiment, it will be appreciated that various modifications and alterations might be made by those skilled in the art without departing from the spirit and scope of the invention. The invention should therefore be measured in terms of the claims which follow.

What is claimed is:

1. In a computer system comprising a central processing unit (CPU), memory and an output display, said memory including a back buffer, cache buffer and front buffer, a method for displaying full-motion animation on said output display comprising the steps of:
  - a. generating a sprite list for each scene of said animation comprising a plurality of sprite objects, each of said sprite objects comprising horizontal, vertical, and depth attributes for display on said output display;
  - b. ordering said sprite objects on each sprite list based on said depth attribute;
  - c. comparing said sprite objects in a previously displayed scene with sprite objects in a current scene to determine whether any sprite objects have changed;
  - d. updating said cache buffer to contain all sprite objects that have not changed from said previously displayed scene to said current frame and have a higher depth attribute than any sprite object that has changed, said updating step comprising:
    - adding a sprite object from said current scene to said cache buffer when no sprites having a higher depth attribute have changed and said sprite object is not located in said cache buffer; and
    - removing a sprite object from said cache buffer when a sprite object, stored in said cache buffer, has changed in said current frame by repainting sprite objects to said cache buffer in said current sprite list order up to said sprite object that has changed;
  - e. copying said sprite objects stored in said cache buffer to said back buffer;
  - f. adding sprite objects to said back buffer, in said current sprite list order, when sprite objects identified on said

**13**

- current sprite list are not located in said cache buffer; and
- g. switching said back buffer and said front buffer such that sprite objects for said current frame are displayed on said output display device.
2. The method for full-motion animation as claimed in claim 1 wherein said back buffer and front buffer are implemented with video random access memory (VRAM) and said cache buffer is implemented with dynamic random access memory (DRAM).
3. The method for full-motion animation as claimed in claim 1 wherein said back buffer, said front buffer and said cache buffer are implemented with video random access memory (VRAM).
4. The method for full-motion animation as claimed in claim 1 wherein each frame of said full-motion animation is displayed at 30 frames per second.
5. The method for full-motion animation as claimed in claim 1 wherein said sprite objects are displayed in a National Television Standards Committee (NTSC) video format.
6. An apparatus for displaying full-motion animation on said output display comprising:
- a cache buffer; a back buffer for storing an animation scene prior to display; a front buffer coupled to said output display for storing an animation scene being rendered on said output display;
  - a sprite list generator element for generating a sprite list for each scene of said animation comprising a plurality of sprite objects, each of said sprite objects comprising horizontal, vertical, and depth attributes for display on said output display, sprite list generation means ordering said sprite objects on each sprite list based on said depth attribute;
  - a comparator element for comparing said sprite objects in a previously displayed scene with sprite objects in a current scene to determine whether any sprite objects have changed;
  - a cache buffer updating element coupled to said comparator element and said cache buffer for storing all sprite objects that have not changed from said previously

10

15

25

30

35

40

**14**

- displayed scene to said current scene and have a higher depth attribute than any sprite object that has changed, said cache buffer updating element comprising:
- an adding element for adding a sprite object from said current scene to said cache buffer when no sprites having a higher depth attribute have changed and said sprite object is not located in said cache buffer; and
  - a removal element for removing a sprite object from said cache buffer when a sprite object, stored in said cache buffer, has changed in said current frame by repainting sprite objects to said cache buffer in said current sprite list order up to said sprite object that has changed;
  - a copy element for copying said sprite objects stored in said cache buffer to said back buffer, said copy element adding sprite objects to said back buffer, in said current sprite list order, when sprite objects identified on said current sprite list are not located in said cache buffer; and
  - a switching element coupled to said front and back buffers for switching said back buffer and said front buffer such that sprite objects for said current frame are displayed on said output display device.
7. The apparatus for full-motion animation as claimed in claim 6 wherein said back buffer and front buffer comprise video random access memory (VRAM) and said cache buffer comprises dynamic random access memory (DRAM).
8. The apparatus for full-motion animation as claimed in claim 6 wherein said back buffer, said front buffer and said cache buffer comprise video random access memory (VRAM).
9. The apparatus for full-motion animation as claimed in claim 6 wherein each frame of said full-motion animation is displayed at 30 frames per second.
10. The apparatus for full-motion animation as claimed in claim 6 wherein said sprite objects are displayed in a National Television Standards Committee (NTSC) video format.

\* \* \* \* \*



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 APPLICANT	ATTORNEY DOCKETT NO.
08	252460		

EXAMINER
----------

Huang, P.

ART UNIT	PAPER NUMBER
----------	--------------

2317 7

DATE MAILED:

EXAMINER INTERVIEW SUMMARY RECORD

All participants (applicant, applicant's representative, PTO personnel):

(1) Po Huang (3) Charles Kulas  
(2) Chris Shin (4)

Date of interview 8/13/96

Type:  Telephonic  Personal (copy is given to  applicant  applicant's representative).

Exhibit shown or demonstration conducted:  Yes  No. If yes, brief description:

Agreement  was reached with respect to some or all of the claims in question.  was not reached.

Claims discussed: 1

Identification of prior art discussed: Cooper

Description of the general nature of what was agreed to if an agreement was reached, or any other comments:

Was agree that Cooper does not teach dropping unnecessary frames in the buffer. Applicant has agree to define the sequence of events more clearly to overcome the Cooper reference.

(A fuller description, if necessary, and a copy of the amendments, if available, which the examiner agreed would render the claims allowable must be attached. Also, where no copy of the amendments which would render the claims allowable is available, a summary thereof must be attached.)

1. It is not necessary for applicant to provide a separate record of the substance of the interview.

Unless the paragraph below has been checked to indicate to the contrary, A FORMAL WRITTEN RESPONSE TO THE LAST OFFICE ACTION IS NOT WAIVED AND MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW (e.g., items 1-7 on the reverse side of this form). If a response to the last Office action has already been filed, then applicant is given one month from this interview date to provide a statement of the substance of the interview.

2. Since the examiner's interview summary above (including any attachments) reflects a complete response to each of the objections, rejections and requirements that may be present in the last Office action, and since the claims are now allowable, this completed form is considered to fulfill the

## PATENT

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Case Docket No. CJK-1

In re Application of:  
 Inventor(s) Charles J. Kulas  
 Serial No. 18/252,460  
 Filed Nov. 13, 1996 May 31, 1994  
 Art Unit: 2317  
 Examiner: Huang, Po  
 For: SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED  
 INTERACTIVE PRODUCTIONS  
 ASSISTANT COMMISSIONER FOR PATENTS  
 Washington, D. C. 20231

TRADEMARK OFFICE MAIL ROOM NOV 13 1996

SEARCHED INDEXED  
109 4 Y 1990  
GROUP 2300

Sir:

This is an amendment in the above patent application and includes an attachment which is incorporated by reference. The signature and Certificate of Mailing below serves as a signature and Certificate of Mailing for the attachment(s) in the absence of any signature(s) thereon. The following items are enclosed.

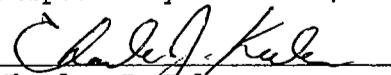
- A Certificate of Mailing.  
 A receipt post card.  
 Amendment or other response to Office Action.  
 Request for Extension of Time.
- [ ] \_\_\_\_\_  
 [ ] \_\_\_\_\_

The fee for this response has been calculated as shown below:

	CLAIMS REMAINING	HIGHEST PAID FOR	NO. EXTRA	RATE (SMALL ENT.)	FEE
TOTAL	10	20	0	x \$11	0
INDEPENDENT	4	5	0	x \$39	0
MULTIPLE DEPENDENT CLAIMS PRESENTED				\$250	0
				TOTAL \$	0

- Enclosed is check number \_\_\_\_ in the amount of the above total.  
 A duplicate of this letter is enclosed.  
 Please date stamp and return the enclosed receipt post card.

Respectfully submitted,

  
 Charles J. Kulas  
 Registration No. 35,809

## CERTIFICATE OF MAILING

I hereby certify that this document and any document referenced herein is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: 11-7-96

19/216 6K2317  
PATENT



THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED

NOV 4 1996

In re application of:  
Inventor(s) Kulas, Charles J.  
Serial No. 13/252,460  
Filed: May 31, 1994  
Art Unit: 2317  
Examiner: Huang, Po  
For: SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED  
INTERACTIVE PRODUCTIONS

GROUP 2300

ASSISTANT COMMISSIONER FOR PATENTS  
Washington, D. C. 20231

REQUEST FOR EXTENSION OF TIME

Sir:

Applicant(s) hereby petitions for an automatic extension of time in the second month to respond to the office action dated June 7, 1996 thereby extending the time for response to November 7, 1996.

Check number 1368 in the amount of \$195.00 is included, made payable to the Commisioner of Patents and Trademarks.

Respectfully submitted,

Charles J. Kulas  
Registration No. 35,809

Please direct correspondence to:  
Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107

3/22/96

CERTIFICATE OF MAILING	
I hereby certify that this document and any document referenced herein is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: <u>11-7-96</u>	
35809 Req. No.	
(Applicant, Assignee or Reg. Representative)	

35809  
11-7-96  
195.00



#9

## Change of Address Notification

In Re Patent Application of:

Inventor(s): Kulas, Charles J.

Re. 35809

Filed: May 31, 1994

NOV 4 9 1996

Serial No.: 08/252,460

GROUP 2000

Art Unit: 2317

Examiner: Huang, Po

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

Assistant Commissioner for Patents

Box OED

Washington, DC 20231

Shirley B. Rasheed:

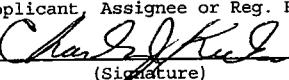
My mailing address has changed. Please send all further correspondence in the above-referenced patent application to the following address, effective immediately:

Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107

Thank You.



Charles J. Kulas  
Reg. No. 35,809

<b>CERTIFICATE OF MAILING</b>	
I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: <u>11-7-96</u>	
Req. No <u>35809</u>	
(Applicant, Assignee or Reg. Representative)	
 (Signature)	



THE UNITED STATES PATENT AND TRADEMARK OFFICE

# 10 /  
PATENT /  
B /  
RN /  
12-3-96

In Re Patent Application of:

Inventor(s): Kulas, Charles J.

Filed: May 31, 1994

RECEIVED

NOV 4 9 1996

Serial No.: 08/252,460

GROUP 2300

Art Unit: 2317

Examiner: Huang, Po

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

Assistant Commissioner for Patents

Washington, DC 20231

AMENDMENT

Sir:

Responsive to the Office Action dated June 7, 1996, please amend the subject application as follows:

IN THE CLAIMS:

Please cancel claims 7, 12 and 13.

Please amend claims 1, 2, 4, 5 and 11 as follows:

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

creating a first animation sequence of digital frames showing a first action;

creating a second animation sequence of digital frames for selective display in place of the first animation sequence, wherein the second animation sequence is a selectable path in the interactive production and shows a second action; 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.

5  
A. (Twice Amended) 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 the ~~CD-ROM~~ further includes identification information associated with data describing the frames, wherein a computer system is used to play back the interactive production, wherein the computer system comprises a processor, user input device, display screen memory including a buffer, and a ~~CD-ROM drive for reading the medium~~, the method further comprising the following steps performed under the control of the processor:

continuously reading the interleaved frames and identification information from the ~~medium CD-ROM~~;

[storing the] using the identification information to store only frames of the first animation sequence into the buffer and to skip frames of the second animation sequence so that the skipped frames of the second animation sequence are not stored in the buffer;

displaying [only] the frames [of the first animation sequence] in the buffer 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 performing the following steps:  
continuously reading the interleaved frames and identification information from the

medium CD-ROM:  
using the identification information to [identify the data in the buffer associated with the frames of the first] store frames of the second animation sequence into the buffer and to skip frames of the first animation sequence so that the skipped frames of the first animation sequence are not stored in the buffer;

[discarding the identified data in the buffer associated with the frames of the first animation sequence;] and

displaying the frames in the buffer [of the second animation sequence] on the display screen to play back the second animation.

*Sub C1* 5. (Twice Amended) 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 showing a first action interleaved with frames corresponding to a second animation sequence showing a second action 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;

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 frames corresponding to the first animation sequence, and, upon generation of the select signal, for displaying on the display screen frames corresponding to the second animation sequence in place of displaying one or more frames corresponding to the first animation sequence.

11. (Amended) 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 CD-ROM further including tags associated with one or more frames on the CD-ROM, wherein the tags indicate which sequence the one or more frames associated with a given tag belongs to, the apparatus comprising:

a computer system including a processor, user input device[, memory including a buffer,] and a display screen;

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

means for reading the tags from the CD-ROM;

means responsive to signals from the user input device to select an animation sequence; and

means for using the tags to buffer and display frames from the selected animation sequence while skipping frames from the other non-selected animation sequence. [determine which of the retrieved frames are stored into the buffer;

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 using the tags to display on the display screen only frames corresponding to the second animation sequence.]

#### REMARKS

This application has been carefully reviewed in view of the Office Action in which claims 1-13 were rejected under 35 U.S.C. § 103 as being unpatentable over Beachy in view of Cooper.

#### **THE 35 U.S.C. § 103 REJECTION**

The present claims include limitations not disclosed in the prior art.

As discussed at an in-person interview with the Examiner, claims 4 and 11 include the limitation of “skipping” frames of an animation sequence on CDROM so that the frames “are not stored in the buffer.” That is, the frames corresponding to a “selected one of the” animation sequences are stored into a buffer for display on the display device. Other frames corresponding to a “non-selected” animation sequence are not stored in the buffer. In other words, the frames are

“dropped.” Support for this limitation may be found in the specification as originally filed beginning at page 19, line 28. By not storing frames from the non-selected sequence the system of the present invention operates more efficiently since the processing time of the central processor (or other processors in the system) is not burdened with transferring data which will not be displayed.

Beachy teaches a method to get more frames into a buffer by using only some of the pixels from each frame. Beachy at col. 4, lines 8-15. Note that Beachy’s frames are from the same sequence (see, e.g., col 3, lines 63-66). Beachy seems to allow for some frames to be omitted from the frame buffer. Col. 4, lines 15-18. However, the frames are still from the same sequence and are omitted under operator control “when all of the image files won’t fit into the frame buffer.”

In contrast, the present invention, as recited by claims 4 and 11, skips frames from a non-selected sequence while storing frames from a selected sequence. The storing and skipping is done regardless of the size of the sequence, i.e., regardless of the number of “image files,” as in Beachy, because the system of the present invention displays the images at a rate to substantially keep up with the rate of images read from the CD-ROM. Beachy teaches away from this approach, and Beachy is incompatible with the present invention, since Beachy proposes a small buffer to hold a limited (sometimes “decimated”) sequence of images for an extended period of time so that an operator can perform a “motion check” of the images. Beachy at col. 4, lines 40-45. Unlike the present invention, Beachy’s approach can not apply to real-time display of images streaming from a CDROM.

In Cooper, when only one sequence is desired to be played back from an interleaved recording both sequences are still read and stored in buffers. See Cooper at col. 8, lines 63-70.

Claims 1 and 5 include the added limitation that “a first animation sequence showing a first action” is interleaved with a second animation sequence showing “a second action different from the first action.” This differs from Cooper which uses two cameras to capture frames of a single action for motion analysis. See, e.g., Cooper’s Fig. 1 showing cameras A and B pointing at a single scene 16. Cooper interleaves the frames from the two cameras when it is desired to study alternating camera angles on each frame to assist in the motion analysis.

Thus, Cooper’s application is a situation where a single motion is studied from two camera angles. Cooper does not disclose nor make obvious the present invention because there is

no way that interleaving frames from two different actions can help study the motion of either of the actions. In fact, such interleaving would greatly hinder the viewing and motion analysis of either of the actions.

In view of this communication, all claims are now believed to be in condition for allowance and such is respectfully requested at an early date. The Examiner is invited to contact the undersigned attorney at 408-955-5485, if necessary.

Respectfully submitted,

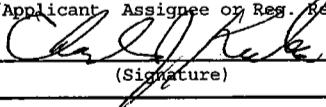


Charles J. Kulas

Registration No. 35,809

Dated: November 6, 1996

Please Send Correspondence to:  
Charles J. Kulas  
244 Texas St.  
San Francisco, CA 94107  
Phone: (408)955-5485  
Fax: (408)955-5490

<b>CERTIFICATE OF MAILING</b>	
I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: <u>11-7-96</u>	
Reg. No. <u>35809</u>	
(Applicant, Assignee or Reg. Representative)	
 (Signature)	



**UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office**

Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231

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

B3M170121

CHARLES J. KULAS  
244 TEXAS STREET  
SAN FRANCISCO CA 94107

EXAMINER

HUANG, P

ART UNIT PAPER NUMBER

2317

11

DATE MAILED: 01/21/97

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

**OFFICE ACTION SUMMARY**

- Responsive to communication(s) filed on Am dt B filed 11/13/96.
- This action is FINAL.
- Since this application is 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 D.C. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire 3 month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

**Disposition of Claims**

- Claim(s) 1-6, 8-11 is/are pending in the application.
- Of the above, claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- Claim(s) \_\_\_\_\_ is/are allowed.
- Claim(s) 1-6, 8-11 is/are rejected.
- Claim(s) \_\_\_\_\_ is/are objected to.
- Claims \_\_\_\_\_ are subject to restriction or election requirement.

**Application Papers**

- See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.
- The drawing(s) filed on \_\_\_\_\_ is/are objected to by the Examiner.
- The proposed drawing correction, filed on \_\_\_\_\_ is  approved  disapproved.
- The specification is objected to by the Examiner.
- The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. § 119**

- Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).
- All  Some\*  None of the CERTIFIED copies of the priority documents have been received.
- received in Application No. (Series Code/Serial Number) \_\_\_\_\_.
- received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

\*Certified copies not received: \_\_\_\_\_

- Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

**Attachment(s)**

- Notice of Reference Cited, PTO-892
- Information Disclosure Statement(s), PTO-1449, Paper No(s). \_\_\_\_\_
- Interview Summary, PTO-413
- Notice of Draftsperson's Patent Drawing Review, PTO-948
- Notice of Informal Patent Application, PTO-152

Serial Number: 08/252,460

Page 2

Art Unit: 2317

1. Applicant's amendment B filed on 11/13/96 in response to examiner's Office Action has been reviewed.
2. Claims 7, 12, and 13 have been canceled. The following rejections now apply.
3. In amendment B filed on 11/13/96, applicant states "Please amend claims 1, 2, 4, 5, and 11 as follows:". However, examiner has found no amendment for claim 2 and has assumed that no amendment was made to claim 2.
4. Claims 1 - 6 and 8 - 11 are presented for examination.
5. The text of those sections of Title 35, U.S. Code § 103 not included in this action can be found in a prior Office Action.
6. Claims 1 - 3, 5, 6, 8, and 9 are rejected under 35 U.S.C. § 103 as being unpatentable over Rodesch, patent no. 4,475,132.
7. As to claim 1, Rodesch teaches the invention substantially as claimed, including a method comprising:

Serial Number: 08/252,460

Page 3

Art Unit: 2317

creating a first animation sequence of digital frames showing a first action [A1 - A7 of fig. 4];

creating a second animation sequence of digital frames for selectively display in place of the first animation sequence [F1 - F8 of fig. 4]; and

writing the first and second animation sequences of frames by interleaving the frames of the first animation sequence with the frames of the second animation sequence [col. 5 line 59 - col. 6 line 46].

Rodesch does not explicitly teach the interactive production being recorded in a CD-ROM. However, one of ordinary skill in the art would have recognized that CD-ROM is well known in the art.

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the CD-ROM into the teaching of Rodesch because writing to a CD-ROM would provide Rodesch's teaching with more flexibility by allowing it to be used in a multi-media system.

8. As to claim 2, Rodesch teaches displaying only the frames of the first animation sequence [col. 6 line 60 - col. 7 line 16],

Art Unit: 2317

accepting signals from the user device selecting the second animation sequence and in response to the signal displaying only the frames of the second animation [col. 7 lines 21 - 64].

9. As to claim 3, compressing and decompressing is well known in the art.

10. As to claims 5 and 6, it is the apparatus claims of method claims 1 - 3, therefore, it is rejected under the same rational.

11. As to claim 8, writing and using a tag is well known in the art.

12. As to claim 9, using the tag to prevent loading of the associated data is a matter of design choice.

13. Claim 4, 10, 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rodesch, patent no. 4,475,132 as applied to claim 1 above, and further in view of Cooper, patent no. 4,789,894.

Art Unit: 2317

14. Cooper, patent no. 4,789,894 was cited as prior art in paper no. 6.

15. As to claim 4, Rodesch teaches interleaving of animation sequences as applied to claim 1 above. Cooper teach a method comprising:

storing frames into a buffer [col. 8 lines 38 - 42];  
displaying only the frames of the first animation sequence [col. 8 lines 63 - 68]; and  
in response to the signals, using the identification information to store frames of the second animation sequence in a buffer and display the second animation sequence [col. 8 line 35 - 70].

Cooper teaches storing the first sequence and second sequence in two different buffer. Since they are stored in different buffer and since frames from one buffer is display, it is obvious that the other frames not being display is skipped.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Rodesch and Cooper because Cooper's teaching would provide

Art Unit: 2317

flexibility to Rodesch's system by allowing buffer to be used to display sequences more efficiently.

16. As to claim 10, writing and using a tag is well known in the art.

17. As to claim 11, it is similar to claim 4, therefore, it is rejected under the same rational.

18. Applicant's arguments with respect to claims 1 - 6 and 8 - 11 have been considered but are deemed to be moot in view of the new grounds of rejection.

19. Applicant's amendment necessitated the new grounds of rejection. Accordingly, **THIS ACTION IS MADE FINAL**. See M.P.E.P. § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 C.F.R. § 1.136(a).

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS FINAL ACTION IS SET TO EXPIRE THREE MONTHS FROM THE DATE OF THIS ACTION. IN THE EVENT A FIRST RESPONSE IS FILED WITHIN TWO MONTHS OF THE MAILING DATE OF THIS FINAL ACTION AND THE ADVISORY ACTION IS NOT MAILED UNTIL AFTER THE END OF THE THREE-MONTH SHORTENED STATUTORY PERIOD, THEN THE SHORTENED STATUTORY PERIOD WILL EXPIRE ON THE DATE THE ADVISORY ACTION IS MAILED, AND ANY EXTENSION FEE PURSUANT TO 37 C.F.R. § 1.136(a) WILL BE CALCULATED FROM THE

Serial Number: 08/252,460

Page 7

Art Unit: 2317

MAILING DATE OF THE ADVISORY ACTION. IN NO EVENT WILL THE STATUTORY PERIOD FOR RESPONSE EXPIRE LATER THAN SIX MONTHS FROM THE DATE OF THIS FINAL ACTION.

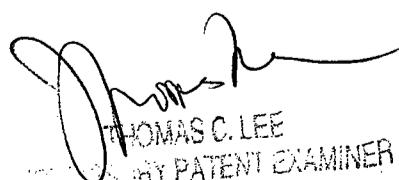
20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Po Huang, whose telephone number is (703) 308-5230. The examiner can normally be reached Monday through Friday from 7:30 AM to 5:00 PM.

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) 308-5359.

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

January 15, 1997

  
THOMAS C. LEE  
PRIMARY PATENT EXAMINER  
230

<b><i>Notice of References Cited</i></b>			Application No.	Applicant(s)		
			08/252,460	Charles J. Kulas		
			Examiner	Group Art Unit		
			Po Huang	2317	Page 1 of 1	
<b>U.S. PATENT DOCUMENTS</b>						
	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	
A	RE. 33,662	08/13/91	Blair et al.	463	3	
B	4,475,132	10/02/84	Rodesch	386	92	
C	4,847,690	07/11/89	Perkins	348	483	
D	5,113,493	05/12/92	Crosby	395	173	
E	5,339,413	08/16/94	Koval et al.	395	680	
F	5,434,678	07/18/95	Abecassis	386	52	
G	5,462,275	10/31/95	Lowe et al.	463	4	
H						
I						
J						
K						
L						
M						
<b>FOREIGN PATENT DOCUMENTS</b>						
	DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
N						
O						
P						
Q						
R						
S						
T						
<b>NON-PATENT DOCUMENTS</b>						
	DOCUMENT (Including Author, Title, Source, and Pertinent Pages)				DATE	
U						
V						
W						
X						

**United States Patent [19]**

Blair et al.

111 E

[45] Reissued Date of Patent: Aug. 13, 1991

[45] Reissued Date of Patent: Aug. 13, 1991

- [54] **TV ANIMATION INTERACTIVELY  
CONTROLLED BY THE VIEWER**  
[76] Inventors: Preston E. Blair, 26212 Mesa Dr.,  
Carmel, Calif. 93923; Frank S.  
Preston, 3003 Larkspur Run,  
Williamsburg, Va. 23185  
[21] Appl. No.: 410,651  
[22] Filed: Sep. 21, 1989  
(Under 37 CFR 1.47)

#### **Related U.S. Patent Documents**

Reissue of:

- Reissue Cr.  
[64] Patent No.: 4,695,953  
Issued: Sep. 22, 1987  
Appl. No.: 831,170  
Filed: Apr. 14, 1986

**U.S. Applications:**

- [63] Continuation of Ser. No. 526,464, Aug. 25, 1983,  
abandoned.

[51] Int. Cl.<sup>5</sup> ..... G06F 15/44; G11B 31/00;  
A63F 9/22

[52] U.S. Cl. ..... 364/410; 273/85 G;  
273/316; 273/DIG. 28; 352/87

[58] Field of Search ... 364/200 MS File, 900 MS File,  
364/410, 411, 521; 434/307, 308, 309, 323;  
273/316, 333, 1 E, 85 G, DIG. 28, 185 A, 185  
E, 220/607, 618, 620, 312/55, 87, 240/220

[56] References Cited

U.S. PATENT DOCUMENTS

- |           |         |                |               |
|-----------|---------|----------------|---------------|
| 3,706,850 | 12/1972 | Fisher et al.  | 178/18        |
| 4,060,915 | 12/1977 | Conway         | 35/9 A        |
| 4,101,132 | 7/1978  | Conrey et al.  | 273/DIG. 28 X |
| 4,185,825 | 1/1980  | Bromley        | 273/DIG. 28 X |
| 4,210,329 | 7/1980  | Steiger et al. | 273/DIG. 28 X |
| 4,285,523 | 8/1981  | Lemelson       | 278/DIG. 28 X |
| 4,305,131 | 12/1981 | Best           | 364/410 X     |
| 4,333,152 | 6/1982  | Best           | 364/521       |
| 4,343,469 | 8/1982  | Kunita et al.  | 273/185 A     |
| 4,359,223 | 11/1982 | Baer et al.    | 273/DIG. 28 X |
| 4,360,345 | 11/1982 | Hon            | 434/307 X     |
| 4,395,045 | 7/1983  | Baer           | 273/DIG. 28 X |
| 4,422,105 | 12/1983 | Rodesch et al. | 364/521 X     |
| 4,449,198 | 5/1984  | Kroon          | 364/900       |
| 4,490,810 | 12/1984 | Hon            | 364/900       |
| 4,518,361 | 5/1985  | Conway         | 434/307       |
| 4,571,640 | 2/1986  | Baer           | 273/1 E X     |
| 4,580,782 | 4/1986  | Ochi           | 273/85 G X    |

#### FOREIGN PATENT DOCUMENTS

8403722 8/1984 RGT Int'l Appl 361/410

#### OTHER PUBLICATIONS

- McGuigan, C. et al., "Mini-Movies Make the Scene", *Newsweek*, Aug. 8, 1983, 79.

Solomon, C., "Fantasy, technology meets in Dragon's Lair", *Los Angeles Times*, Aug. 9, 1983, 1,5.

Rifkin, I., "Video industry draws new life from animation", *Daily News*, Aug. 11, 1983.

"Videodisc games to hit the arcades this summer", *Sof-talk*, Aug. 1983, 268,273.

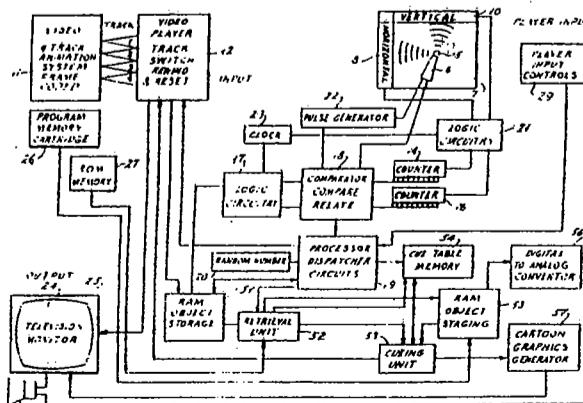
Wood, D. B., "Interactive Laser Game Ignites Enthusiasm-and Controversy", *Christian Science Monitor*, Feb. 17, 1987.

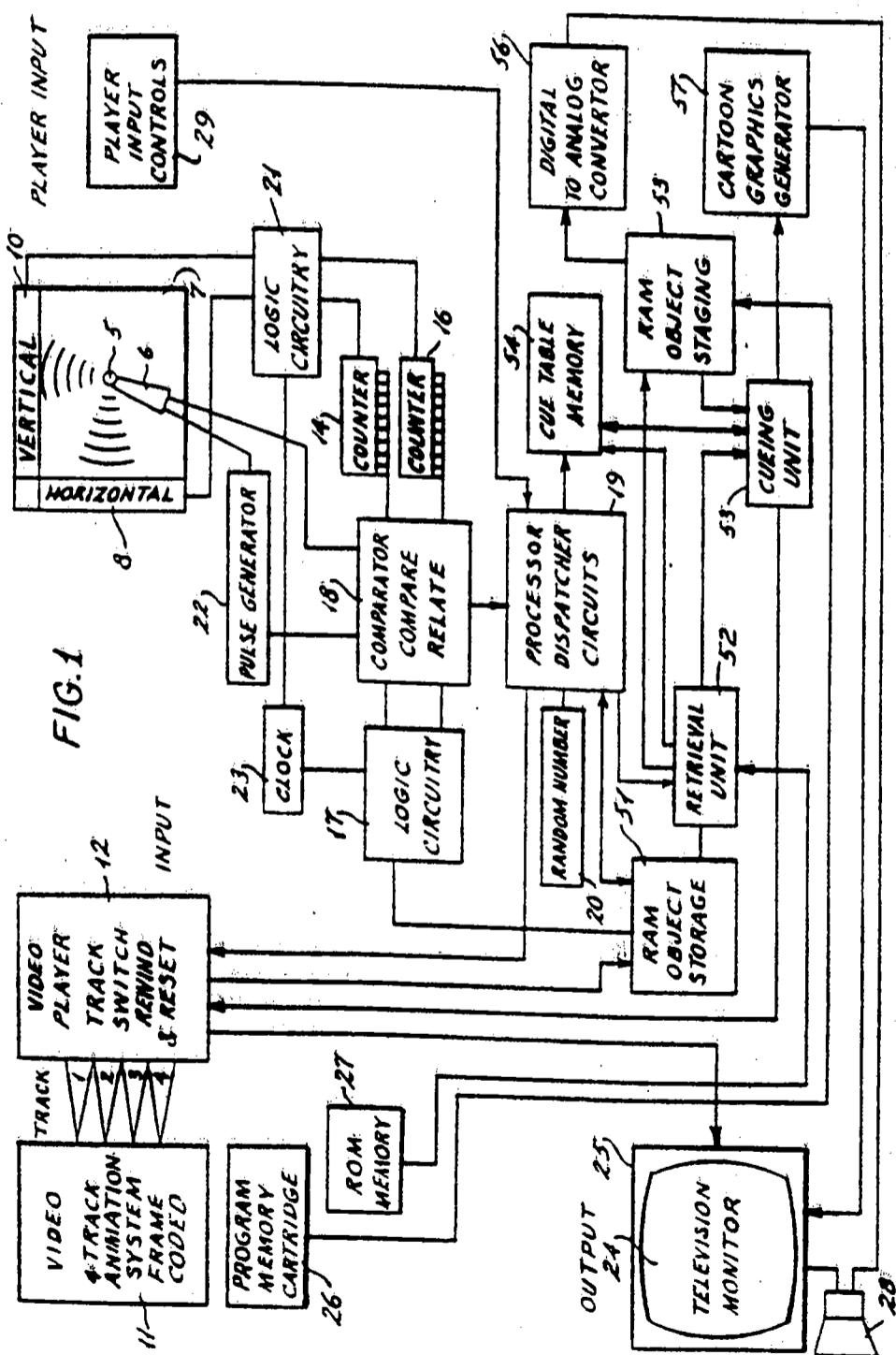
*Primary Examiner*—Clark A. Jablon  
*Attorney, Agent, or Firm*—Schroeder, Davis & Orliss  
[57] **ABSTRACT**

## ABSTRACT

The motion picture branching method is superseded by an animation method which enables rapid and repeated switching of multiple tracks of different camera-originated animation of the same character during continuous action in a scene, and enables branching at the termination of an action to multiple actions or scenes. This method is the basis of a double-circuit video system that enables a player to repeatedly touch or hit an animated character during a continuous action as displayed on a projection screen or television monitor and thus change the action repeatedly. Another system embodiment enables the player to swing a racket before the screen or television monitor, hit the mid-air projected image of a perspective ball animated action, return the ball back to the animated character opponent, and play a simulated game during the the player exercises the same skills used to play the game simulated. An ultrasonic transducer of a playing instrument and a microphone combination on the television face or at angles to the playing action produces a readout of the television product position or the mid-air position of the playing instrument relating to the game. The readout signal is converted into digital form and compared to a similar readout in digital form of the position of the character or object in the frame of the animation displayed by the television and digitally coded in the video tape or disc.

33 Claims, 9 Drawing Sheets





U.S. Patent

Aug. 13, 1991

Sheet 2 of 9

Re. 33,662

FIG. 2

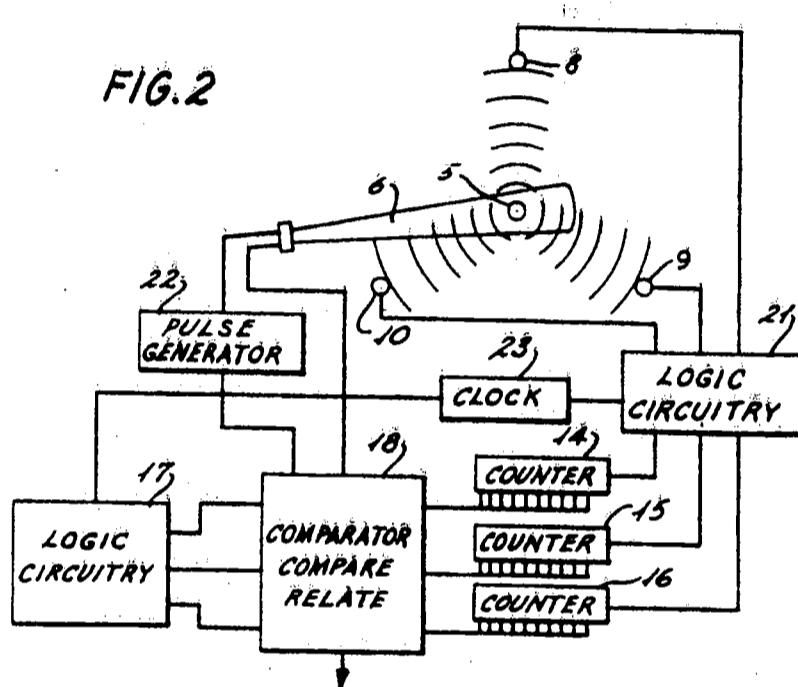
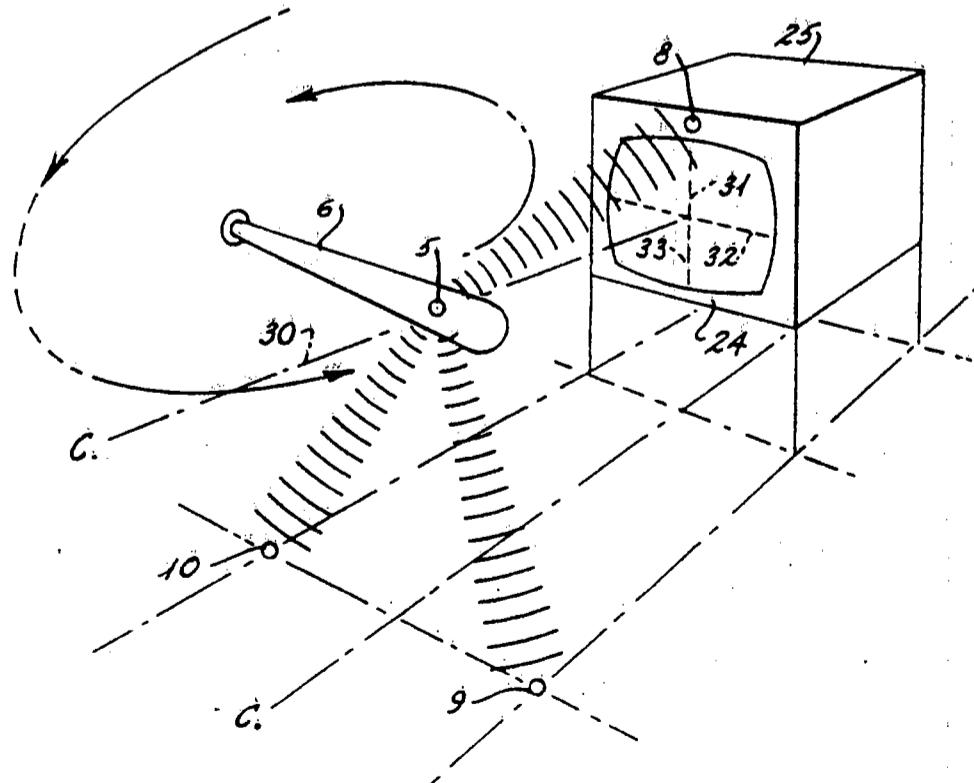


FIG. 3



U.S. Patent

Aug. 13, 1991

Sheet 3 of 9

Re. 33,662

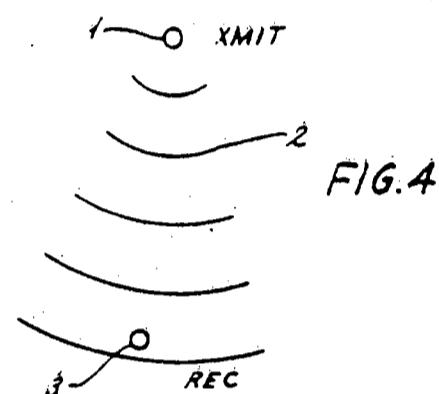


FIG. 4

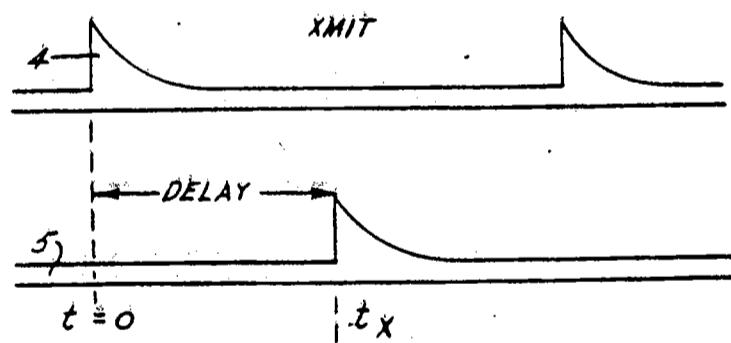
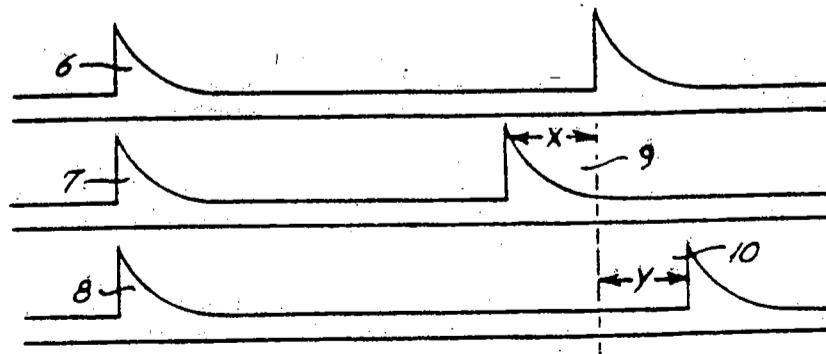


FIG. 5



U.S. Patent

Aug. 13, 1991

Sheet 4 of 9

Re. 33,662

FIG.7

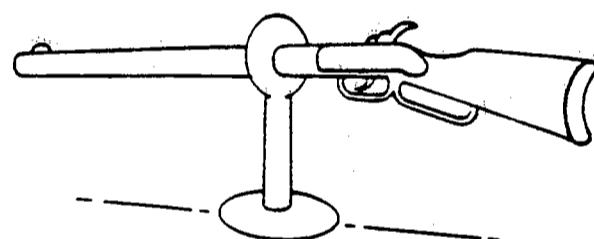
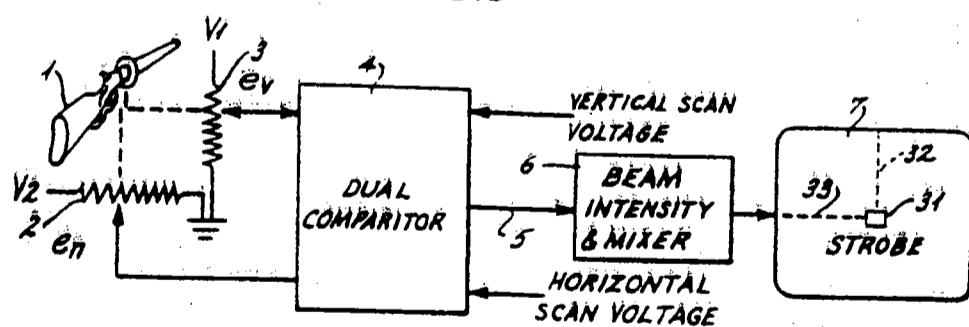


FIG.8



U.S. Patent

Aug. 13, 1991

Sheet 5 of 9

Re. 33,662

FIG.9

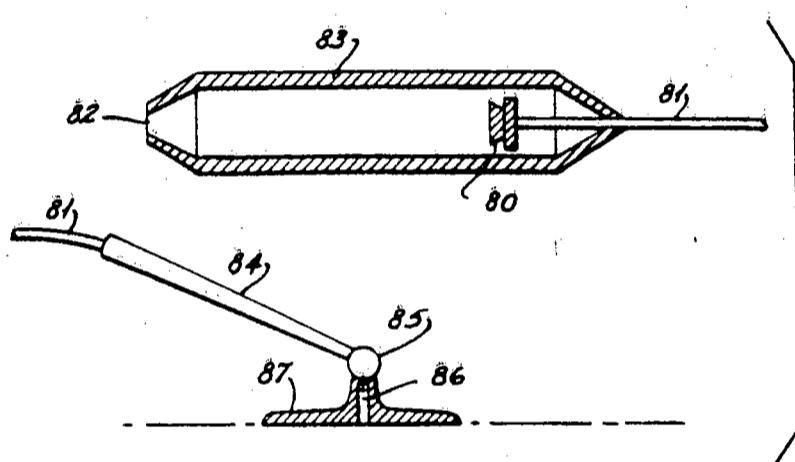
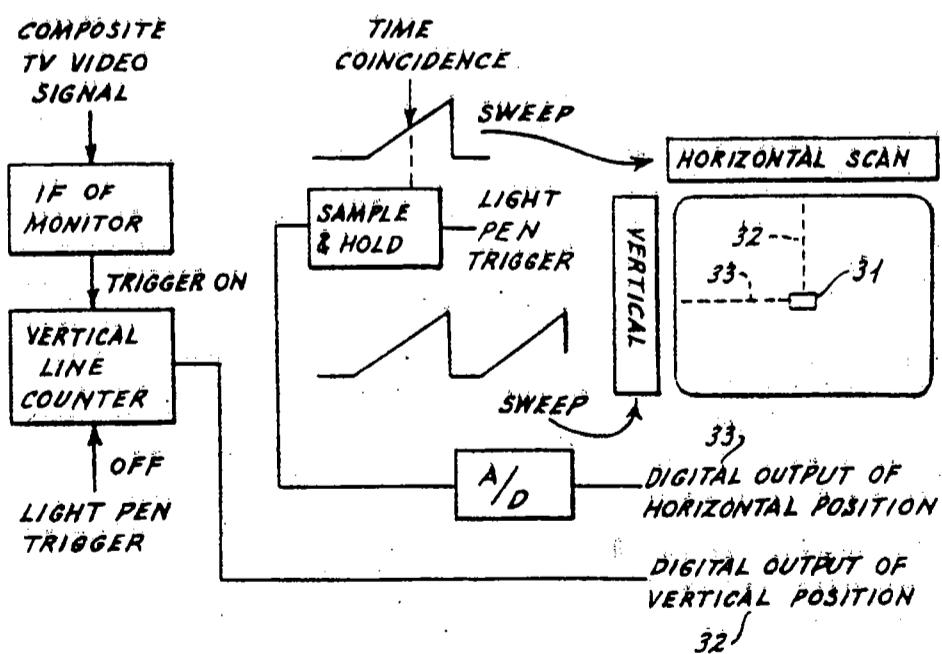


FIG.10



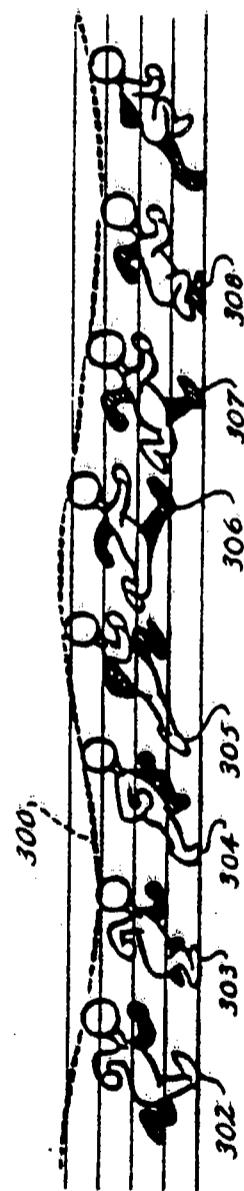
**U.S. Patent**

**Aug. 13, 1991**

**Sheet 6 of 9**

**Re. 33,662**

*FIG. IIa*



*FIG. IIb*



U.S. Patent

Aug. 13, 1991

Sheet 7 of 9

Re. 33,662

FIG.12

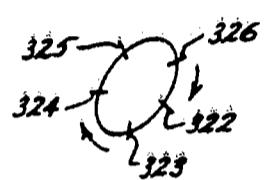


FIG.13

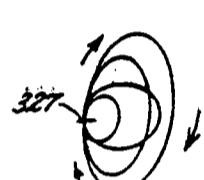


FIG.14

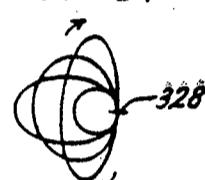


FIG.15

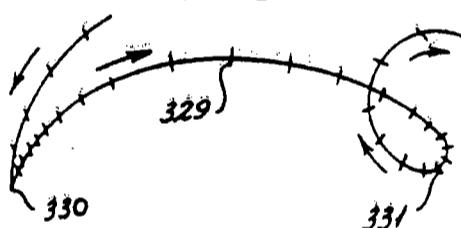


FIG.16

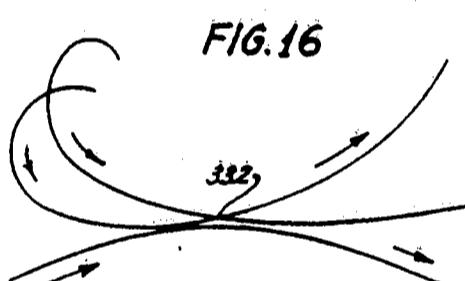


FIG.17

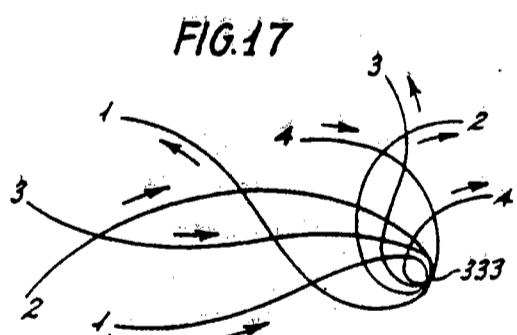


FIG.18

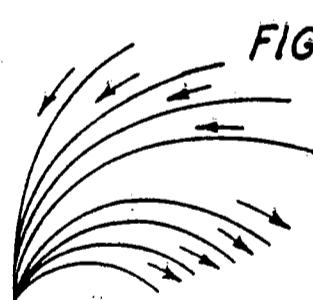


FIG.19

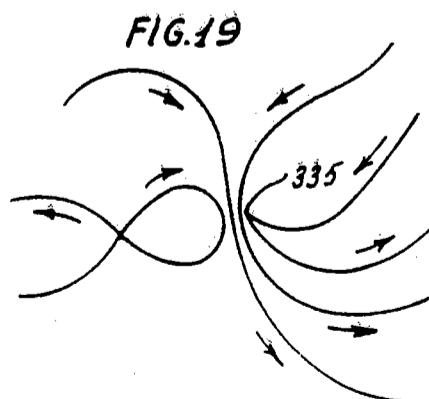
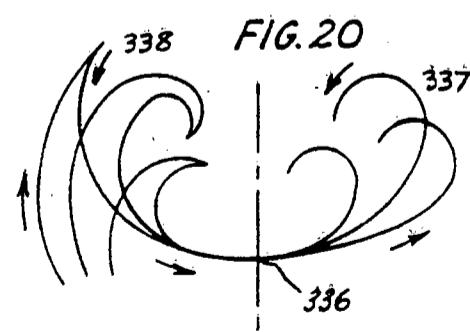
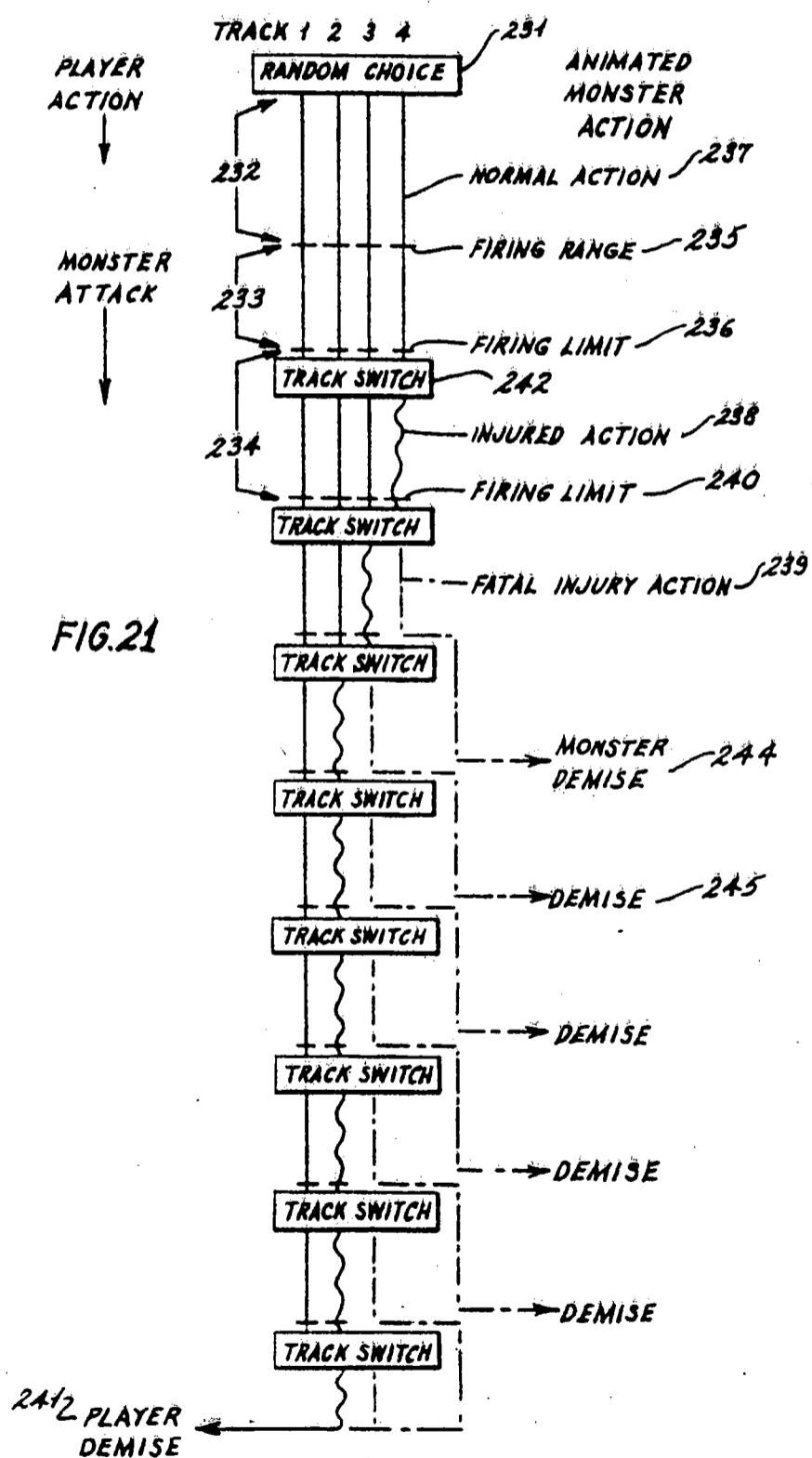
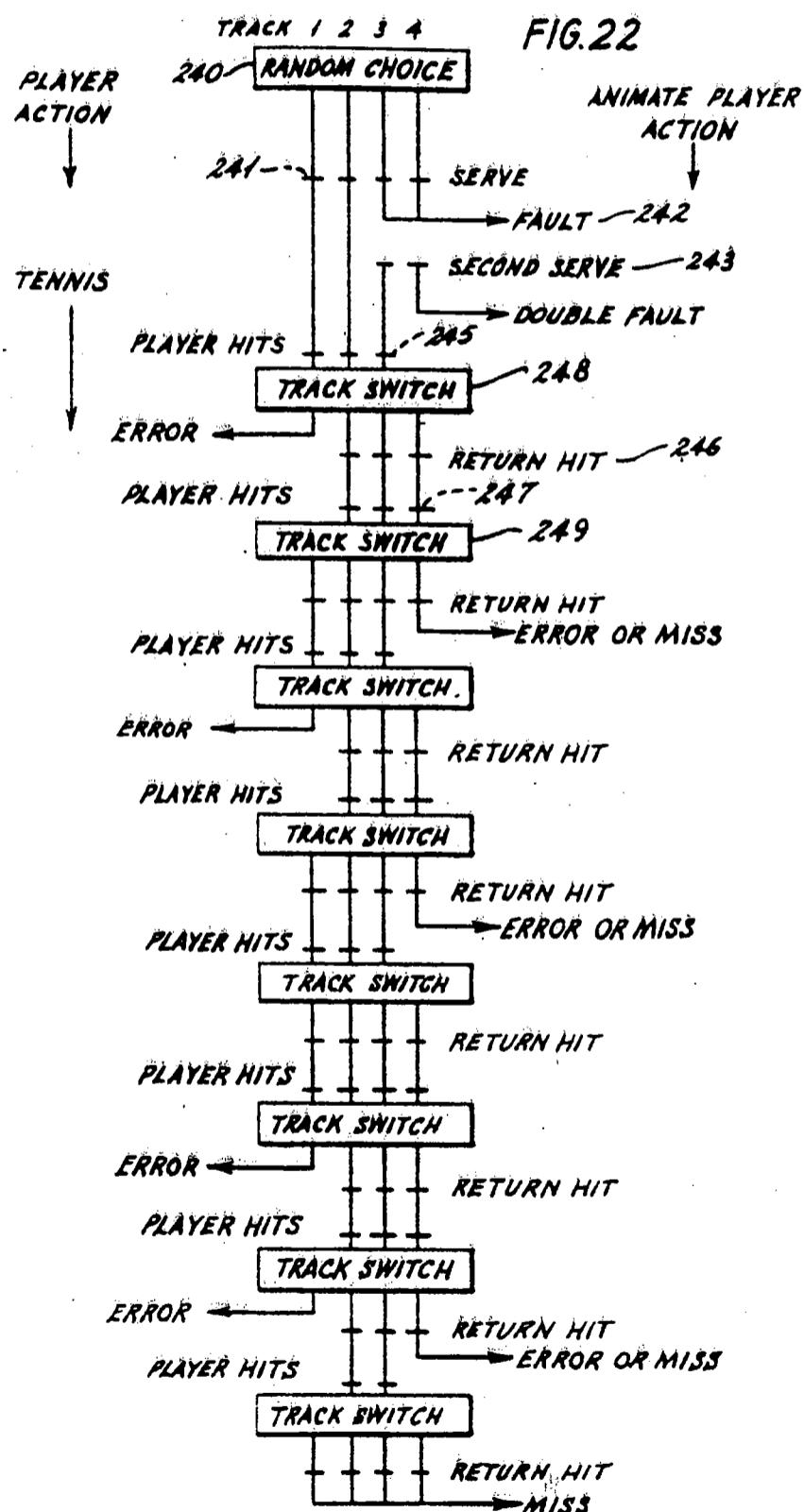


FIG.20







**TV ANIMATION INTERACTIVELY CONTROLLED  
BY THE VIEWER**

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

This application is a continuation of Ser. No. 526,464, filed Aug. 25, 1983 and now abandoned.

**BACKGROUND OF THE INVENTION**

The present invention pertains to an animated cartoon method incorporating a method of motion picture branching as controlled with human skill. The action of the user switches the display to make the picture respond interactively. A plurality of means for such action are specified in this invention. The preferred means for embodiment herewith described is ultrasonic position detection or simulation. The system matches readouts from an ultrasonic transducer and microphone combination placed on the face of a television monitor, or at angles to the user's game action, and from digitally coded data on a videotape, disc, or film, all in the context of a video game or educational system.

Prior-art video game devices enable players to control video images via buttons, knobs, and control sticks or wheels. These operating devices exercise limited finger movements and in no way simulate the actions and skills required of the player in the real-life games simulated by the video games.

Prior art systems are unable to allow the player to strike, throw-at, hit, or shoot a gun at the television monitor and hit a moving animated character or object and change the action multiple times at frictional to several second intervals in the course of an animated scene according to accurate drama and according to the skill of the player or operator in so doing.

Prior art systems are unable to allow the user to swing a baseball bat, tennis racket, or other game instrument interactively in accord with the projected perspective of the animated scene, in front of the television monitor, and thus hit or change animated projectory according to a game or dramatics.

Another problem endemic to the prior art systems with digitally produced animation is the confined and restricted graphics which coincide with needlepoint art in the large grid squares. Primarily two dimensional, such digital animation is inferior to the camera-originated and the computer type animation of theatrical productions. Accordingly, it is the object of this invention to afford the three dimension effectiveness of the camera-originated type animation to video game graphics which are controlled by the player.

Prior art video branching system use apparatus that switches between two or more channels or picture quadrants on the conclusion of a scene or dramatic action. Such switching depends on the viewer's judgement as expressed by pushing, buttons, other restricted video game controls, or a voice response as in the embodiment of U.S. Pat. No. 4,305,131 issued to Best. The result of such switching or branching is the beginning of another scene, episode, or game action. Different type actions can be animated in a transition number of frames into a common matching drawing that affords multiple branching, but these converging actions terminate the

multiple different actions, and result in multiple beginnings of actions.

Accordingly, it is the object of this invention to provide an animation method that affords rapid and repeated switching from and to continuously radically different types of action smoothly, logically, and according to the dramatics.

Another object is to simulate the exact playing action, requiring the player to exercise the same muscular coordination, eye-hand coordination, and kinesthetic skills of the real-life game or dramatic action.

Other objectives are to create a camera-originated three dimensional visual game simulation of the view from a player's eyes that approaches reality; and to accompany the foregoing with the audio voices of umpires, spectators, and scorekeepers together with the sound effects of the operator's and the animated player's action.

**SUMMARY OF THE INVENTION**

The foregoing objects and others are achieved in accordance with the principles of this invention by an ultrasonic transducer and microphone combination that produces a signal which is responsive to the placement 25 of the head of the invention playing instrument on a plastic shield which is over the face of the television monitor. This signal is converted to digital form and compared to a digital readout from the coded motion picture frame of the animated object location and displayed at the time of the instrument contact. This comparison is used to determine a specific alternate track and a switch by the multiple track video apparatus to that track, or to determine that the player had missed and the animation should continue on the same track.

In one embodiment a playing missile such as a ball or dart is cast by the player at the face of the television monitor. A point microphone combination placed on the monitor face and below on both sides of the path of the thrown missile are responsive to an ultrasonic transducer in the missile, and produce digital signals which measure the path of the missile, the speed, and the contact. A readout of this contact digital position is compared to the digitally coded animation.

Three or more point microphones are placed on the monitor face and below a baseball batter, or a tennis player, as a configuration similar in use to a home plate or a base position. The playing instrument, bat, or racket contains an ultrasonic transducer at the simulated ball-contact point. As the player swings the instrument 50 at the animated illusion of an approaching ball, the angles and distances between the transducer and microphone are measured in respect to the passage of the transducer over the simulated home-plate playing and hitting area. The resulting digitally determined mid-air position is compared to the coded animation as a projected position in front of the television where the animated ball illusion would travel if it were reality, and where the player would hit the ball. The speed of the player's swing may also be measured from the microphone readout, and this factor may effect the animated ball return in the event of a hit ball. In the event of a hit or a coincidence of the digital numbers during comparison, the proximity of the exact coincidence, and the angle of coincidence may effect the swinging and resulting ball return action.

Other methods are described that allow the action or the voice of the user to switch the display to make the animated picture respond interactively.

The animation method of the invention applies to multiple tracks of operating animation and the switching thereof. Rapid and multiple switching may occur in the course of a scene of only several seconds duration without effecting the followthrough and smooth action of the animation drawing progression or the illusion of reality. This is accomplished by a method of forced relating and joining interwoven, parallel, and converging animation paths and thus affording switching, back and forth, at close intervals, and by a method of delayed switching to compensate for animated illusion factors or dramatic factors simulating nature.

The system consists of a plurality of game or educational episodes of a minute or more duration and that allow a plurality of switching points— as a tennis volley or a player at bat during a baseball game. On the conclusion of each of these camera-originated animated film episodes, the system switches to a circuit of digitally produced animation displaying scenes that score or supplement the video game or educational system. During this period a random number device may select one of the sixty to a hundred episodes of animation—in the preferred embodiment—and a rewind to the selected episode is effected. Thus the system of multiple scenes interactively switched by the action of the user and multiple episodes that may be determined by random choice, user action, or user choice creates such a complexity of graphic pattern and dramatic odds that the uncertainties of reality are simulated although the animation is pre-photographed, prerecorded, and programmed.

The preferred embodiment of the invention incorporates a single track that carries the audio and picture combination of four different animation actions which in the absence of other processing would be displayed one in each quadrant of the television tube. The system incorporates blanking circuitry which eliminates video signals from all but one quadrant and circuitry that selects this television action in one quadrant and then centers and expands the picture to occupy the entire raster of the tube. The concept is detailed in U.S. Pat. No. 3,837,003. The audio in U.S. Pat. No. 3,845,498.

#### DESCRIPTION OF THE DRAWING

The invention will be more fully comprehended from the following detailed description and accompanying drawing in which:

FIG. 1 is a schematic block diagram showing one embodiment of the invention using a video-tape and a video-tape reader for the primary circuit and a secondary circuit of digitally produced animation for scoring and diagram scenes.

FIG. 2 is a schematic block diagram of an alternate section of FIG. 1. The method incorporates three single-point microphones placed in a triangular pyramid configuration before the television monitor as illustrated in FIG. 3. The microphones measure the distance from a mid-air transducer inside a swinging playing instrument. The digital coordinates of the mid-air ball hit position are compared to the coordinates of a projected position of the perspective in the animation action displayed.

FIG. 3 illustrates the placement of the single-point microphones in a triangular pyramid configuration with the playing instrument transducer before the television monitor.

FIG. 4 diagrams ultrasonic transponder transmission of pulse waves to a single-point receiver.

FIG. 5 diagrams pulses transmitted and received.

FIG. 6 diagrams transmitted and received pulses graphically explaining the Doppler Effect.

FIG. 7 is an illustration of a rifle version of a video game Joy-Stick instrument used to control a cursor.

FIG. 8 is a schematic diagram of an alternate section of FIG. 1 incorporating the Joy-Stick rifle as a rheostat determining the strobe position on the television monitor. The coordinates of the strobe position are compared to animation position coordinates as diagrammed in FIG. 1.

FIG. 9 is a cross section diagram of a playing instrument, light pen, or other embodiment incorporating a photo cell or diode light measurement.

FIG. 10 is a schematic block diagram of an alternate section of FIG. 1. A light pen or other embodiment of this light measurement device incorporates the diagrammed circuitry to establish player determined position coordinates which are compared to coded animation coordinates.

FIG. 11(a) and 11(b) are illustrations of animated walk and run cycles. The primary path of character action is shown. The illustration of the run cycle reveals a wider action in the path of the character drawings.

FIG. 12 is a diagram of a path of action for a character running or walking in a stationary position on a moving pan background.

FIG. 13 is a diagram of four different character paths of action all moving on a pan background, and all incorporating a coincidence at the lift drawings in these cycles of the same animated cartoon character.

FIG. 14 is a diagram of four different paths of action, all moving in a stationary position on a moving pan background, and all incorporating the coincidence of a forced drawing at a universal position in the sinking action of the cycles.

FIG. 15 diagrams an angular reversal of action, a widely spaced fast action progression of animated drawings, and a circular reversal in the path of action.

FIG. 16 diagrams a point of parallel between two different animation actions and an intersection of paths of actions that are moving in the same general direction.

FIG. 17 diagrams four converging circular reversals actions at a drawing with forced details designed to fit all four actions and serve as a switching point.

FIG. 18 diagrams four animated actions converging at the point of angular reversal in a drawing with forced details designed to fit all four actions.

FIG. 19 diagrams a switching point of the invention animation method incorporating a circular reversal, a parallel action, and an angular reversal of animated action.

FIG. 20 diagrams three different golf swing actions that coincide according to the forced drawing method at the point of golf ball contact, loss of silhouette definition, and widely spaced fast action.

FIG. 21 is a schematic diagram of an embodiment of the animation method that schedules an episode single scene of an attack against the player by a dragon, monster, tiger, bear, cave man, man from space, or other dangerous adversary.

FIG. 22 is a schematic diagram of an episode of the animation method as one embodiment of a tennis game volley.

#### DETAILED DESCRIPTION

Referring to FIG. 1 there is shown a block diagram of the invention video amusement system as an embodi-

ment of two different types of video-game circuits. The primary circuit incorporates camera-originated motion pictures with audio as supplied by video tape 11 and video tape player 12. The invention method of animation switching is incorporated in the animation art and coded in the track 1. At the completion of a game episode the tape rewinds to another game episode according to a random number device 20 (pseudo-random number device). During this rewind a secondary animation circuit displays the scoreboard with the score of the game and related audio. The secondary circuit incorporates digitally produced animation. In embodiments the secondary circuit may supply the audio of audience response, score announcements, and the video of superimposed scores similar to a DX run on an animation camera. The secondary circuit embodiment diagrammed in FIG. 1 is prior art which is described in detail in U.S. Pat. No. 4,305,131 Best. The embodiment diagrammed is incorporated in said patent of a video game system which affords simulated conversation and voice orders between the human operator and the apparatus. Accordingly, the complete apparatus diagrammed in FIG. 1 may be enlarged to constitute a video game that allows the player to talk to, and answer questions from the apparatus with verbal response and orders that effect the course of the animation, and to touch or otherwise contact the animation picture and effect the course of the animation, or a combination of both verbal response and player contact with the televised animated character.

The invention encompasses the multiple combinations and variations of the specified elements of the invention. For example, the player's skill or mental choice as indicated by touching or speaking creates a change in the animation as described herewith. Hundreds of other scenes may follow that are a continuation of the player's choice or skill as specifically related to this one original scene. Instead of a rewind to another scene or episode that is determined by random number mechanics, as specified in the preferred embodiment, the player's action or choice may determine rewinds to another scene or episode. During such rewinds the secondary circuit displays digital animation scenes as described in the preferred embodiment; however, the scope and spirit of the invention encompasses the obvious addition of video disc player or another tape player to the invention secondary circuit that would effect a continuous series of camera-originated animation scenes. Further, the use of the word "animation" in the present invention application in no way restricts the obvious scope and spirit of the invention to hand drawn animated graphics. The invention encompasses any type of timed or humanly manipulated motion pictures. Such scope encompasses optically-timed live-action motion pictures and computer type motion picture graphics.

Referring to FIG. 1, a clear transparent sheet of heavy plastic 7 is installed over the screen 24 of the television monitor 25. The two strip microphones 8 and 10 are mounted along the mutually perpendicular edges of the sheet 7 in a position above and at the side of screen 24. A playing instrument 6 includes a tip 5. Instrument 6 is connected to a pulse generator 22 which causes the tip 5 of instrument 6 to produce repetitive ultrasonic pulse outputs. The ultrasonic pulses can be produced by a ceramic transducer or high voltage sparking between two electrodes on the instrument 6. Instrument 6 also includes an internal switch which

indicates whether or not the contact tip 5 is in contact with the surface of sheet 7.

The two strip microphones 8 and 10 are responsive to the ultrasonic pulses produced by instrument 6. These microphones 8 and 10, in conjunction with clock 23, counters 14 and 16, and logic circuitry 21, measure the time for the propagation delay between the radiation of a pulse at instrument 6 and its arrival at the respective microphones 8 and 10. Logic circuitry 21 can comprise circuits well known in the art for combining the outputs from microphones 8 and 10, pulse generator 22 and clock 23 to control counters 14 and 16 which count or measure the time delays. Counters 14 and 16 also comprise known electrical circuits. The measured time delays establish the coordinates of the location of contact tip 5 on the surface of sheet 7 at the time of any particular pulse output. The foregoing method of ultrasonic pulse measurement to establish coordinates of an instrument contact location is prior art. The application of this ultrasonic method to a television video recording apparatus using the invention animation switching method is unique and unknown in the art.

The animator's field position of the vital target point in the animated character or object is entered by the animator on each frame of the production exposure sheet that instructs the exposure of those drawings of animation that detail an object in a position to be contacted, fired at, shot, hit, touched, or otherwise effected by the player's action and skill. Such locations per frame are converted to digital binary coordinates. The motion picture frames are counted, specified, and used to enter a table of values holding these object or character positions and vital data on the tape or disc at the beginning of each episode. All of these numbers are stored in the circuit R A M 51 of FIG. 1 and retrieved when applicable to the animation action and video game play. This process is simpler than recording the data on each frame as it appears. The frame number is recorded at intervals on the video tape track 11 and used as a track guide instead of a frame meter which would create problems in the event of a tape breakage. Special track coding allows fast rewinds to specific frames located from track signals. Referring to FIG. 1 (may be FIG. 11) the coded field positions with frame numbers are stored by player 12 from tape or disc 11 into R A M 51 at the beginning of each episode. This table of values in R A M 51 supplies pertinent data to logic circuitry 17 on time to be compared to the coordinates of the player's actions and skills as described heretofore in the various embodiments of position determination.

The comparator circuit 18 of FIG. 1 compares the two field coordinates derived from the digital readouts of counters 14 and 16 to the two coordinates of the animation target position from logic circuit 17 and R A M 51.

The comparator 18 instructs the processor circuit 19 if the player has scored a hit at, or close to, the point of coincidence, giving the degree or proximity of the hit to the point of coincidence and the angle of animation progression that will result from an off-center hit. Based on this data an order for a track switch, if indicated, is dispatched by processor 19 to video tape player 12 which switches between tracks 1, 2, 3, and 4 based on the program of the episode and track instruction from program memory cartridge 26, R A M 55, and cueing (cuing) unit 53.

Referring to FIG. 1, the secondary circuit animation and scoreboard graphics is generated by cartoon graph-

ics generator 57 from digital data which may be read along with digitalized audio from R O M memory 27 or other magnetic mass-storage device. Retrieval unit 52 is a conventional peripheral input controller which stores into memory the digitally coded blocks of information obtained from R O M memory 27. This information includes control data which retrieval unit 52 stores into random access memory (R A M) 51 for use by dispatcher unit 19, and audio and/or graphics data which unit 52 stores into R A M 55 for use by cueing unit 53. The control data includes cue commands and schedule commands. Cue commands specify short term operation during an interval of time, while schedule commands represent longer term points in time, and form chains which define and relate to alternate (multiple track) schedule. Dispatcher 19 controls the course of the animated cartoon or scoreboard audio-video and stores cue commands into cue table 54. Cueing unit 53 executes the cue commands. Cueing unit 53 repeatedly scans cue table 54 to get commands telling it what to do and the time it should done. Dispatcher unit 19 may request successive blocks of control information from retrieval unit 52 and output into cue table memory 54 a schedule (called a cue table) of operations for cueing unit 53. Dispatcher 19 repeatedly updates the cue table schedule as the cartoon progresses. Dispatcher 19 processes the various optional player input control 29 which may be input via the conventional video game hand-held instruments and stores the different player commands into cue table 54.

As described, dispatcher 19 controls the course of the cartoon and stores cue commands into cue table 54. Cueing unit 53 executes the cue commands at the times specified therein by conveying to the cartoon graphics generator circuit 57 blocks of binary-coded data previously stored into R A M 55 by retrieval unit 52, and these blocks of data are used by the cartoon graphics generator 57 to generate cartoon frames which are then displayed on television monitor 25. Digital audio passes from R O M memory 27 through retrieval unit 52 to memory 55 to digital to analog converter 56 and hence to system speaker 28. The binary coded data stored into R A M 55 is reinforced by individual game data supplied by program memory cartridge 26. The circuits described as components of the secondary digital animation circuit are known in the prior art.

The primary four track video tape animation circuit of the present invention, which is diagrammed in FIG. 1, is operated and scheduled by the processor dispatcher 19 which has control of the course of the camera-originated animation. The comparator 18 furnishes the results of the player's action to the processor 19 which instructs the switching of tracks 1, 2, 3, and 4 with game position to video tape player 12. At the termination of a game episode either random number generator 20, player input 29, or instrument 6 placement on sheet 7 specifies an episode to processor 19 which instructs tape player 12 of the end. The progress of the game comprising both animation circuits is controlled by processor 19 based on data from program game memory 26 and R O M memory 27 and the operation of the secondary circuit retrieval, cueing, and memory circuits as described above. As ordered by processor 19, this instruction is sent to tape player 12 by the cueing unit 53.

The invention operates using film, video tape, video disc, or digital stored and/or generated animation. All of these video mediums can be coded and adapted to the

invention circuits and apparatus. The embodiment described is video tape.

The motion picture film embodiment of the invention operates using multiple film tracks of operating film read by multiple heads, multiple tracks of animation exposed on one motion picture film in a parallel or alternate exposure, conventional single frame exposure with each frame carrying a picture combination of four quadrants, and multiple tracks of such quadrant type film. All of such film methods may be incorporated in film cassettes. Film quadrants are masked in projection. Such film is projected on screens that are related to the microphone installation described. The large and wide screen projection afforded by film creates a panoramic animation display that is interactive with the player or human operator use of large instrument simulations as airplane cockpits, automobile driving interiors, or various weaponry all of which move when guided by the player during action requiring a simulated skill. Simulated audio effects and the use of voice recognition, command, questions, and animation reaction to such audio adds to the simulation effect.

In FIG. 1 the digital memory cartridge 26 can be various types of memory and may be plugged in to change the scenario. And, memory cartridge 26 may supply the coded animation used by the primary circuit of the invention.

The video tape embodiment of the invention may operate using multiple video tape tracks read by multiple heads, multiple video and audio tracks incorporated in a single tape on a line sharing basis, or a single track signal that carries a picture quadrant using a blanking circuit that centers and enlarges the selected picture, or a combination of the above methods.

An embodiment of ultrasonic position detection FIG. 2 and FIG. 3 incorporates an installation of single-point microphones for the face 24 of television monitor 25, which is the location of microphone 8, and to the right and left of center on the floor, which are the locations of microphones 9 and 10. The circuitry of counting the microphone reading of pulses is similar to FIG. 1. FIG. 2 is a block diagram section which replaces the similar circuits and apparatus of FIG. 1. Using the three triangular formation variable measurements, 5 to 8, 5 to 9, 5 to 10, a mid-air location of the hitting point transducer 5 of the swinging bat is determined and interactively related to the animation—which may be a baseball pitcher throwing a baseball, in perspective, at the player. Simulating reality, the player swings the bat 6 or racket at the oncoming animated ball. As the transducer 5 in the bat passes a mid-air position over the plate (or triangular microphone installation) where the perspective projection of the animated ball (if it were real) would be hit, the proximity to coincidence of animation code position and player instrument is determined, the angle between ball code position and bat hit position 5 is determined, the velocity of the player's bat is measured from the microphone readout, and the animation reacts accordingly.

In other embodiments of ultrasonic position detection the playing instrument 6 of FIG. 2 or FIG. 3 may be a portable missile such as a dart, rubber ball, shuttlecock, or suction tip arrow that comprises batteries, electrodes, and switches to effect ultrasonic pulse output. A switch assembly may be installed under the sides of a clear plastic shield over the face 24 of the television monitor 25 of FIG. 3 that would be responsive to pressure on the plastic shield, and thus indicate the point in

Re. 33,662

9

time of missile contact. The position of the thrown missile is calculated by three vectors 5-8, 5-9, 5-10 diagrammed in FIG. 2 and FIG. 3. Another embodiment uses the two strip microphones 8 and 10 of FIG. 1.

Various means of ultrasonic measurement may be used according to game needs, distances, and vector geometry. FIG. 4 shows an ultrasonic transponder XMIT 1 which transmits audio waves 2 or pulses in response to an electrical signal. These are received by single-point microphone REC 3. FIG. 5 shows these pulses transmitted in position 4 and received at position 5. To avoid problems with ambiguity, the transmit pulses must be spaced in time farther apart than the longest distance of interest to be measured. A clock or digital counter is set to zero by the transmit pulse and started up.

When the first part of the received signal arrives the timing is stopped. The time  $t_x$  of 5 can be converted to distance when the velocity of sound in the medium is used as a scale factor. This process is simple if it is all in one plane, and if point receivers and transmitters are used, and there is no reflections. An additional measurement can be made using the Doppler Effect. Refer to FIG. 6 position 6. This is the same as FIG. 5 position 4 and represents the case where the transmitter XMIT is not moving relative to the receiver REC. If the transmitter XMIT is controlled to send out pulses at fixed time intervals measured within the transmitter, the additional Doppler Effect measurement can be made. In FIG. 6 position 7 the transmitter is moving toward the receiver and moves the distance X 9 between pulses. Likewise as in FIG. 6 position 8 where the transmitter is moving away a distance Y 10 occurs. The Doppler Effect is determined by measuring the frequency of the received pulses. This measures the velocity of the transmitter relative to the receiver, and is a factor in determining the velocity of the transmitter playing instrument.

An embodiment of position detection incorporating a Light Pen is diagrammed in FIG. 9 and FIG. 10. The Light Pen is so called because it is used like a pen - but it does not produce any light. In FIG. 9 a photo cell or diode 80 is enclosed in a case 83. An opening hole 82 allows light exposure to reach the diode 80 which is connected by, and signals on, the wire 81. Such device is incorporated as part of a playing instrument of the invention video game—or educational system. The Light Pen is held on, or close to, the face of the TV CRT with the opening 82 lined up parallel to the face. The detector inside produces a pulse when the cathode ray beam of the television sweeps past the Light Pen opening 82. The pulse signal is used to "tell" the system where the Pen 83 is located by referring the pulse back to the sweep circuits of the TV system. Referring to FIG. 10, the horizontal position is found by taking a sample of the sweep voltage at the pulse time, and from this determining the percent of the sweep or time. The vertical position of the Pen contact is determined by counting the sweeps to the contact point on the face of the monitor. The Light Pen device may be encased in a plurality of forms or playing instruments. One embodiment similar to a spatula is illustrated in FIG. 9. The handle holds a limited movement knuckle joint 85. The Light Pen 86 is encased in the extension from the knuckle joint 85 to a contact surface 87 which may be in multiple shapes including a strung racket to simulate a badminton racket. Thus the player touches the television face with the simulated racket to hit and return an

10

animated shuttle-cock across the net to an animated opponent.

Other embodiments of position detection incorporate known circuits and devices as those named a Mouse, a Tracking Ball, and a Joy-Stick. The Tracking Ball and Joy-Stick use the same type of circuitry as diagrammed in FIG. 8. Such devices control the movement of a strobe, a cursor, or some other displayed element. FIG. 7 illustrates a mounted rifle playing instrument which is a variation of a Joy-Stick. The aiming of the rifle in a socket joint moves rheostats 2 and 3 as shown in FIG. 8. The voltage readout effects means which are well known to effect a strobe. Pulling the trigger of the rifle activates the apparatus. Using this position orientation as a guide the player reaims the rifle to hit or cover the animation target and effect the animation via the unique means of this invention. Other circuitry for the rifle device uses a simple switch, rather than the voltage readout. The switch turns on a counter which is shut-off by the player when the strobe spot is in the desired position. The counter signal can be used to move the strobe or cursor—and activate means giving a digital readout to compare to the coded animation position. Combinations of such strobe position means may be incorporated with the previous devices described as additions to enhance simulation.

The animation method comprises means of rapid and repeated switching in multiple possible directions from converging multiple directions in paths of action, and in this process retaining a continuity of shape progression that is dramatically believable and does not violate an illusion of reality while retaining widely divergent drawing in the key extreme positions between the different actions and thus retaining the character of each action. The invention applies this method to a system of multiple switching positions that may be spaced (timed) as close together as twelve frames ( $\frac{1}{2}$  second) or longer periods that amount to an entire scene. A plurality of such switching positions may occur at even intervals or widely variable intervals during the course of the animation.

For an understanding of animation technique, the reader is referred to two art instruction books by the inventor, Preston Blair, titled: "Animation" No. 26 and "How to ANIMATE Film Cartoons" No. 190 published by Walter Foster Art Books. The examples of an animated walk and run cycle illustrated in FIGS. 11(a) and FIG. 11(b) are from "Animation" No. 26.

The invention can accommodate the switching at certain times in an action to make the animation look smooth. At these proper moments in a movement a degree of forced drawing is introduced to also accommodate the switching. The types of switching points of the invention are illustrated in diagrams and drawings in FIGS. 11(a) through 20.

In FIG. 11(b) the path of action 301 indicates the primary direction of movement of the character in a walk cycle. Path of action 300 indicates the wider movement of a run cycle. The paths of action referred to in this invention are these primary or principle paths of action of the entire character mass, as indicated by paths 300 and 301. Secondary paths of action in a character as the reversing swing of the arms and hands and the circular lifting and falling of the feet action with other secondary animation paths may, in certain scenes and staging, become the primary action of the scene.

An animation character walk or run cycle constitutes two steps or forward leg actions. The switching points

Re. 33,662

11

of the invention occur at points where all converging animations have the uniform right or left leg action. In FIGS. 11(a) and 11(b) the extremes of action are the point of foot contact 302-307 and 312-317, the recoil position 303-308 and 313-318, and the high lift position 305 and 315. The extreme arm and leg positions of the walk 302-307 and run 312-317 have silhouette definition that separates and gives character to each action, whereas, the method points of silhouette loss and action convergence are during the upward lift action 304 and 314. The action drawing at these points is forced into a brief proximity by the invention method for switching purposes. The method forcing or revision of the action in both side-view cycles is effected without damaging or causing the two actions to look like each other. The reason is that the vital silhouettes of the different looking extremes have not been tampered or altered. In side-view actions, as here illustrated, switching occurs—and method forced drawing occurs—at areas of animation wherein silhouette definition is lost, which is due in this case to the passing of the arms and legs in action. Other cases of silhouette loss and opportunity for method forced drawing occur in foreshortened perspective action. These are actions wherein the character is progressing directly at, or away from the viewer. An opportunity for the method forced drawing occurs in most animated actions. The method is illustrated by different golf swings as diagrammed in FIG. 20. The wind-ups 338 differ widely in path and character drawing as do the follow-through actions 337. The method switching and forced drawing is at the ball contact 336 when silhouette definition is lost. Position 336 also illustrates another method switching and forced drawing point in an animated action. Such points occur at the center of widely spaced or fast actions. This type of switching point is diagrammed in FIG. 15 position 329. All types of leaps, jumps, dives, or other quick actions contain such switching and forced drawing points that allow for the method. The movement of the background effects the pattern of paths of action. The walk and run cycle becomes circular in the example diagrammed by FIG. 12 wherein the character progresses in a stationary area while on a moving pan. The switching method point of FIG. 12 cycles would thus be at the rising action point 324 or 325. Four such cycles thus converging momentarily for switching in FIG. 13 incorporate the switching point at the lift 327. Other types of cycles offer switching according to the method in FIG. 14 at the sinking point 328 of the actions.

The method switching points also occur at points in actions of momentary uniformity as: parallel actions, circular reversals or recoils, angular reversals or recoils, various intersections of actions that are angular and traveling in the same direction together, a uniform turning of characters, and twirls. Obvious switching points in animation are also used by the invention as: from a pose or held position, from a point of brief disappearance of the characters behind objects, or from brief movement offstage.

FIG. 15 diagrams the method points in actions of angular reversal 330 and circular reversal 331. Both of these actions occur in recoils and many other actions. FIG. 18 diagrams four actions into a common point of angular reversal 334. FIG. 17 diagrams four circular reversal actions that incorporate method switching at point 338.

FIG. 16 diagrams a method switching point at an angled intersection of two actions (moving together)

12

and a third parallel action to the point of intersection. Thus FIG. 16 shows two switching examples: during parallel actions, and during actions with angled intersection.

5 Combinations of switching patterns are diagrammed in FIG. 19. The switching point 333 combines two parallel actions with an angular point of reversal and a circular point of reversal. Accordingly, the invention method enables many different types of animated action 10 to be connected by multiple switching points which are of multiple types in combination.

Each episode of the invention games is diagrammed or planned according to a schematic schedule. Educational system lessons are similarly planned.

15 Referring to FIG. 21, a schematic schedule of an embodiment of a dragon or monster attack episode diagrams a three dimension foreshortened action which races forward from the distant background directly at the player. The video tape rewinds to one of a hundred different wild animal attacks determined by random number mechanics. One of four tracks is selected by random choice 231. In area 232 the monster advances from the distance reaching the firing range of the player 235 at which time the frames are coded allowing a player hit. During the advancing area 233, a hit in a vital area of the dragon monster will effect a switch of the animation at track switch 242 to the track four injured action 238 of the monster. A scheduled delay between firing limit 236 and the track switch 242 compensates for dramatic and animation factors needed in certain actions. If the player can hit the injured monster a second time in the following advance area 234, the injured action track action is switched to the fatal injury action 239 at switching point 243. The animated 20 demise of the attacking monster occurs at demise 243. Thus two hits are required to injure the monster and fatally wound the beast in two advancing areas. Each demise 244, 245, etc. is a different dramatic end of the monster. If the player's skill is insufficient to save him from destruction by the monster he will meet his fate at player demise 241. If the game is played by two players, the survivor is the winner. If played by two teams, the surviving team wins—or those still alive. Variations of this game and schedule incorporate space attack by alien spaceships, underwater shark attacks, aerial dog-fights, or shoot-outs of western or other styles, etc. In a game in which the attacker shoots at the player, the vulnerable position of the player is registered by the player using the instrument position detection and establishment means of the invention. Such player position embodiment may incorporate a gun belt with the transducer 5 of FIG. 3 installed in the buckle center.

25 Referring to FIG. 22 a schematic schedule of a tennis game volley is diagrammed. The game is played by two players who alternate in volleys against an animated tennis opponent, and the game is scored exactly as tennis. At the start of each volley, the video tape rewinds to one of a hundred different tennis volleys that is determined by random choice. The volley begins with the random choice 240 (unevenly weighted) of one of four tracks of the animated tennis player preparing for the serve 241. On track 1 the serve is good and the ball animates to the foreground to a position where the human player hits the ball 245. On track 2 a similar action occurs with a different action and paths of action. On track 3 the first serve is a fault 242, but the second serve is good and the ball advances in a different path of action to the player hit positions 245. On track 40

4 the animated player double faults 244 and the volley is over. If the human player misses the ball the volley is over at track switch 248. If the player hits the ball 245, the track selection 248 and return of the ball is determined by the accuracy and the related angle of return. Track 1 hits the net in an error and the volley ends. Track 2, 3, and 4 return the ball in different patterns which converge as the animated character hits the ball at the return hit 246 back to the following human player hit position 247. The game is played from the perspective of the eye of a tennis player. The ball is hit from the distant background and comes forward in perspective to the player hitting area. As the game moves from side to side to then tennis court the perspective of the entire court is animated—using computor type theatrical animation—to simulate reality. Thus the camera-originated animation includes both the hand-drawn full character animation and the computor type animation used in science fiction features.

Many game embodiments may be simulated by the invention animation and player action. Suspense and dramatic situations place the player in a simulated danger or in positions requiring skills and reflex action. Such animation simulates, for example, the three dimensional visual action confronting the eye of a space traveler, an explorer attacked by ancient monsters in an unknown land resembling the pre-historic, an underwater swimmer attacked by huge sharks, octopuses, torpedos from enemy submarines, and frogmen denizens of the deep. An eskimo is attacked by polar bears and wolves. On an African Safari rhinos, elephants, and tigers attack the player. Based on World War I airplanes, an aerial dogfight casts the player in the cockpit of a plane being attacked by on-coming planes and destroying enemy planes and dirigibles according to his accuracy with a gun. The clouds of the sky pass by in the realism of feature type three dimension animation art. The player searches every mist for the appearance of the devil Red Baron and his squadron of executioners. Suddenly the sound of enemy machines guns fill the air as planes attack from every angle. If all forms of combat the player may use the ultrasonic detection means of the invention to shoot at attackers, and the ultra sonic means enable the animated attacker to shoot at or contact the player - thus ending the game. In these 45 many fantastic adventures the player is allied and helped by friends. Dolphins help the player attack huge strange denizens of the deep sea. SUPERMAN, BAT-MAN, BUGS BUNNY, THE HULK, WONDER WOMAN, friendly space creatures, or the likes of DICK TRACY help, talk to the player, ask the players questions about strategy, and act according to the player's verbal response as described in the previously referenced Best ('131) patent. Thus, funny rabbits, wise owls, friendly raccoons, big birds, small mice, giant dragons, and tiny insects may speak, act, and reason with the player in educational games based on player action and voice.

PING-PONG, badminton, volleyball, baseball, tennis, skeet-shooting, and other ball-return games are 60 embodiments. Games as baseball may use camera-originated animation with diagrammatic digital animation. Games may be played on diagram scenes displaying playing boards that use episodes of camera-originated animation to advance the game play.

Other games of combat incorporate embodiments wherein the player evades being shot be gunfighters or hit by cavemen throwing stones by his movement be-

fore the face of the television monitor. A transducer mounted on the player creates a digital readout that may coincide with digital position of the oncoming bullet or other animated projectile unless the player is able to move out of the way—or line of fire.

While the invention has been described with reference to a specific embodiment it is understood that various modifications, alternate construction and equivalents may be employed without departing from the true spirit and scope of the invention.

The invention uses are not restricted to video games, and the uses of the invention incorporate educational systems with memory circuits that record and grade the player action and response.

Alternate versions of the circuits described may be used to displace those specified. Therefore, the above description, circuits specified, apparatus specified, and graphic diagrams should not be construed as limiting the scope of the invention.

The invention is the combination of the above methods, circuits, and apparatus. Such a combination results in a unique operation and unknown function in prior art, although fractions of such combination are prior art.

The invention combination is defined by the appended and following claims.

I claim:

1. A video game system providing repeated switching of multiple tracks of different actions of the same animated character according to the skill of the operator in contacting the camera-originated animation display, comprising:  
multiple tracks of animated motion picture production of different actions of the same character providing coded frames for track switching and coded location coordinates of said character target,  
a video-audio input terminal providing means to operate multiple tracks of animation, providing for the switching thereof during operation, and providing for film track selection, masking, and centering to the full raster,  
a player input terminal including a playing instrument deployed before the display with a transducer and a microphone combination mounted on the mutually perpendicular sides of the face of said monitor, providing means for producing digital signals representative of the two coordinates of the location of said playing instrument with respect to the image on said monitor,  
means of entering a table of values representing digital coordinates of the locations of animation targets per frame and means for entering episode cueing (cuing) data into memory storage at the start of the episode,  
means of retrieval of said digital coordinates of the location of said target area at a designated frame from the memory storage,  
means of comparing and matching the coordinates of the location of said playing instrument transducer to the coordinates of the location of said animated target area retrieved from said memory storage at a designated frame, to obtain the category of proximity,  
means of processing and dispatching to effect the switching of multiple tracks of animation to the said video-audio terminal based on the said determined category of proximity obtained by the comparison of the location of said playing instrument

15

transducer to animated target area and frame cueing from said memory storage,  
continuing means of processing and dispatching for instructing said video-audio terminal to re-wind multiple tracks of animation to another episode determined by means of random choice or player input at the termination of an episode.

2. Apparatus as set forth in claim 1 as a primary circuit, and a secondary circuit providing digital animation, digital graphics and audio scoring the game and related to the game-originated animation and player's actions, and controlling the progress of both primary and secondary circuits, comprising:

a second player input terminal providing means of system control for the said means of processing and dispatching,

permanent memory means providing a mass storage of digital graphics image data for a means to generate cartoon graphics for a video monitor display, said means of retrieval retrieving said digital graphics image data from said permanent memory means for storage in a random access memory means,

said permanent memory means providing storage of control data including cue commands, said cue commands comprising schedule commands for said processing and dispatching means,

said means of retrieval retrieving said control data from said permanent memory means for storage in said random access memory means, said permanent memory means storing audio and/or graphics data for a random access memory of object video staging,

said means of retrieval retrieving said audio and/or graphics data from said permanent memory means for said random access memory of object video staging,

said processing and dispatching means controlling the course of the digital animation,

a cue table memory means for storing cue commands from said processing and dispatching means,

a cueing unit repeatedly scanning said cue table memory means to get said cue commands, and to execute said cue commands,

said processing and dispatching means requesting successive blocks of control information from said means of retrieval for operations scheduling instructions for storage in said cue table memory means for use by said cueing unit, [ - ] said processing and dispatching means repeatedly updating said operations scheduling instructions as the cartoon progresses,

said cueing unit executing cue commands at the times therein specified by conveying to the said means to generate cartoon graphics blocks of binary coded data from said means of retrieval from storage in said random-access memory of video staging, to generate cartoon frames for display on said video monitor,

program memory cartridge means for providing individual game data to reinforce said audio and/or graphics data stored in said random-access memory of object video staging,

said means of retrieval obtaining digital audio data from said permanent memory means for said random-access memory of object video staging, to be sent to a means of converting digital data to analog form, and hence to a system speaker,

16

said processing and dispatching means controlling progress of the game involving said primary and secondary circuits using data from said program memory cartridge means, said permanent memory means obtained through said means of retrieval and said cueing unit, said means of processing and dispatching ordering said control information of the game for the primary circuit to be sent from said cueing unit to the video-audio input terminal.

3. Apparatus as set forth in claim 2 utilizing the video system and monitor circuits to measure the playing instrument action and position, comprising:

means providing the playing instrument position determination by analysis of the video monitor display's cathode ray beam sweep circuits or other display drive circuits using a pulse signal from a light detector at the playing instrument position, and pulse time to determine the position of the sweep and thereby define the relative relationship of the playing instrument within the image.

4. Apparatus as set forth in claim 2 utilizing the circuits of the video monitor to effect a strobe or cursor and to measure their coordinates of scene field location, comprising:

means created by both the height N/S and the width E/W movement of the playing instrument activating respective circuits creating and controlling movements of a strobe or cursor aimed at the animation target position, and activating analysis providing a digital readout of the cursor/strobe position for comparison to the coordinates of the position of the animation target, and embodiment of a rifle mounted in an adjustable socket joint activating said N/S and E/W circuits to effect a strobe or cursor at the position on the image of the video monitor corresponding to the position aimed at by the rifle.

5. Apparatus as set forth in claim 2 including means enabling the player to operate a second playing instrument in a simulation of an athletic game, and repeatedly hit designated mid-air projected positions of displayed foreshortened perspective animated actions, and effect returns of said animated actions to the video monitor,

wherein said player input terminal includes the second playing instrument containing a transducer, a configuration of single-point microphones before the face of a television monitor, and providing means for producing digital signals representative of the coordinates of the mid-air location of said playing instrument transducer with respect to the said microphone positions,

wherein said means of entering a table of values enters the mid-air location,

and wherein said means of retrieval retrieves the mid-air location,

and wherein said means of comparing and matching compares and matches the mid-air location.

6. Apparatus as set forth in claim 5 including means providing a video game based on the speed of the playing instrument and the resulting effect of such speed measurements in the displayed animation action, comprising:

said player input terminal including the second playing instrument containing a transducer controlled to send out pulses at fixed time intervals, and one or more microphones arranged to permit measurements of the components of playing instrument motion,

means of measuring the frequency of received pulses from said playing instrument transducer by said microphone and by other microphone installations, to provide means for determining the Doppler Effect, and thus determining the velocity of said playing instrument,

means to combine the velocity and mid-air location to improve the information used by the system logic for track switching and scoring.

7. Apparatus as set forth in claim 5 providing a portable playing instrument,

wherein said player input terminal includes a portable playing instrument containing a power source, a controlled time base, a pulse generator synchronized by the time base, and an ultrasonic transducer.

8. Apparatus as set forth in claim 7 providing means for a game during which the player constantly attempts to evade mid-air projected foreshortened perspective animation action,

wherein said player input terminal includes two portable transducer units, one as a playing instrument component, and the other attached to the player, wherein said means for producing digital signals representative of the coordinates produces signals of the location of each of two portable transducer units with respect to a configuration of microphones.

9. Apparatus as set forth in claim 7 incorporating alternate means of video-audio recording and display, comprising:

means to utilize motion picture recording for large screen projection,

and means to utilize video recording of pictures as images and/or digital representations.

10. An interactive video game system comprising: moveable user input means disposed in front of a display, means said moveable user input means manipulable by a user and including transmitter means for generating and transmitting position signals representative of the location of said moveable user input means with respect to said display means;

storage means for storing and playing back data representing a plurality of randomly selectable action sequences of visual images, said storage means including storage media for storing said data in multiple tracks, each said action sequence comprising at least one coded frame representative of said visual images, each said coded frame including coded location coordinates associated with said visual images, said storage means responsive to selection signals for switching between said multiple tracks for selection and playback of action sequences for display;

said display means coupled to said storage means for displaying selected ones of said plurality of action sequences;

memory means for storing a table of values, said values representing digital coordinates corresponding to said coded location coordinates of said coded frames in a selected action sequence, said table of values entered in said memory means at the commencement of display of a selected action sequence;

control means coupled to said storage means, said display means and said memory means, including receiving means for receiving said transmitted position signals, said control means for determining the position of said moveable user input means and comparing said position to said digital coordinates associated with

a selected visual image of a currently displayed action sequence for determining the position of said moveable user input means with respect to the position of at least one currently displayed selected visual image, said control means responsive to said moveable user input means respective position for selecting a next successive one of said plurality of action sequences for display.

11. An interactive video game system as in claim 10 wherein said transmitter means comprises a transducer and a signal generator, said signal generator generating said position signal, said position signal being radiated by said transducer.

12. An interactive video game system as in claim 11 wherein said transducer comprises an ultrasonic transducer and said signal generator comprises a pulse generator for generating an ultrasonic pulse signal, said ultrasonic transducer radiating a repetitive ultrasonic pulse signal.

13. An interactive video game system as in claim 12 wherein said receiving means comprises at least two transducers, said transducers disposed on said display means in spaced-apart relationship adjacent said displayed action sequences, each of said transducers receiving said transmitted position signal and providing location signals indicative of said moveable user input means position with respect to said transducers.

14. An interactive video game system as in claim 13 wherein said moveable user input means comprises a player instrument allowing a user to contact a display screen of said display means indicating a user response to a currently displayed action sequence, said player instrument coupled to said control means.

15. An interactive video game system as in claim 14 wherein said player instrument includes switch means disposed at a contact tip thereof, said switch means providing a signal indicating when said contact tip is in touch contact with said display screen.

16. An interactive video game system as in claim 12 wherein said receiving means comprises a plurality of transducers disposed in spaced apart-position, each of said transducers receiving said transmitted position signal and providing location signal indicative of said moveable user input means position with respect to said transducers.

17. An interactive video game system as in claim 16 wherein at least one of said plurality of transducers is mounted on said display means adjacent said displayed action sequences.

18. An interactive video game system as in claim 16 wherein said moveable user input means comprises a player instrument adapted for user movement responsive to said currently displayed action sequence.

19. An interactive video game system as in claim 18 wherein said player instrument comprises a baseball bat, said baseball bat including a transducer disposed at a predetermined impact point, said baseball bat further including a pulse generator coupled to said transducer, said transducer radiating a repetitive pulse signal, said receiving transducers receiving said repetitive pulse signal and providing location signals indicative of the position of said impact point with respect to said receiving transducers.

20. An interactive video game system as in claim 19 wherein said baseball bat is coupled to said control means, said baseball bat adapted for swinging motion provided by a user responsive to said currently displayed action sequence, said selected visual image comprising a pitched baseball, said controller means selecting a next successive one of said action sequences for display by said display

*means in response to said impact point position with respect to said pitched baseball image at a predetermined point in space during the swing of said baseball bat.*

21. An interactive video game system as in claim 10 wherein said control means includes selection means responsive to a user input for selecting a desired action sequence to be displayed following the conclusion of a current play episode, said play episode comprising at least one of said plurality of action sequences.

22. An interactive video game system as in claim 10 wherein said control means includes random selection means for randomly selecting next action sequence to be displayed following the conclusion of a current play episode, said play episode comprising at least one of said plurality of action sequences.

23. An interactive video game system as in claim 10 wherein said randomly selectable action sequences of visual images comprise prerecorded animated motion picture segments, each said segment comprising coded frames of visual images for track switching, said segments stored in 20 multiple tracks of said storage media.

24. An interactive video game system as in claim 23 wherein said motion picture segments are arranged in related groups forming game episodes, each of said groups comprising at least one of said motion picture elements, 25 said game episodes stored on multiple tracks on video tape media.

25. An interactive video game system as in claim 24 wherein said storage means includes tape reading means coupled to said control means for reading said video tape media for display and rewinding said video tape, said tape reading means responsive to control signals generated by said control means for rewinds of said videotape to play selected motion picture segments or game episodes.

26. An interactive video game system as in claim 25 wherein said control means generates said control signals providing for random selection of a game episode to be displayed at the conclusion of a currently displayed game episode in accordance with a random number device.

27. An interactive video game system as in claim 25 wherein said control means generates said control signals for selection of a desired game episode in response to a user input.

28. An interactive video game system as in claim 23 wherein said storage media comprises a video disc.

29. An interactive video game system as in claim 10 wherein said randomly selectable action sequences of visual images includes camera-originated animated motion picture segments and computer generated graphics.

30. An interactive video game system as in claim 29 wherein said randomly selectable action sequences of visual

*images are stored in digital form in a read-only-memory means.*

31. An interactive video game system as in claim 30, wherein said read-only-memory means comprises a plug-in game cartridge.

32. A method of simulating an active game in which a player participates comprising the steps of:

storing data in coded frames on multiple tracks defined in storage media, said data representing a plurality of randomly selectable action sequences of visual images; assigning coded location coordinates of selected visual images associated with each said coded frame;

storing a table of values, said values representing digital coordinates corresponding to said coded location coordinates of said selected visual images;

displaying selected ones of said plurality of action sequences on a display means, said display means for displaying selected ones of said plurality of action sequences, said table of values entered in a memory means at the commencement of display of a selected action sequence;

providing a moveable input means disposed in front of said display means, said moveable input means adapted for user manipulation for generating position signals representative of the location of said moveable input means in response to a currently displayed action sequence;

generating and transmitting position signals representing the position of said moveable input means; receiving said transmitted position signals;

comparing the position of said moveable input means with said digital coordinates associated with at least one of said visual images to determine the position of said moveable input means with respect to the position of a currently displayed selected visual image;

selecting a next successive one of said action sequences for display by said display means in response to the location of said moveable input means with respect to said currently displayed selected image; and switching between said multiple tracks for selection and display of said next successive action sequence.

33. The method of claim 32 including the additional steps of:

forming game episodes comprising at least one of said plurality of action sequences; and randomly selecting a next successive game sequence to be displayed at the conclusion of a currently displayed game sequence in accordance with a random number generator.

\* \* \* \* \*

**United States Patent [19]**  
**Rodesch**

[11] Patent Number: 4,475,132  
[45] Date of Patent: Oct. 2, 1984

[54] INTERACTIVE VIDEO DISC SYSTEMS

[76] Inventor: Dale F. Rodesch, 13020 Broili Dr.,  
Reno, Nev. 89511

[21] Appl. No.: 341,964

[22] Filed: Jan. 22, 1982

[51] Int. Cl.<sup>3</sup> H04N 5/76

[52] U.S. Cl. 358/342; 273/DIG. 28;

434/43

[58] Field of Search 358/310, 311, 312, 335,  
358/342, 22; 360/10.1, 10.2, 14.1, 14.2, 14.3;  
273/DIG. 28, 313; 364/521; 434/43, 307, 323

[56] References Cited

U.S. PATENT DOCUMENTS

- |           |         |                  |       |           |
|-----------|---------|------------------|-------|-----------|
| 3,732,362 | 5/1973  | Kinjo            | ..... | 360/10.1  |
| 4,094,013 | 6/1978  | Hill et al.      | ..... | 358/342   |
| 4,139,868 | 2/1979  | Eisenberg et al. | ..... | 358/335   |
| 4,142,209 | 2/1979  | Hedlund et al.   | ..... | 358/342   |
| 4,305,131 | 12/1981 | Best             | ..... | 364/521 X |
| 4,360,876 | 11/1982 | Girault et al.   | ..... | 364/521   |
| 4,361,849 | 11/1982 | Bolger           | ..... | 358/312   |
| 4,402,018 | 8/1983  | Wada et al.      | ..... | 358/342   |
| 4,423,497 | 12/1983 | Sugiyama et al.  | ..... | 358/312   |

FOREIGN PATENT DOCUMENTS

2649353 5/1977 Fed. Rep. of Germany ..... 358/335

OTHER PUBLICATIONS

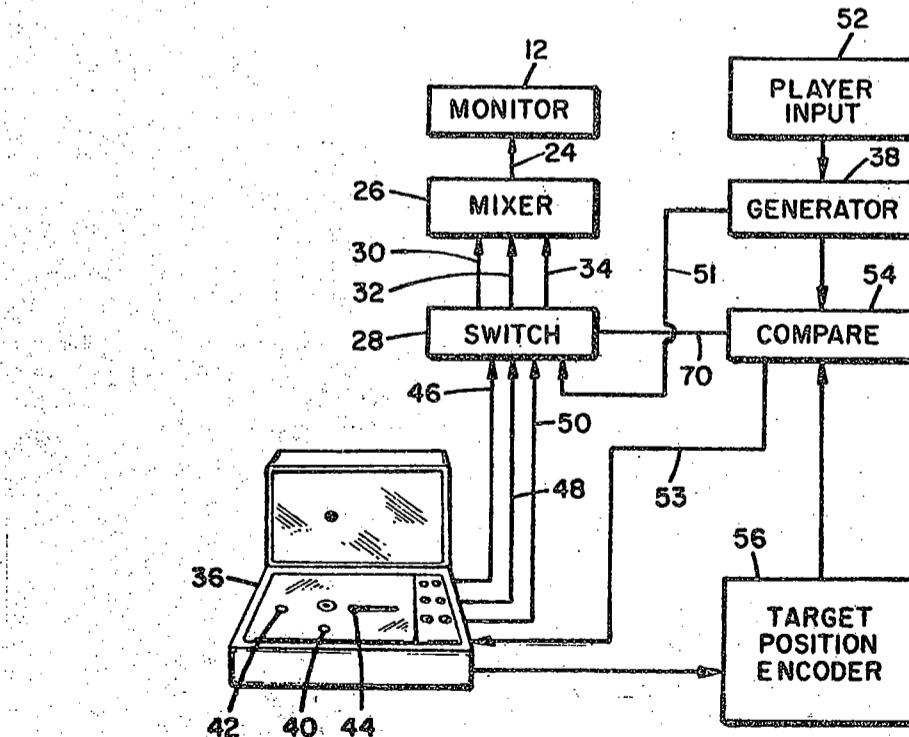
Michel Mathieu, "A Random Access System Adapted  
for the Optical Video-disc: Its Impact on Information  
Retrieval", pp. 80-83.

Primary Examiner—Donald McElheny, Jr.  
Attorney, Agent, or Firm—Grover A. Frater

[57] ABSTRACT

An interactive video disc system employs discs in  
which video presentations are arranged in parallel in a  
single spiraled track by being arranged in adjacent  
turns, or portions of turns, of the spiral. Additional  
capacity for interaction is achieved by the use of multi-  
ple heads and switching from one presentation to an-  
other during the formation of picture lines. Different  
scenes of video information are arranged in an inter-  
leaved format on the disc. This format permits special  
effects involving superimposition of the different scene  
data to obtain a combined smooth flowing scene with-  
out any disturbance in display of the combined scene.

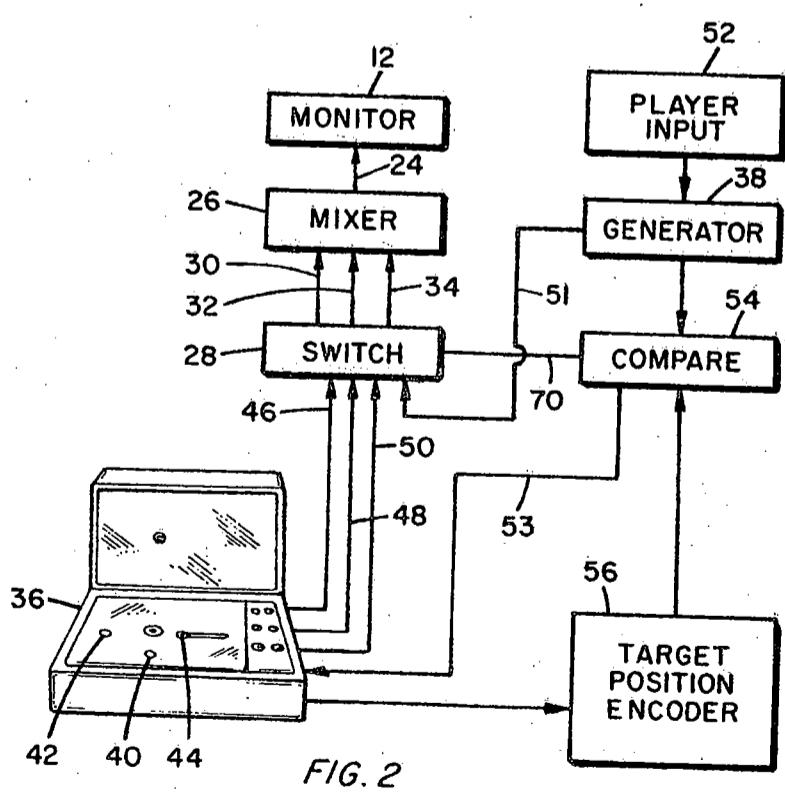
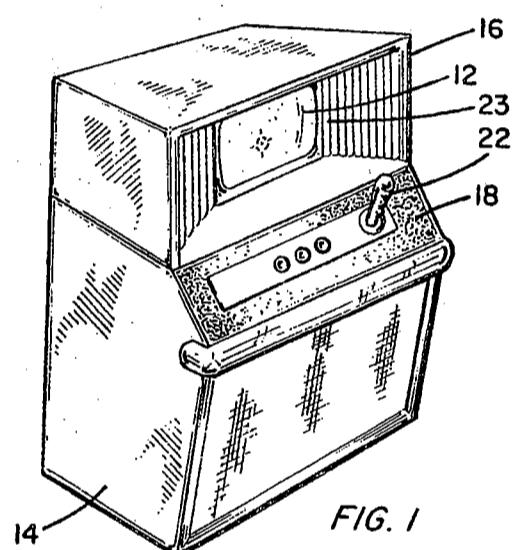
14 Claims, 4 Drawing Figures



U.S. Patent Oct. 2, 1984

Sheet 1 of 3

4,475,132



U.S. Patent Oct. 2, 1984

Sheet 2 of 3

4,475,132

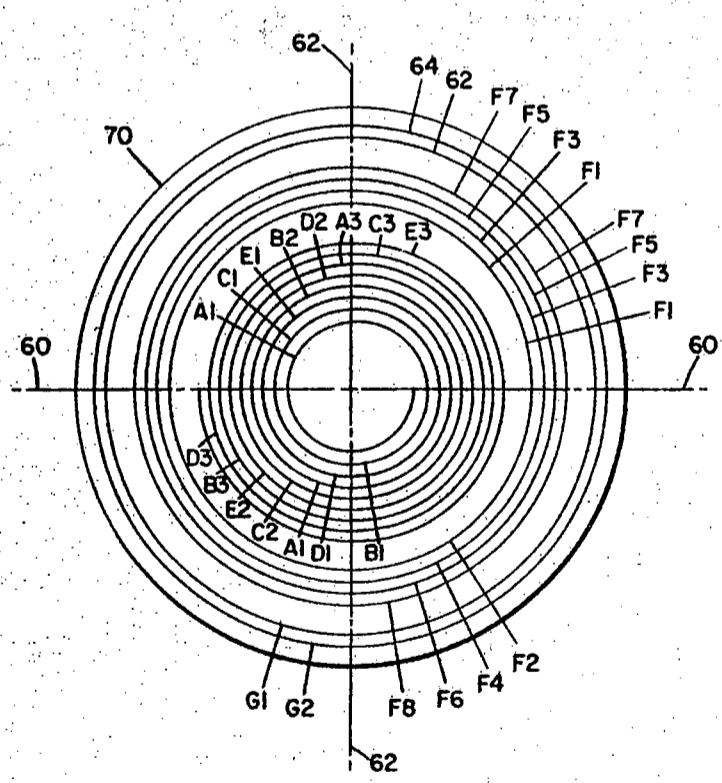
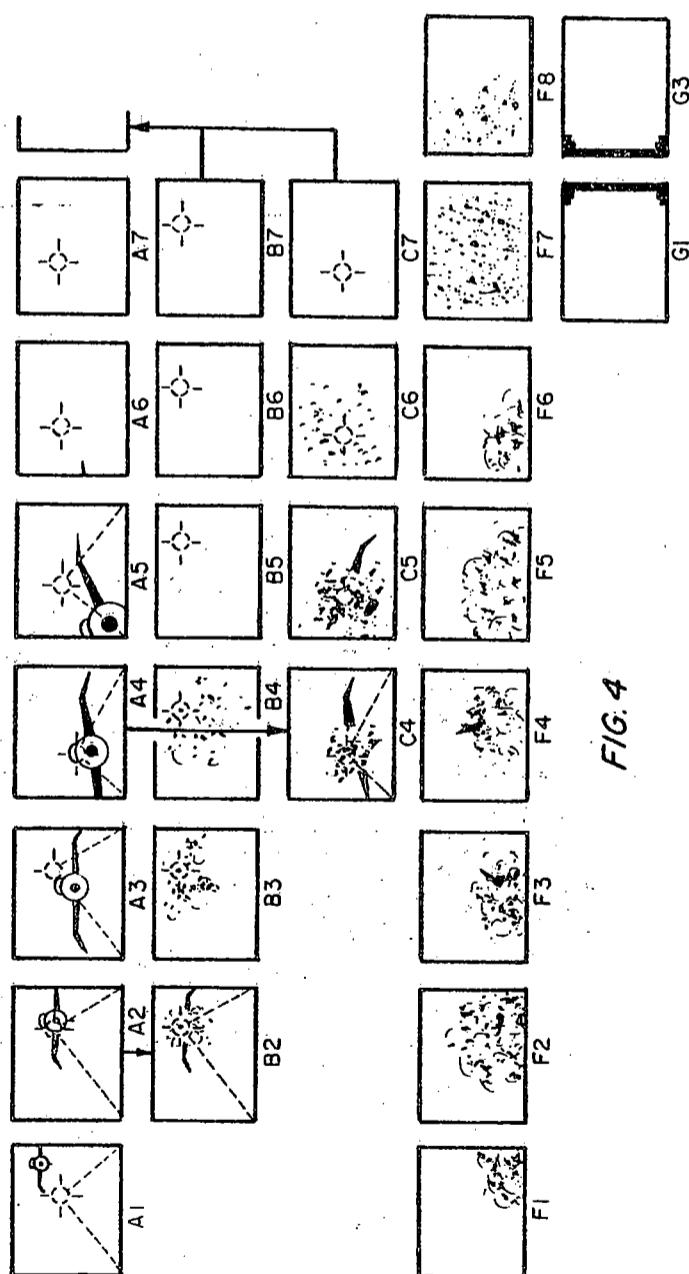


FIG. 3

U.S. Patent Oct. 2, 1984

Sheet 3 of 3

4,475,132



## INTERACTIVE VIDEO DISC SYSTEMS

### TECHNICAL FIELD

This invention relates to interactive video disc systems and to video discs for such systems.

### BACKGROUND ART

An interactive system is one which presents to a user a program taken from a number of predefined response programs selected on the basis of the user's inputs. Examples are found in interviewing machines, teaching machines, and interactive games.

Prior systems usually employed prerecorded questions and responses, each on a different magnetic tape which were selected and played back in response to simple user inputs. The number of different responses was limited by the number of magnetic tape players that could be packaged in a system.

In some cases the responses and questions were recorded sequentially in segments on a single tape and the tape was run in either direction to search for a particular segment. That scheme involves variable delays between the time of input and response because search time is a function of the position which the segment to be played occupied along the length of the tape.

In the case of video tape, the size and cost of playback units was sufficiently great to make the use of multiple players prohibitive. Most video output machines employ a single tape or disc, or a single film, and they resort to searching for the required segments.

The time delay can not be tolerated in the case of video games. The game machines abandoned analog techniques and employed digital information instead. The display was limited to "stick men," to vertical or horizontal movement on the display screen, to localized increases in brilliance, and other simplistic representations. Early games utilized a limited number of responses stored in semi-conductor memory devices which were selected by simple digital logic units in response to a limited number of permissible user inputs. Addition of central computing units and more memory made increased variety and complexity possible, but it did not overcome the limitation to "stick men" and other simplistic displays.

Prior to this invention true interaction, using analog video pictures, was not available. It has been made available by the invention in a number of different forms.

### DISCLOSURE OF INVENTION

The invention employs one or more of several basic techniques and apparatus to implement their use. In one of those techniques, two or more video presentations are arranged in parallel tracks on a storage medium. In the preferred mode, those presentations or scenes take the form of successions of video picture frame information which, in some tracks, may be repeated several times over the length of the track. The information is recovered using multiple reading units or by selectively shifting the track and reading unit to "read" the information in a selected track.

Another technique involves the use of endless or loop tracks on which video picture frame information is formed so that the same frame or sequence of frames may be displayed repetitively. When the information is stored on a disc, or cylinder, or other medium, the several regions of which may be accessed simulta-

neously, it is not necessary to use endless or loop tracks, but the technique is otherwise the same.

Combined with the parallel track arrangement described above, or the digital techniques which have been developed in the video game art, or both, the endless loop technique permits interactive programs that are more versatile, interesting, realistic and persuasive than has been possible in the past.

These several techniques can be practiced with film or tape, but it is preferred to employ video discs which are read using light. Reading with laser beams is preferred. The information recovery apparatus, or "head," includes a low wattage laser, a laser light sensor and an optical system which includes a mirror moveable to direct the laser output beam to a selected one of a number of proximately placed turns of the information track.

The conventional video disc is formed with a spiralled track extending for as much as 54,000 turns in a twelve inch disc. The recorded information is contained in a series of conformations, or pits, which extend along and form the track. The information which constitutes one video picture frame is contained in segments of the track which are separated by short segments in which synchronizing signals are recorded. There is no technical reason why several tracks cannot be formed and spiralled in parallel, or as a series of concentric circles, or why single and multiple track spirals and circular tracks cannot all be used in different areas of the same disc to be read by different heads. All are useful in the invention. What form is preferred will depend upon the application for the system and the form of interaction and display to be provided.

At present, the single spiralled track is preferred because that has become the video disc industry standard, and apparatus for making and using single spiral discs is readily available.

There are two standard video disc machine forms and modes of operation. In the first, the disc is rotated at uniform angular velocity, and in the second, the disc is rotated such that the information track passes the read head at uniform linear velocity. In the first case, the synchronizing information between the fields lies on common radial lines. Two fields, which together form a frame, are recorded in each turn of the spiral so that each field begins and ends on one of two diametrically positioned radial lines which together form a diameter line of the disc. The heads used with such discs include a means for adjusting the optical system to permit reading the information in any of two or more adjacent turns of the spiral. The adjustment of the optical system can be changed to switch from one turn to another during detection of the synchronizing segments of the track. That facility is used to provide the "stop frame," reverse, a fast forward, and fast reverse - capabilities of standard video disc play back machines.

The invention can, and in the preferred form does, utilize that turn selection facility. The spiralled track is viewed as a succession of adjacent but individual turns which may be selected one from the others at each radial line that marks the ends and beginnings of fields. Video picture frames from different successions of pictures are recorded on adjacent turns of the track. Using the constant angular velocity format, two or more fields, each taken from a different sequence of fields, are recorded on at least some of the turns of the spiralled track. The succeeding fields, or field groups, are re-

corded on an adjacent turn or a turn two or three (or even more) removed from the turn on which the preceding field, or field group, of the series was recorded. In each case, the succeeding field or group, begins on the radial line at which the preceding field, or group, ended. The read head must be moved from one turn to another to read the fields of the series. The movement is accomplished optically by track switching and by movement of the head itself over the surface of the disc. Switching from one sequence of frames, or visual presentation, to another presentation or sequence of frames is accomplished by optical switching between turns.

Optical switching combined with simultaneous and/or alternate reading of other track segments by additional heads provides a number of presentation alternatives that can make possible a very effective interactive system. The preferred system is one which offers maximum alternatives, and thus combines turn switching and multiple heads, some movable and some fixed. In addition, the preferred system also incorporates a means for displaying information from different video presentations in different areas of a single video frame.

The system is made interactive when the user can introduce any of a number of inputs and evoke an appropriate response from a number of available responses. The invention provides a means for accomplishing interaction which incorporates apparatus for utilizing the preferred techniques of the invention.

#### BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings:

FIG. 1 is a perspective view of a video game machine which embodies the preferred form of the invention;

FIG. 2 is a diagram, partly in schematic form and partly in block form, of the elements of the machine of FIG. 1;

FIG. 3 is a schematic representation of the video disc which forms a part of the preferred embodiment of the invention; and

FIG. 4 is a schematic representation of several series of video picture frame sequences that are recorded on the disc of FIG. 3.

#### DESCRIPTION OF THE PREFERRED EMBODIMENT

Which of the features of the invention are employed, and which structural arrangement best serves the purpose, will vary from application to application. For example, interactive teaching machines used to sell products or to teach bank patrons how to use automatic tellers may be configured quite differently, and certainly will be different from video game machines.

The best mode for practicing the invention is illustrated in the video game application. The most popular are the "space war" games, and that is what has been selected for illustration and description. The object in this version of the game is to shoot and destroy attacking fighter craft from a gun position on a space ship. What the player sees on a monitor screen is an analog video display in full color of a science fiction fighter combat and fighter attack such as might have originally been filmed or video taped as a segment of a motion picture or television show. In addition, the display includes a gun sight and a means for firing imaginary projectiles at the fighters on the screen. If the trigger is squeezed, the video program will continue uninterrupted unless, at the time the trigger was squeezed, the reticle was positioned on an attacking craft. In that

circumstance, the display changes and, in full analog color, the fighter explodes at the point at which it was struck. The debris from the explosion continues along the path of the craft and then disappears from view at the edge of the screen. Thereafter, the original program is returned to the screen not at the point at which it was interrupted, but at the point which represents an elapse of program time corresponding to the time that the alternate program was displayed.

FIG. 1 depicts the video game apparatus 10 in finished and preferred form. It comprises a video monitor 12 and a cabinet 14 on which the monitor 12 and its hood 16 are mounted. The sloping panel 18 of the cabinet is styled to simulate the control panel at a gun control station of a spaceship. A handle 22 is provided for directing the reticle which appears on the monitor screen during play and which moves as the handle moves. A trigger on the handle is squeezed by the player to "fire" the guns.

In this model the forward part of the monitor hood is styled to represent a gun port. Side panels, one numbered 23 which is visible, are spaced from the screen in small degree. They add to the realism of the display by masking the edges of the monitor screen when the player stands at the front center of the cabinet, and by shifting the field of view of the screen when the player leans from side to side with gun handle movement.

The cabinet 14 houses the elements, other than the monitor 12, which are included in FIG. 2. The monitor is a conventional unit of the kind used to display television pictures. Video and audio signals are input to the monitor 12 by line 24 from a conventional mixer 26 whose function is to add input signals from switch 28 for application together to the control electrode of the monitor tube. In this case there are three input lines, 30, 32 and 34, from the switch to the mixer. Depending upon switch operation, these lines pass video signals to the mixer from a moveable head and from one or the other of two fixed heads in player 36, and from a digital signal generator 38, respectively.

The video disc player is a standard, constant, angular velocity player modified to add two fixed pick-up heads 40 and 42. The head 44 is the standard moveable pickup which incorporates the track turn switching optics. Heads 40 and 42 also are standard units, and they incorporate the track turn switching feature.

Lines 46, 48 and 50 carry video signals to the switch from heads 40, 42 and 44, respectively. The line 51 furnishes video signals to the switch 28 from the digital signal generator 38.

The player input box 52 in FIG. 2 represents the "joy stick" and trigger switch common in video game practice and which, in this case, includes the handle 22 with its trigger switch. The handle controls the position on the monitor screen of a cross-hair and reticle which are generated by the digital signal generator 38. This generator is the same as those employed in many video shooting games. Just as in those games, squeezing of the player trigger generates signals that produce flashes of light on the monitor to indicate the direction the projectiles follow and to produce a flash of light on the monitor screen at the position of the cross-hairs and reticle.

The same signals that control reticle position on the screen are furnished to a comparator 54 whose output on line 53 determines which of five video program displays, A, B, C, D, or E, is selected by the head 44 for delivery to the switch on line 50. The line 53 signal also dictates which of eight frames will form an F sequence

selected by head 42, and which of two G sequences is selected by head 40. The criteria for display selection is explained below. The system elements which accomplish selection are the conventional digital comparator 54 which matches target position information received from head 44 through target position encoder 56 with digital information received from the reticle position generator 38.

The encoder 56 is a memory unit in which is stored, for each group of frames in the A program sequence, 10 the position on the screen of the fighter ship targets. The A sequence is the presentation of the attacking fighter ships. For the most, those ships appear at the edge of the screen and fly across and out of view at a different edge. In this model the screen is divided by several horizontal and several vertical imaginary lines into area zones. The video presentations B, C, D, and E depict the disintegration of a fighter ship in four different ones of those zones. Selected ones of the F frames are displayed in series, and they depict a close-up, full screen explosion. They are presented only when a fighter is hit at close range. Presentations G3-G4 display what appears to be portions of the spaceship at the sides of the gun port which come into view at the right and the left of the screen when the reticle is moved to the far left and to the far right, respectively.

Portions of the frame sequences that form video presentations A, B and C are depicted in FIG. 4 along with the F frames which depict a close-up disintegration. Also shown are two of the G frames which depict the 30 right and left edges of the gun port, respectively. The D and E sequences are not shown, but are similar to sequences B and C. Except that the G frames, when shown, are shown simultaneously with the A frames, only one of the A, B, C, D, E and F sequences are 35 displayed at any one time.

It is standard practice in the industry to number each frame and to include the number in the video information that forms a part of each frame. The coordinates of each target in each frame of the A sequence are placed in memory in the target position encoder 56 except that it is not necessary to insert coordinates of target positions in which it is decided that no "hit" will be allowed. Along with the coordinates for the target, there is stored a code number to indicate whether the target 40 lies at medium range or close range in each frame. If the target is struck at medium range, then the disintegration of the target will be depicted by one of the sequences B, C or D, whichever shows an explosion closest to the zone occupied by the target when hit. Nearness of a target is represented by increased size. When a target is hit at close range, its disintegration is depicted by a frame selected from the F frames. Frames F1 and F2 depict an explosion at the lower right of the screen, whereas frames F5 and F6 depict an explosion at the lower left. These frames are arranged in two predefined sequences with the other F frames. The sequence is predefined on the basis of target position when hit.

These several programs can be recorded on storage media of any of several types. However, in the preferred embodiment of the invention, the video and related audio presentations are stored on a video disc for use in a laser disc player like the player 36 of FIG. 2. A typical video disc arranged for reading with a laser head is twelve inches in diameter and carries audio and video information in a 54,000 turn spiral track formed by conformations in the disc surface. A modified disc is shown in FIG. 3. The central region of the disc is

formed with a spiral track some of the turns of which are represented schematically and are identified by letters A through E. Surrounding the spiraled track, near the edge of the disc, is a pair of circular tracks identified by the letter G. Between the G tracks and the spiraled track is another group of four circular tracks identified by the letter F. The edge of the disc is identified by the numeral 70. The disc is arranged for rotation at constant angular velocity and the video information is contained in semi-circular segments of the track which begin and end, in the case of the spiraled track and the four F tracks, on the diameter line 60—60. Synchronizing information occurs in each turn along that diameter line on each side of the disc center. It should be noted that other numbers of segments may be used, but the two segment arrangement is now preferred.

The head 40 of player 36 in FIG. 2 lies at right angles to the line on which heads 42 and 44 are placed so that frame change for the video frames represented in the G track begin and end on the diameter line 62—62 which is placed at right angles to the diameter line 60—60.

Information contained in all of the track segments F1 through F8 are depicted in the "F" video frames shown in FIG. 4. The information in segments G1 and G2 is the same, and the information in segments G3 and G4 is the same, so only one of each pair is depicted in FIG. 4. In the spiraled track, the frames of the several sequences A through E are arranged in parallel in special fashion.

Examination of the spiraled track in FIG. 3 will show that each turn of the track contains two sets of video frame information, each taken from a different one of the several presentations A through E. Beginning at the center of the spiral, the first semi-circular segment, the one below the diameter line 60—60, contains no video information. The other half of that turn, the upper semi-circular segment, contains the frame A1 information. The first half of the second turn contains the information for frame B1, and the second half of the second turn includes the information for frame C1. The first half of the third turn contains the frame D1 information, and the last half of the third turn contains the E1 frame information. The first half of the fourth turn contains the A2 information. The arrangement is an example of what may be called "interleaving" of video program information. As the disc rotates counterclockwise in FIG. 3, a laser beam carried by head 44 is directed at the conformations of the track, and is reflected back to a sensor which converts the information contained in the character of the conformations of the track into a video and audio signal. As the disc rotates, the head 44 is moved from a position under the beginning of segment A1 outwardly toward the outer edge of the spiral. The head moves at a rate which ensures that it follows the spiraled track. However, the optical system within the laser read head is adjustable by a signal applied by line 53 in FIG. 2 so that it will read the track conformations in any one of the five turns of the spiral that are immediately under the head.

Let it be supposed that the disc of FIG. 3 is placed in player 36 such that the turns of the track that contain the video information corresponding to frames A1, C1, E1, B2 and D2 overlie head 44. By appropriate adjustment of the signal on line 53 of FIG. 2, the head can be adjusted to read the information in any one of those five frame segments.

If the player does not depress the trigger, no target will be hit, and only the A presentation will be dis-

played on the video monitor. In that case, following a half rotation of the disc during which frame A was read, a signal on line 53 would adjust the optics so that the laser beam would be moved from the end of the first turn of the track to the beginning of the fourth turn of the track. Instead of reading the B1 frame at the beginning of the second turn, it would read the A2 frame at the beginning of the fourth turn. Following the next 180 degrees of rotation of the disc, at the end of the first segment of the fourth turn, the optical system would be transferred to the beginning of the A3 frame in the sixth turn. With each half turn the optical system would be moved from the end of the current half turn, past the preceding turn, to the beginning of the next succeeding turn. That process would continue as the whole of the A presentation was made to appear on the monitor.

It can be demonstrated that the head will have to be moved along the spiral at a rate greater than the normal rate by a factor which corresponds to the number of presentations that are available in the spiralled track.

Next, let it be supposed that the player squeezes his gun trigger during the course of the A presentation. The coordinates of the reticle generated in generator 38 are compared in comparator 54 with the coordinates supplied by the target position in encoder 56 for the A sequence frame currently being displayed. If the coordinates do not match, the signal on line 53 remains unchanged, and the head 48 continues to read the A sequence of frames. If, however, the coordinates of the reticle and the coordinates of the target match, then the signal on line 53 will be changed in accordance with the zone in which the target and reticle were matched, and in accordance with the distance signal supplied by the target position encoder 56. If the target was at medium distance, and in the zone corresponding to the B presentation, the signal on line 53 will cause the optics in the head 44 to shift to the turn in which the B presentation appears at the next synchronizing signal. If the target was struck when in the zone, and at the distance corresponding to the C position, the signal on line 53 would have caused the head 44 optics to shift the laser to the turn containing the C frame.

In the case of a hit when the target is close, the signal provided by the compare unit on line 70 to the switch 28 will switch off the video line 50 and will switch on the video line 48 so that it is the output of head 42 which reaches the mixer and ultimately the monitor 12. The signal on line 53 determines whether the head selects the F frames to show a left corner or a right corner disintegration.

When the reticle positioning handles are pushed fully to the right or left, the generator 38 supplies a signal to the switch 28 which permits the output of head 40 to pass to the mixer for addition to the information supplied by head 44 so that either the G1 and G2 frames, or the G3 and G4 frames, are supplied to the monitor along with the A presentation supplied by head 44.

An internal arrangement permits blanking the A presentation signal during the course of each line sweep in the picture tube, and presentation of an alternate video signal for the remainder of each line. That feature permits the showing of a display one area of which is taken from one sequence of frames and another area of which is taken from a different sequence.

I claim:

1. The method of manufacturing a video disc for use in an interactive system with an information recovery element which is capable of selectively recovering in-

formation from any one of two or more proximate positioned turns of a multiple turn track, which method comprises the steps of:

making a multiple turn track of video information by embodying video information in the disc surface such that each video field group begins and ends on a radial line which is common to the beginning or end of another video field group, and embodying said information to conform to the video information in at least two portions of video field successions each of which successions forms a separate video scene, and embodying said information such that portions of the successions are alternated in series along said track and are arranged such that the end of each portion of the succession lies on the same radial line as does the beginning of the next portion of the same succession but in a different turn of the multiple turn track.

2. The invention defined in claim 1 in which the scene represented by one of said two successions of fields contains portions that are repeated in an interval during which the scene represented by the other of said successions does not contain repeated portions.

3. A rotatable video disc formed with a multiple turn track having video information formed along the track; said video information comprising a plurality of successions of video picture fields, the fields of each succession together constituting a representation of related information;

said video information being arranged along the track in a series of sequences, each sequence being formed as a series of portions of said successions taken one portion from each of said successions, the order in which portions are taken from said successions being the same in each sequence.

4. The invention defined in claim 3 in which each of said portions of successions constitutes one or more of said fields.

5. The invention defined in claim 4 in which the number of fields taken from any one of said successions is the same in each sequence of said series of sequences.

6. A rotatable video disk formed with a multiple turn track having video information embodied along the track;

said video information comprising at least two successions of video picture fields, each succession representing a different video program; each of said successions of fields being divided into at least two portions, each portion constituting at least one field;

said portions being alternated along the length of the track such that a first portion of a first succession of said plurality of successions is followed by a first portion of a second succession of said plurality of successions which is followed by a second portion of said first succession which is followed by a second portion of said second succession;

information reading means for recovering the video information from the track one field at a time; and control means for selectively causing said information reading means to recover the information in the successive fields of a selected one of said successions by recovering the information in a segment of one turn of the track and thereafter recovering the information in a succeeding segment of an adjacent turn.

7. The invention defined in claim 6 in which said information reading means comprises means for simul-

taneously recovering information one field at a time from two segments of said track at spaced points on said disc.

8. The invention defined in claim 6 in which said conformation reading means comprises at least two spaced information readers at least one of which is responsive to control by said control means.

9. The invention defined in claim 6 which further comprises a video display device connected to said information reading means and operative to display information recovered by said information reading means.

10. The invention defined in claim 9 which further comprises means for furnishing to said video display device, for simultaneous visual display with information furnished by said information reading means, video information derived from a user controlled source other than said information reading means.

11. The invention defined in claim 9 which further comprises means for furnishing to said video monitor, for display during the period of selected fields, information derived from more than one segment of the track which segments contain fields from different presentations.

12. The invention defined in claim 6 in which each portion of a succession occupies that segment of a turn which lies on one side of a diameter line of said disc whereby each succeeding portion of any of said successions of presentations occupies a segment which begins in a different turn at the diametric line on the side of the center at which the segment occupied by the preceding portion of the succession ended.

13. The invention defined in claim 6 in which the first portion of said first succession of frames and said first portion of said second succession of frames are each constituted by a like number of frames; and said second portion of said first succession of frames and said second portion of said second succession of frames are each constituted by a like number of frames.

14. The invention defined in claim 13 in which said control means is capable of causing the information reading means to read the portions of information so interleaved in said tracks such that a number of the portions of one succession of said plurality are read in series followed by a reading of a number of the portions of another succession of said plurality in series.

\* \* \* \* \*

# United States Patent [19]

Perkins

[11] Patent Number: 4,847,690  
 [45] Date of Patent: Jul. 11, 1989

- [54] INTERLEAVED VIDEO SYSTEM, METHOD AND APPARATUS  
 [75] Inventor: John D. Perkins, Cary, Ill.  
 [73] Assignee: Isix, Inc., Foster City, Calif.  
 [21] Appl. No.: 16,670  
 [22] Filed: Feb. 19, 1987  
 [51] Int. Cl. H04N 7/04  
 [52] U.S. Cl. 358/143; 358/142;  
 358/181  
 [58] Field of Search 358/145, 143, 147, 181,  
 358/185, 160, 88, 89, 142

[56] References Cited

U.S. PATENT DOCUMENTS

- 2,681,383 6/1954 Loe .  
 3,051,777 8/1962 Lemelson .  
 3,061,669 10/1962 Leek .  
 3,202,764 8/1965 Adams et al .  
 3,391,247 7/1968 Frohbach .  
 3,488,435 1/1970 Eilenberger .  
 3,493,674 2/1970 Houghton .  
 3,580,998 5/1971 Hammond et al. .... 358/181  
 3,586,767 6/1971 Morchand .  
 3,637,926 1/1972 Morchand .  
 3,697,675 10/1972 Beard et al. .  
 3,700,793 10/1972 Borsuk et al. .  
 3,725,571 4/1973 Justice .  
 3,740,465 6/1973 Dorsey .  
 3,743,767 7/1973 Bitzer et al. .  
 3,745,242 7/1973 Justice .  
 3,746,780 7/1973 Sletten et al. .  
 3,751,595 8/1973 Moses .  
 3,889,063 6/1975 Slavin .  
 3,896,487 7/1975 Tesler ..... 358/3  
 3,902,007 8/1975 Justice ..... 358/145 X  
 3,991,266 11/1976 Baer .  
 4,027,333 5/1977 Kaiser et al. .... 358/146  
 4,052,719 10/1977 Hutt et al. .... 340/324 AD  
 4,097,692 6/1978 Felix .  
 4,115,662 7/1978 Guinet et al. .  
 4,161,786 7/1979 Hopkins et al. .... 364/900  
 4,170,832 10/1979 Zimmerman .  
 4,199,656 4/1980 Saylor .  
 4,205,343 5/1980 Barrett ..... 358/147

- 4,233,628 11/1980 Ciciora ..... 358/147  
 4,266,240 5/1981 Levy ..... 358/3  
 4,278,993 7/1981 Suzuki ..... 358/181 X  
 4,287,528 9/1981 Levy ..... 358/3  
 4,393,404 7/1983 Cox et al. ..... 358/147  
 4,419,920 12/1983 Ohe ..... 358/319 X  
 4,467,356 8/1984 McCoy ..... 358/146  
 4,484,328 11/1984 Schlafly ..... 370/85  
 4,510,520 4/1985 Parker et al. ..... 358/12  
 4,514,760 4/1985 Balaban et al. ..... 358/143  
 4,532,547 7/1985 Bennett ..... 358/181 X  
 4,535,356 8/1985 Nakagawa et al. ..... 358/145  
 4,593,318 6/1986 Eng et al. ..... 358/142  
 4,656,512 4/1987 Devino ..... 358/82 X  
 4,658,291 4/1987 Morishita ..... 358/88 X  
 4,672,434 6/1987 Suzuki et al. ..... 358/88

FOREIGN PATENT DOCUMENTS

- 0079426 7/1978 Japan .  
 1420106 1/1976 United Kingdom .

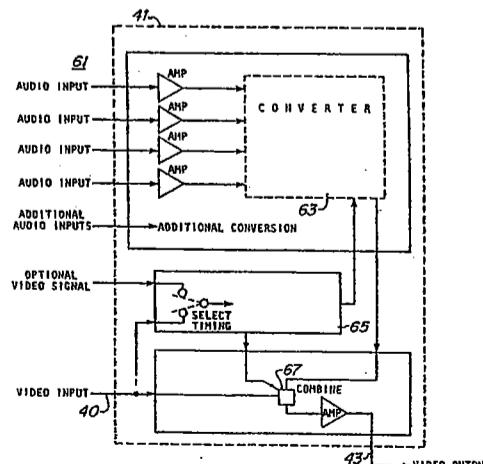
Primary Examiner—Howard W. Britton  
 Assistant Examiner—Victor R. Kostak  
 Attorney, Agent, or Firm—A. C. Smith

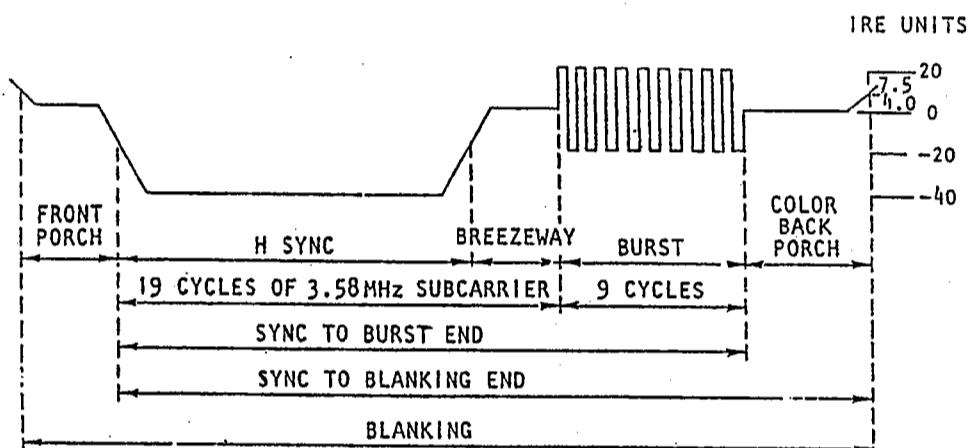
[57]

ABSTRACT

An improved system, apparatus and method are disclosed for interleaving selected video fields of multiple displayable programs to facilitate viewer selection of which programs to display. Selected video fields are stored and continuously displayed until a new, successive video field is selected to update the currently stored video field. In this manner, a sequence of rapidly updated, fixed or frozen video fields provides the appearance of continuous movement of displayed images. Audio signals are included in the video fields of displayable programs to provide audio signals in synchronism with selected video fields. The sequence of interleaved video fields may be assembled in real time or stored as a video tape recording, and the information needed to re-assemble correlated video fields of a selector displayable program from the sequence of interleaved video fields is included in the horizontal scans or traces of each video field in the sequence.

23 Claims, 19 Drawing Sheets





VIDEO CELL LEVELS IN IRE UNITS  
REFERENCE WHITE = 100 REFERENCE BLACK = 7.5  
BLANKING = 0 SYNC = -40 BURST AMPLITUDE = 40

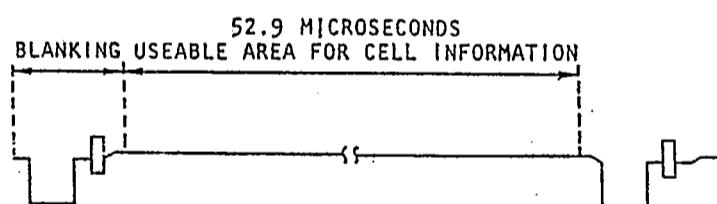


Figure 1 (b)

U.S. Patent

Jul. 11, 1989

Sheet 2 of 19

4,847,690

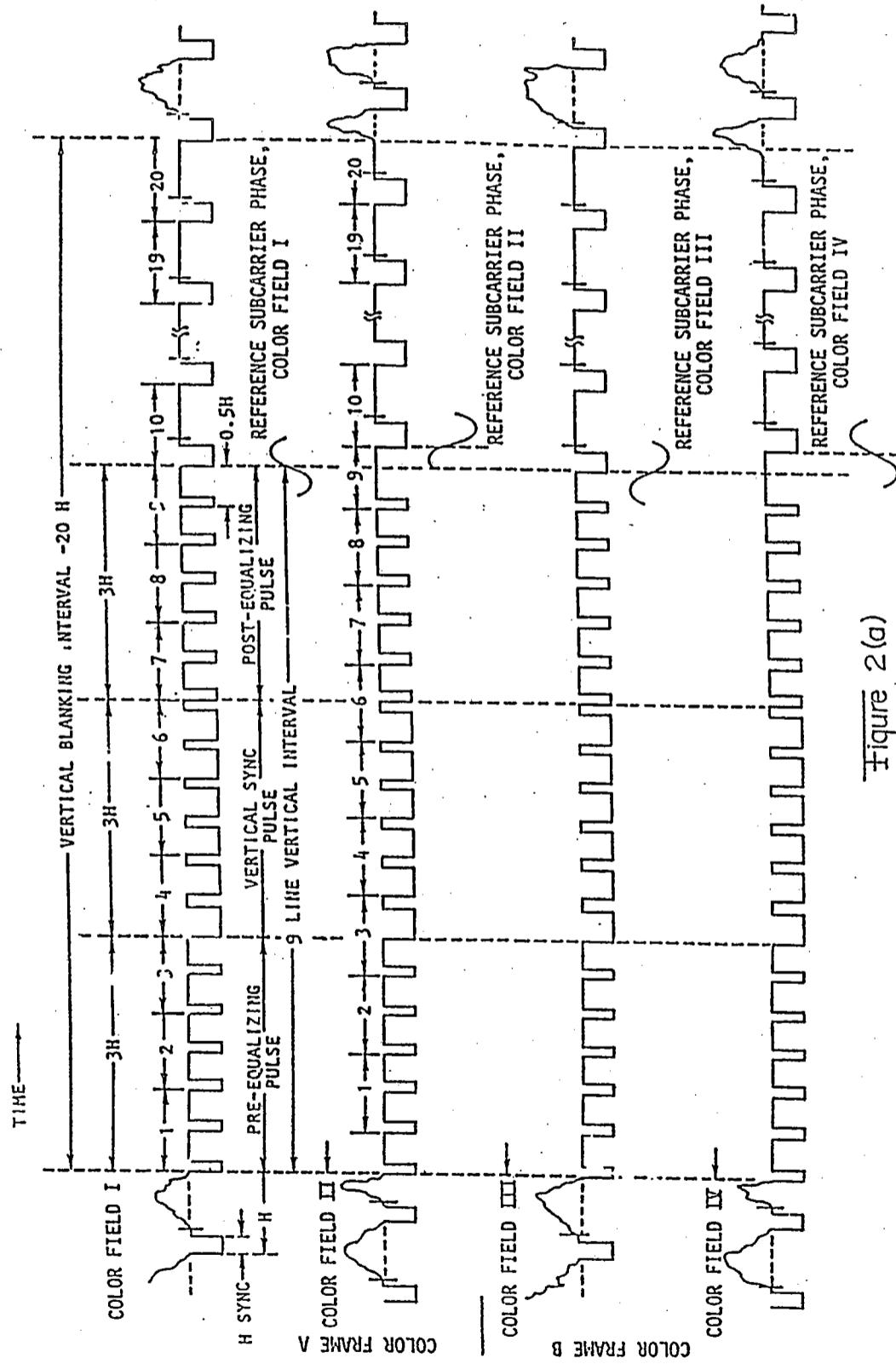


Figure 2(a)

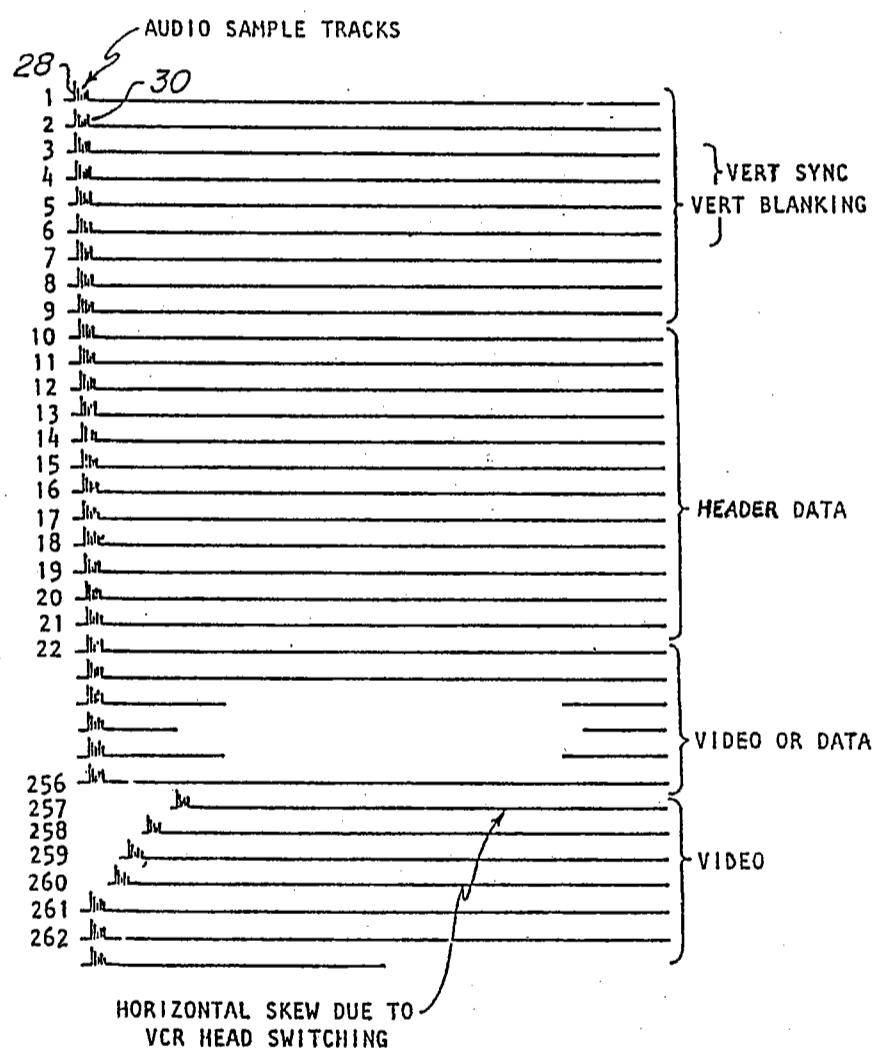


Figure 2(b)

U.S. Patent

Jul. 11, 1989

Sheet 4 of 19

4,847,690

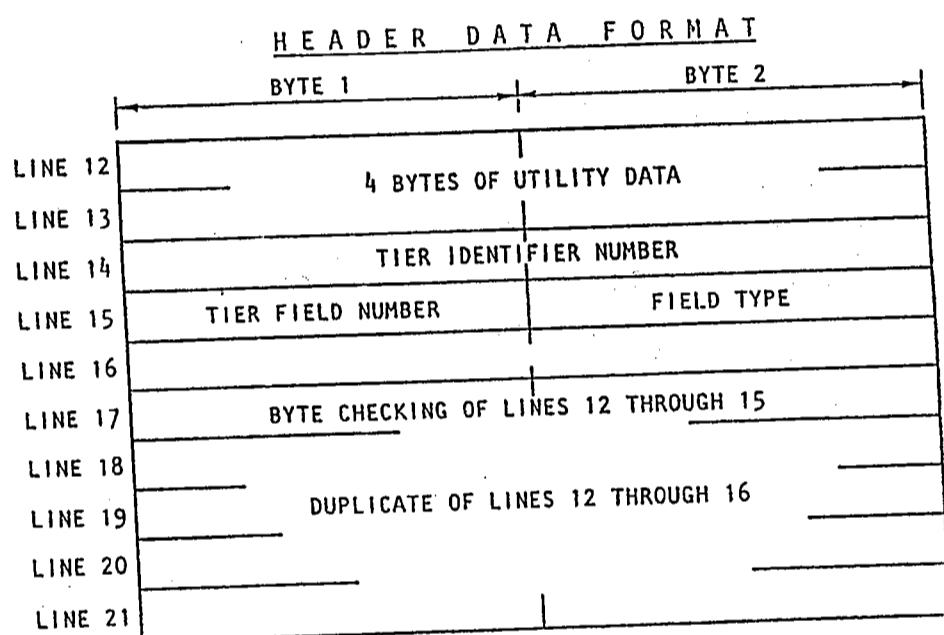


Figure 2(c)

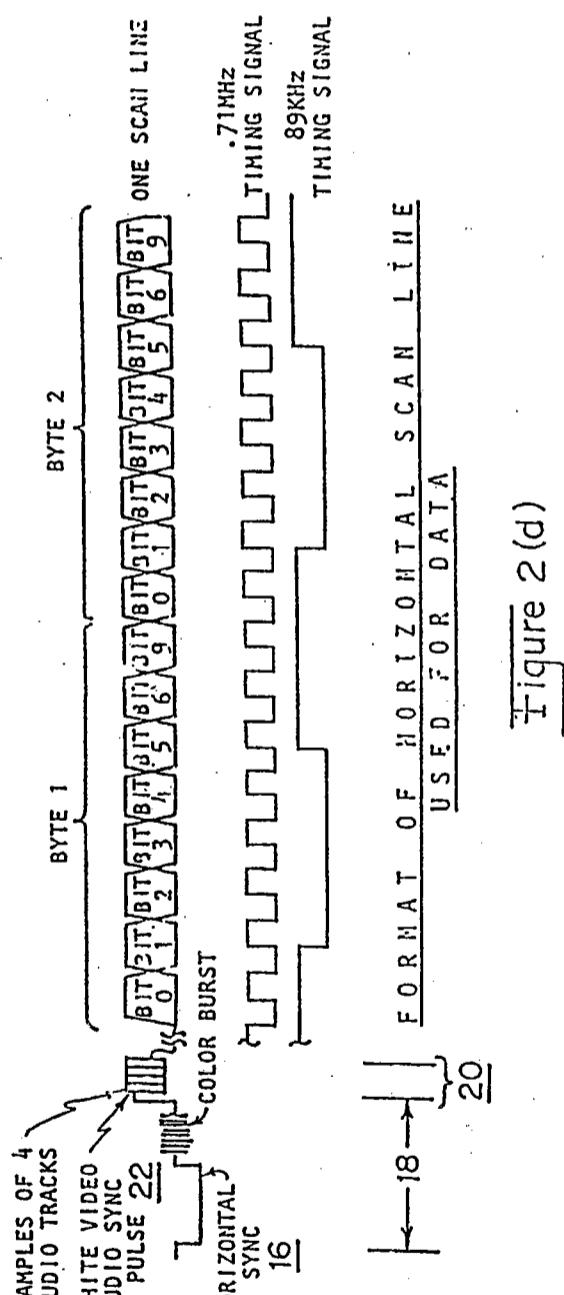


Figure 2 (d)

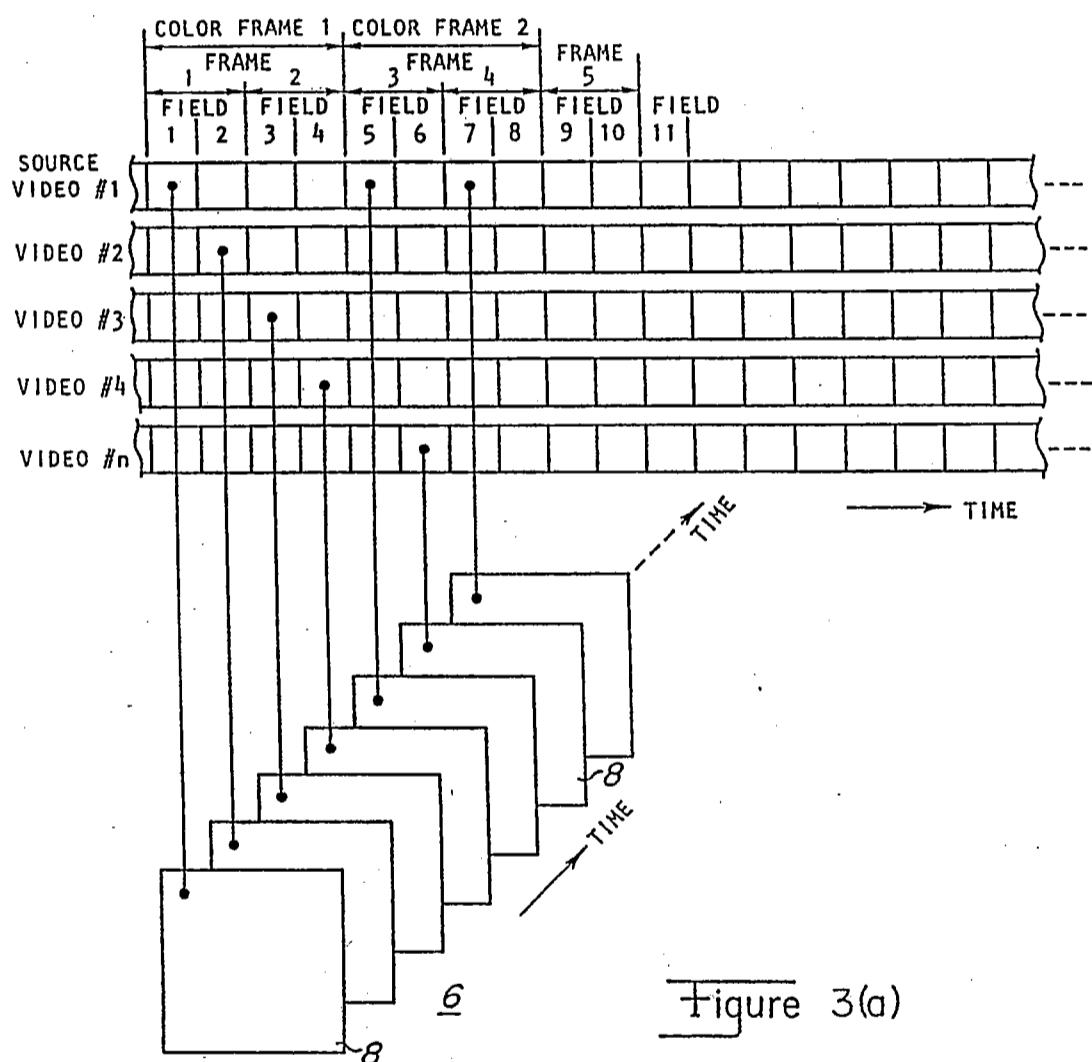


Figure 3(a)

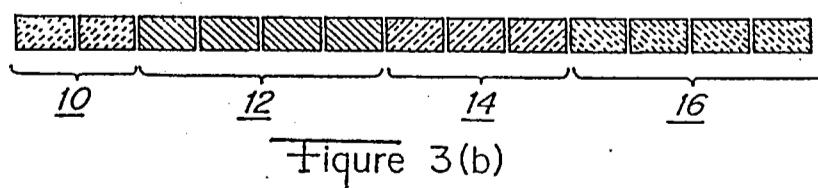


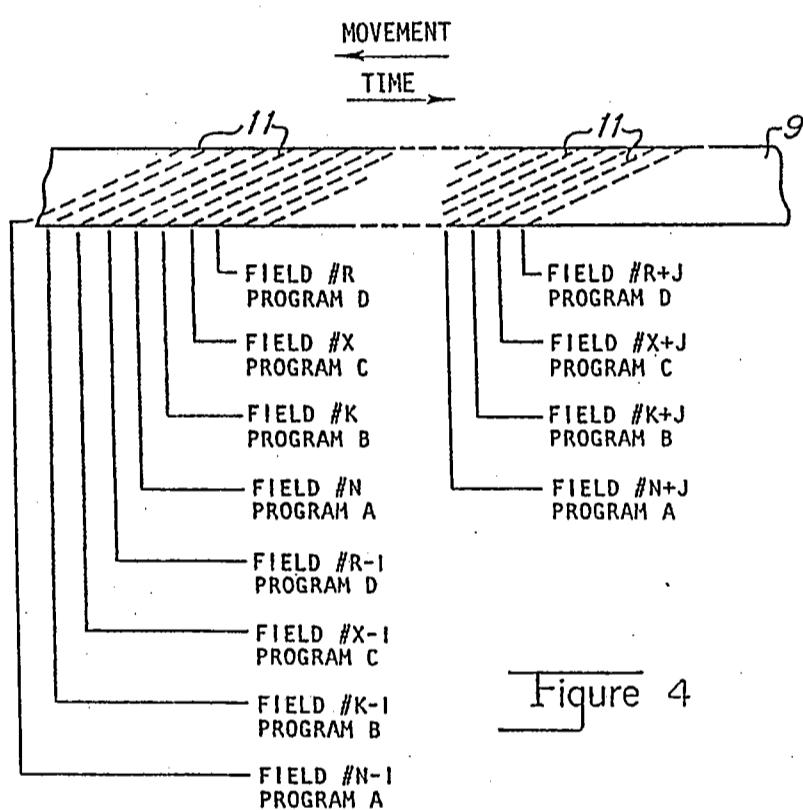
Figure 3(b)

U.S. Patent

Jul. 11, 1989

Sheet 7 of 19

4,847,690



U.S. Patent Jul. 11, 1989

Sheet 8 of 19

4,847,690

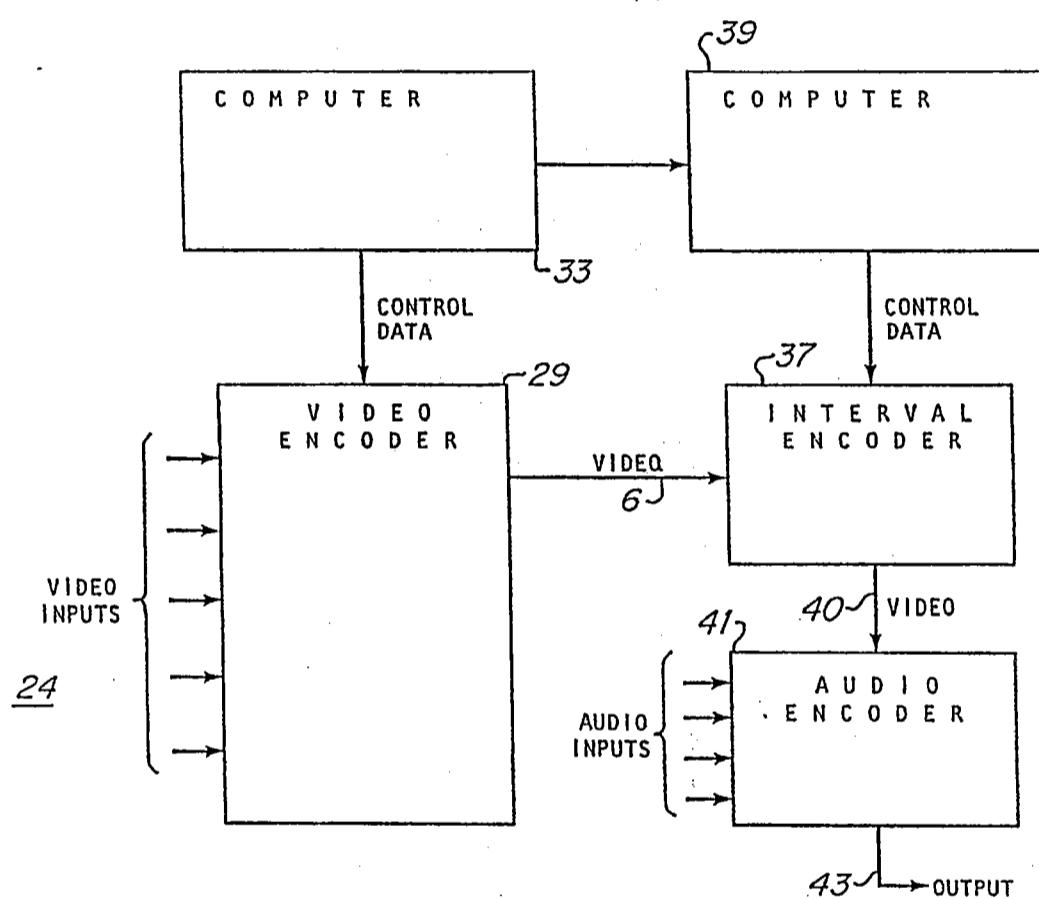


Figure 5

U.S. Patent

Jul. 11, 1989

Sheet 9 of 19

4,847,690

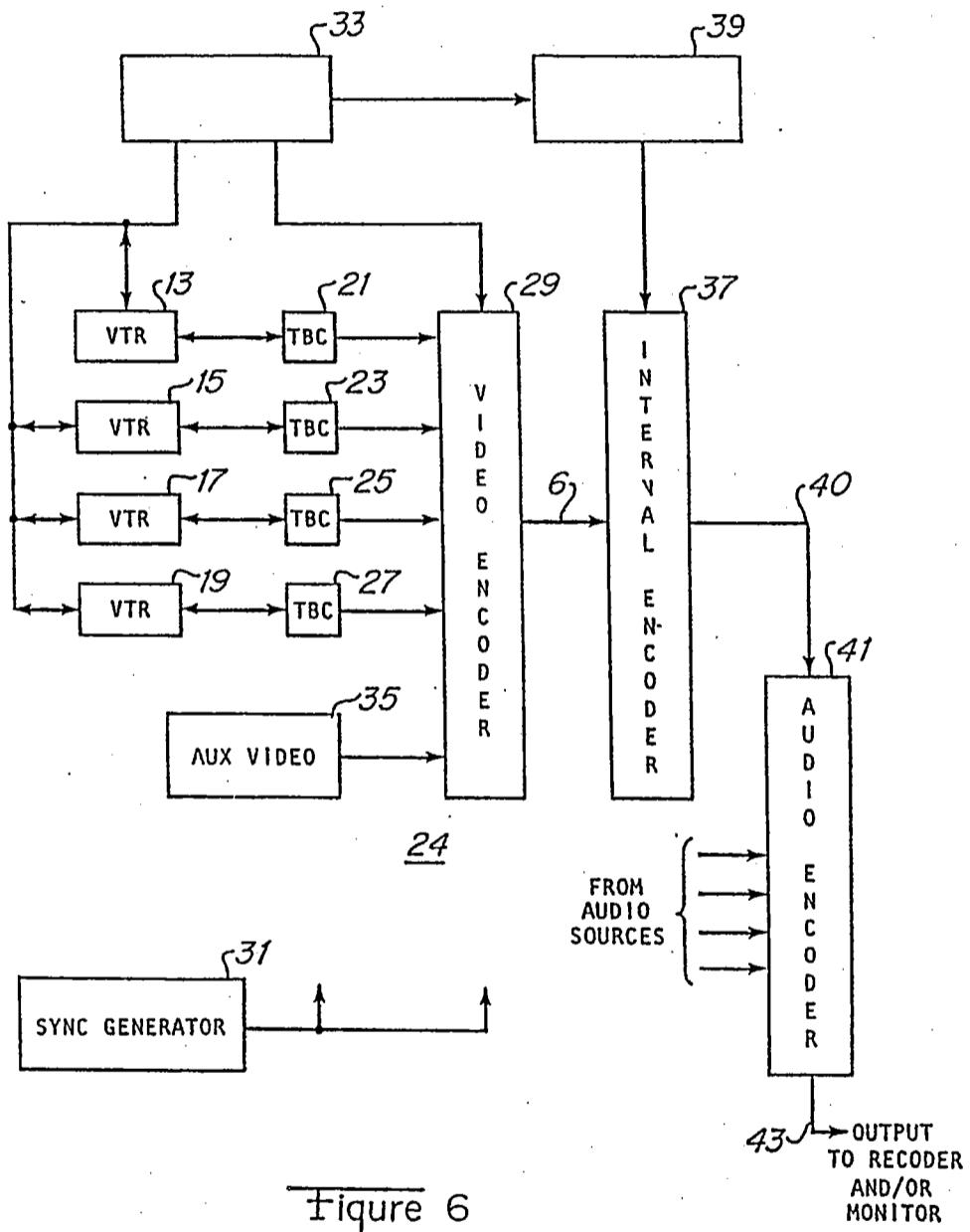


Figure 6

U.S. Patent Jul. 11, 1989

Sheet 10 of 19

4,847,690

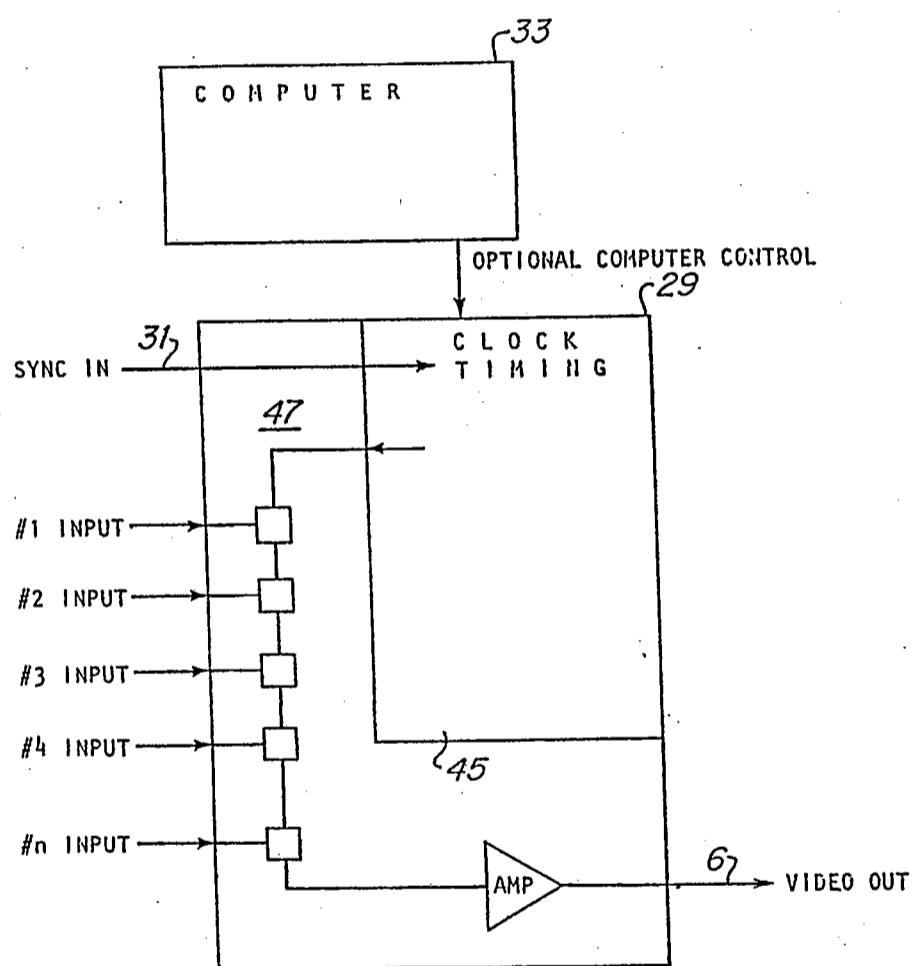


Figure 7

U.S. Patent

Jul. 11, 1989

Sheet 11 of 19

4,847,690

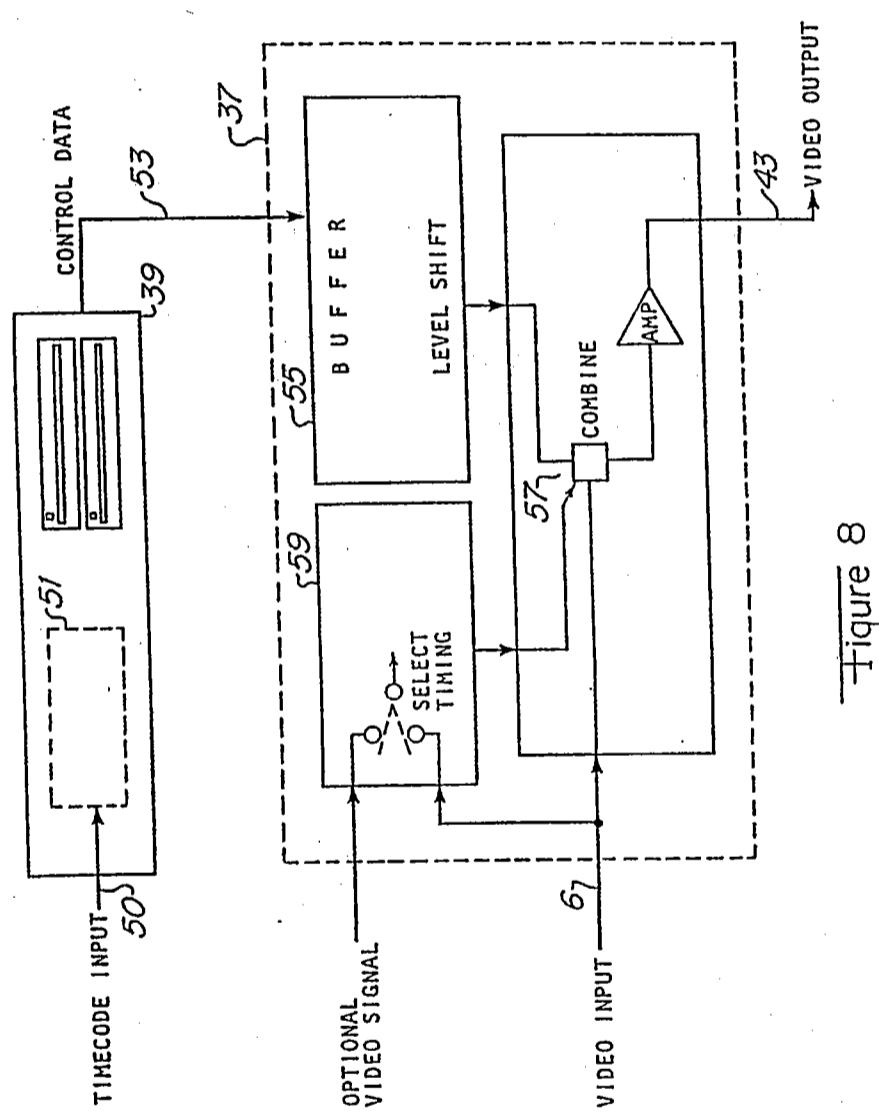


Figure 8

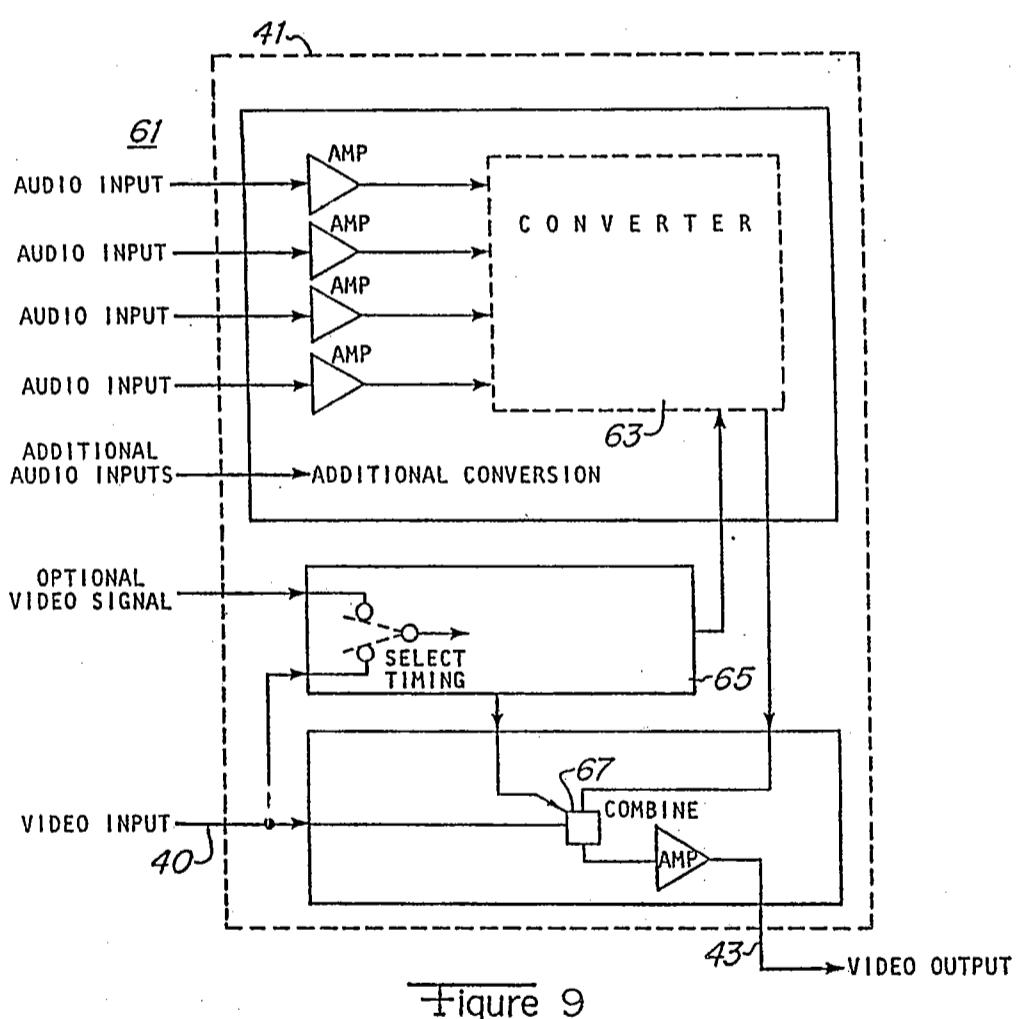


Figure 9

U.S. Patent

Jul. 11, 1989

Sheet 13 of 19

4,847,690

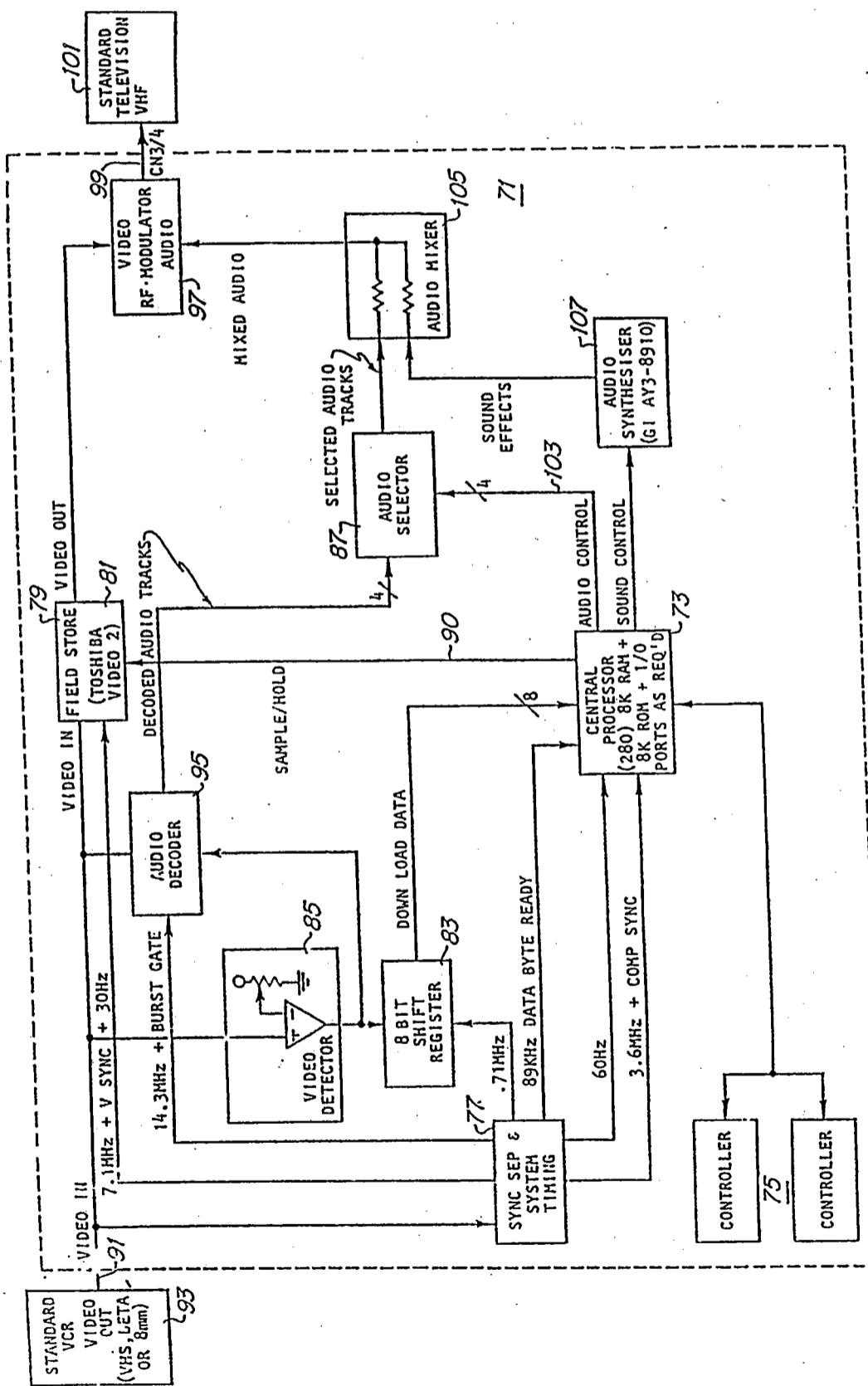


Figure 10(d)

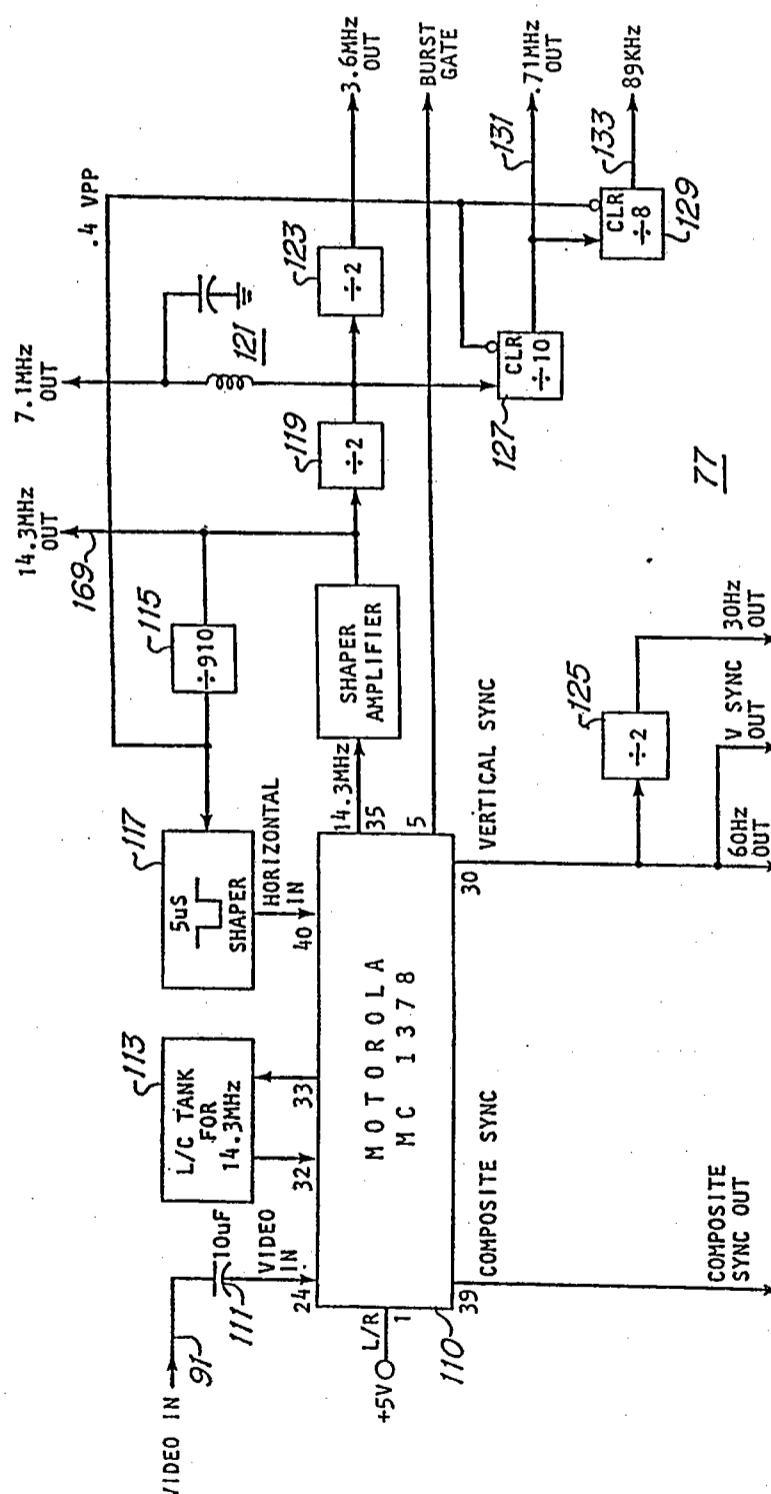


Figure 10 (b)

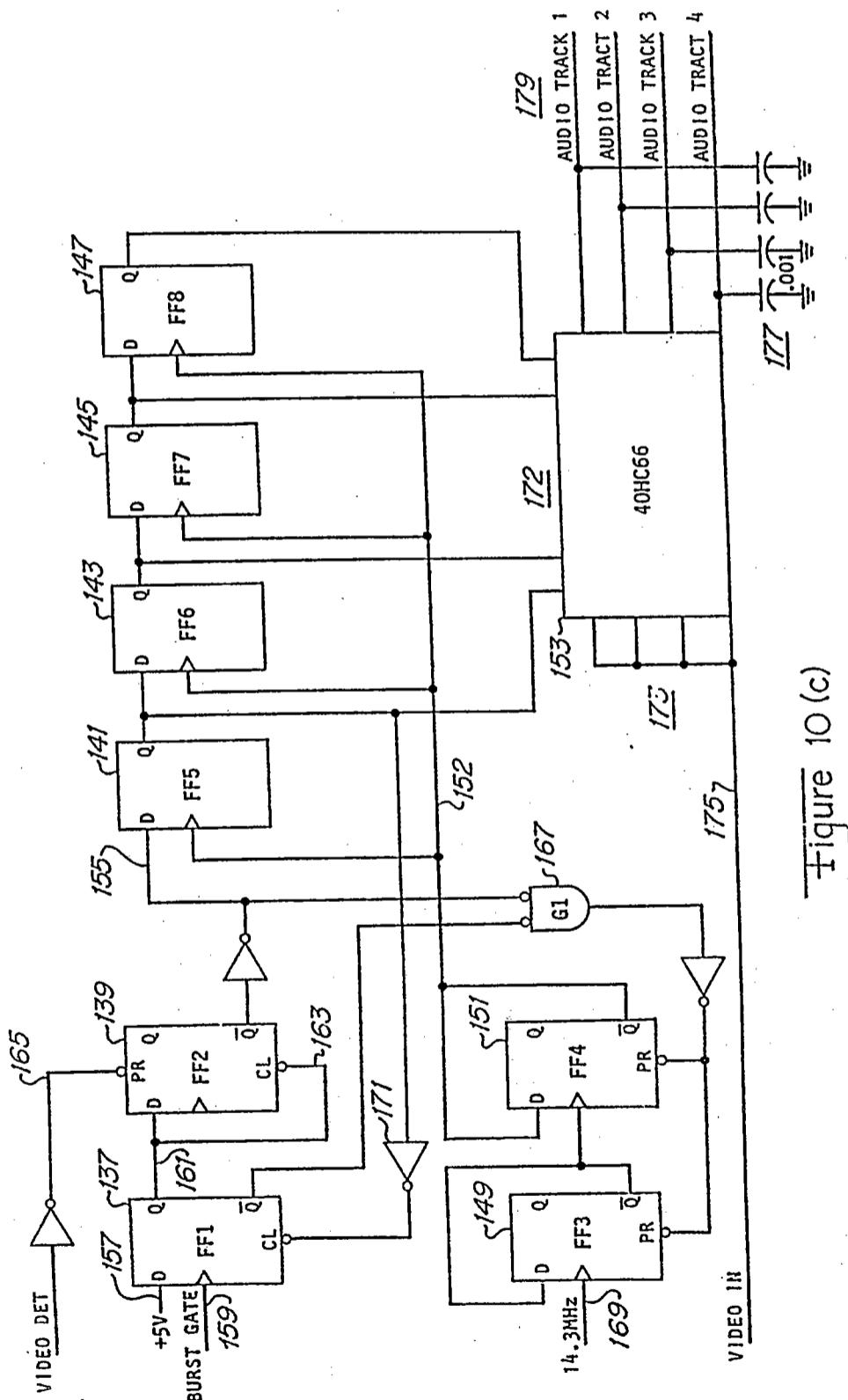


Figure 10 (c)

**U.S. Patent** Jul. 11, 1989

Sheet 16 of 19

**4,847,690**

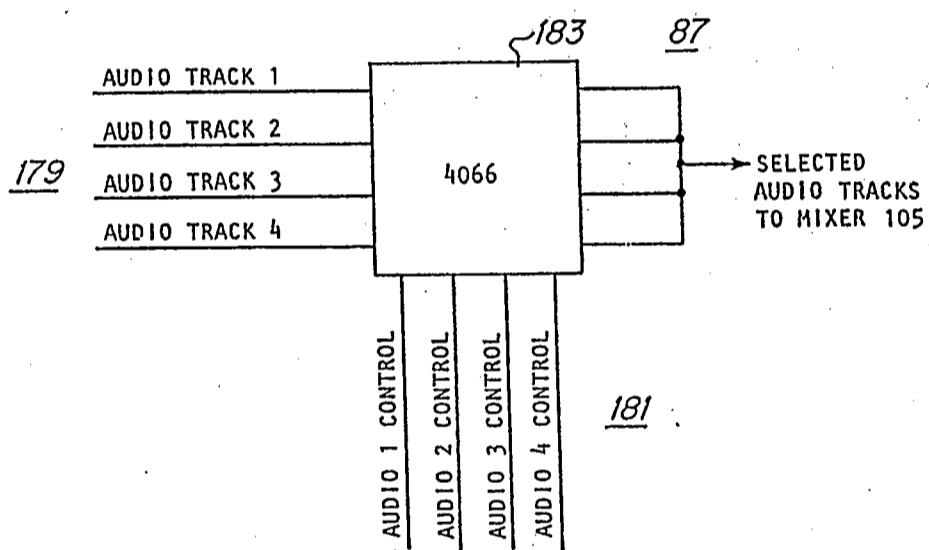


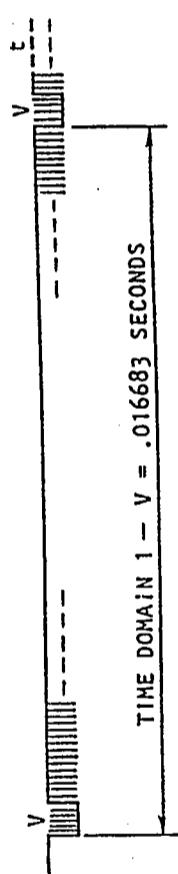
Figure 10 (d)

U.S. Patent

Jul. 11, 1989

Sheet 17 of 19

4,847,690



(AUDIO OF TIME DOMAIN 1 COMPRESSED TO FIT IN TIME DOMAIN 2)

Figure 11 (a)

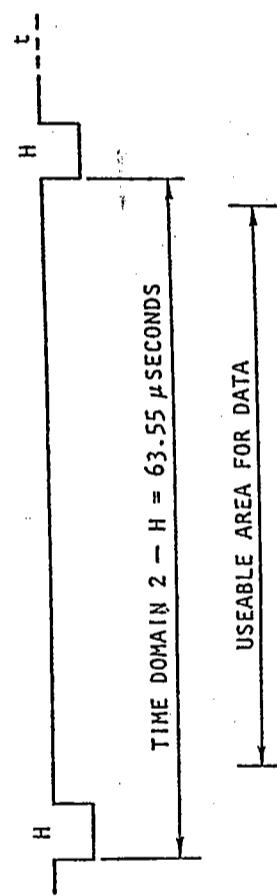


Figure 11 (b)

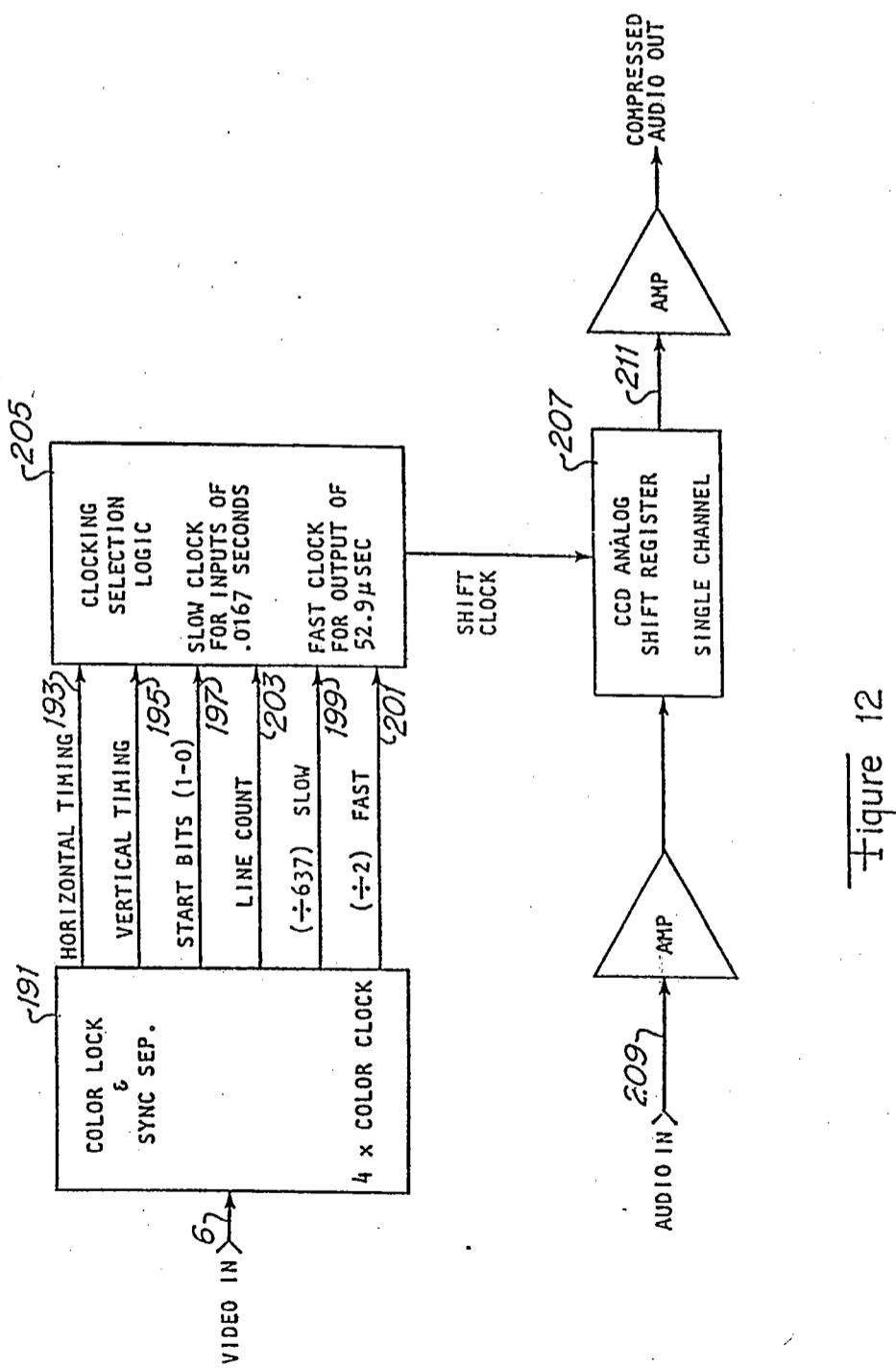


Figure 12

U.S. Patent

Jul. 11, 1989

Sheet 19 of 19

4,847,690

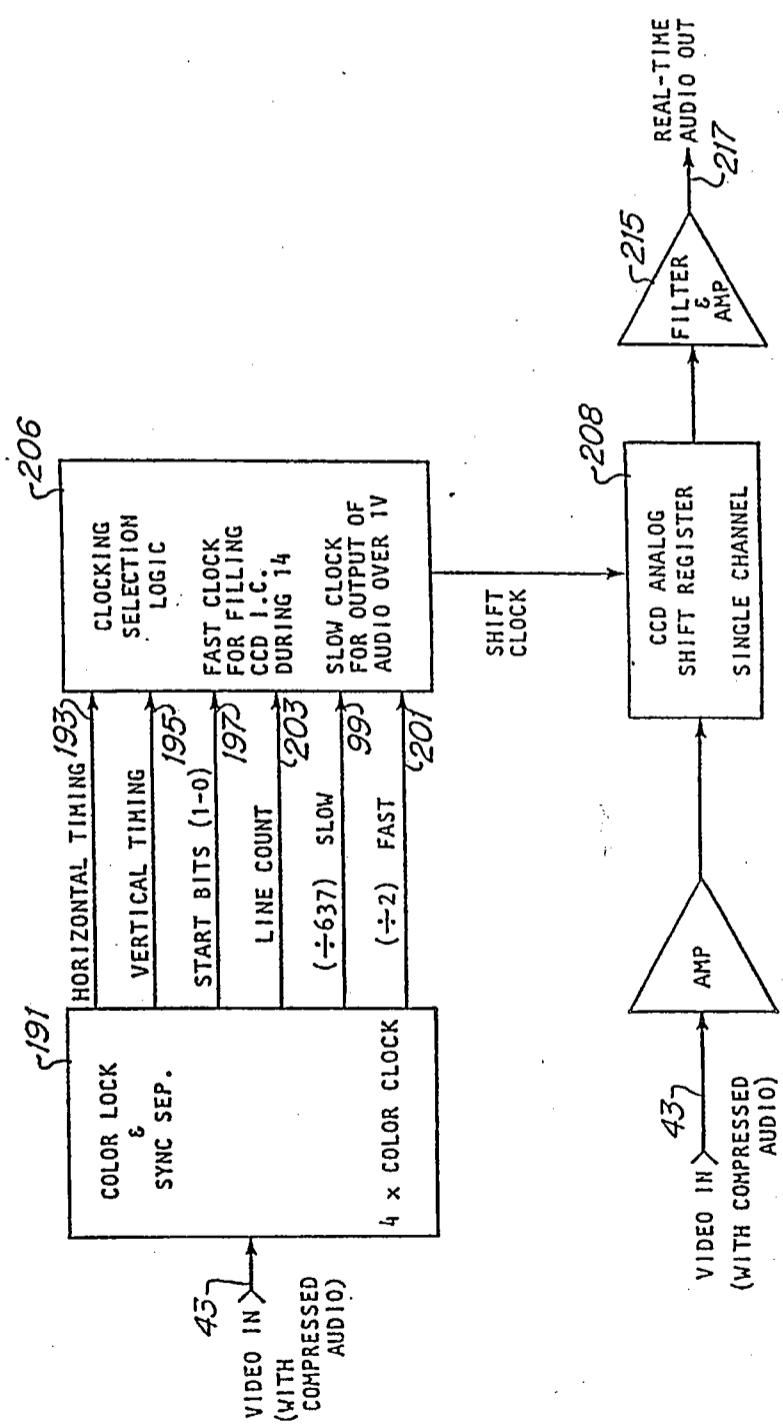


Figure 13

**INTERLEAVED VIDEO SYSTEM, METHOD AND APPARATUS**

**RELATED APPLICATION**

This application discloses subject matter similar to the subject matter disclosed in pending application for U.S. Pat. Ser. No. 016,671, entitled "Video System, Method, and Apparatus" filed Feb. 19, 1987, by Michael Short.

**BACKGROUND OF THE INVENTION**

**1. Field of Invention:**

This invention relates to the system, method and apparatus for displaying selectable ones of multiple television fields that are simultaneously accessible from an interleaved composite video signal, and more particularly to the techniques for interactive display of selectable television program materials that include the images of real people, objects and places or animated images, or combinations thereof.

Certain known forms of processing of television signals permit selective switching between one sequence of images, or 'program', and another sequence of images, or another program, for example, in connection with video games that display a composite real-image background and animated foreground. Selection of one program, or another program for such background displays is conventionally provided by separate tracks of limited length that are pre-recorded, for example, on a video disk. In systems of this type, the background video 'program' may be selectively changed to suit the foreground image (which may be independently generated by computer) by switching between program tracks on the prerecorded video disk.

In other known television schemes, alternate program information may be transmitted on another, non-standard carrier signal, or may be time-shared on a standard carrier signal to constitute split-image display of information schemes of this type are described in the literature (see, for example, U.S. Pat. Nos. 2,681,383; 4,266,240; 4,287,528; 4,393,404; 4,467,356; 4,484,328; 4,510,520; and 4,593,318).

These known schemes are not readily conducive to real-time video displays of continuously selectable program materials of the type, for example, that may be transmitted over cable television networks. Nor may these known schemes be usable for transmitting multiple, real-time programs over a single television channel simultaneously for final selection, editing, and interaction therewith by the viewer.

**SUMMARY OF THE INVENTION**

Accordingly, it is an object of the present invention to provide the system, apparatus and method for interleaving multiple, simultaneous television programs for final selection, editing, and interaction therewith by the viewer. In accordance with the preferred embodiment of the present invention, this and other objectives are accomplished by iteratively interleaving successive fields of different video program materials to form a composite signal for selectable display and reproduction (where a standard video frame comprises two interleaved video fields).

The successive fields of the selected program material are processed to form a video signal that is configured within the NTSC standards for viewing on conventional television receivers. The signal processing

includes a field buffer or storage medium to provide requisite signal information for continuous display of at least one field of the selected program material until updated by a successive field of the selected program material. In this manner, a continuous visual display is formed as a series of rapidly-changing fixed fields of video information. The selection of alternate program material is accomplished by successively updating fixed fields of the alternate program material using the successive fields of the selected alternate program material contained in the composite video signal. Additional information including audio signals, voice, data, and the like, may be included with each successive video field of the multi-program materials separately from the conventional NTSC audio information.

The selection of the program material to be displayed may thus be controlled directly by the viewer for television or telemetry displays, or controlled through an interactive logical control sequence for video games or tutorial programs. The succession of choices of alternate program materials may be stored to recreate the viewer's 'edited' version of the composite video signal. Several selectable programs may be prepared for transmission simultaneously, for example, via cable television for the viewer's selection in real time. In this manner, several sponsors may share the same time slots and pay proportionately less for advertising associated with the multi-programs that reach the viewers more selectively.

**DESCRIPTION OF THE DRAWINGS**

FIG. 1(a) is a pictorial representation of a conventional television signal within the blanking interval;

FIG. 1(b) is a simplified/graphic representation of a composite scan line of a television signal;

FIG. 2(a) is a graph showing several partial television waveforms in accordance with NTSC standards;

FIG. 2(b) is a graph illustrating the horizontal scan lines of a television video field including audio signal information according to one embodiment of the present invention;

FIG. 2(c) is a graphic representation of header and other data included in video fields according to the present invention;

FIG. 2(d) is a graphic representation of a horizontal scan line used for transmitting data and audio within the video field;

FIG. 3(a) is a pictorial, graphic representation of interleaved video fields according to the present invention;

FIG. 3(b) is a graphic representation of video fields re-assembled according to viewer selection from the sequence of interleaved video fields;

FIG. 4 is a pictorial representation of video field scan lines assembled on conventional video recording tape according to the present invention;

FIG. 5 is a general block schematic diagram of circuitry for producing interleaved, multi-program video fields according to the present invention;

FIG. 6 is a block schematic diagram of the encoding system according to the present invention;

FIG. 7 is a block schematic diagram of the video encoder circuitry of the present invention;

FIG. 8 is a block schematic diagram of the interval encoder circuitry of the present invention;

FIG. 9 is a block schematic diagram of the audio encoder circuitry of the present invention;

FIG. 10(a) is a general block schematic diagram of the interleaved video field decoding system of the present invention;

FIG. 10(b) is a block schematic diagram of the sync separation and system timing circuitry of the present invention;

FIG. 10(c) is a block schematic diagram of an audio decoder according to one embodiment of the present invention;

FIG. 10(d) is a block schematic diagram of the audio selector in accordance with the embodiment illustrated in FIG. 10(c);

FIGS. 11(a) and (b) are graphs showing television vertical and horizontal sync intervals, respectively;

FIG. 12 is block schematic diagram of an embodiment of a compressed audio encoder; and

FIG. 13 is a block schematic diagram of an embodiment of compressed audio decoder.

#### DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to FIG. 1(a), there is shown a portion of a conventional television signal within the blanking interval for horizontal retrace of the position of the electron beam in a cathode ray display tube. The basic data cell according to the present invention is the equivalent of one horizontal scan line or trace of an RS-170A composite video signal, as shown in FIG. 1(b). There may be several types of data involved in a data cell in the invention including analog and audio data, several bits of serial digital data, analog video information with color modulation at 3.58 Mhz., and may include a mix of these types of information on horizontal scan lines within vertical blanking and non-vertical blanking portions of a video field.

A logical accumulation of data cells constitutes a video field that contains 262.5 horizontal scan lines or traces of information sufficient for producing a displayable field on the cathode-ray display tube of a television receiver. The first horizontal scan lines or traces 1 through 9 are reserved for vertical synchronization in conventional television signals, and lines 257 through 262 usually suffer degradations from skewed tape recording servos, noise caused by switching of tape-recorder heads, and the like. The remaining intermediate 248 horizontal scan lines that form the central region of a displayable field (and, optionally, all horizontal scan lines) are available to include digital and audio signal information according to this invention in a manner which is compatible with existing NTSC requirements for horizontal line placement within the displayable field.

Referring to FIG. 2(a) there is shown a graph of several conventional television waveforms within the vertical blanking interval. Color fields I and II are associated with color Frame A, and color fields III and IV are associated with color frame B. Of course, such signal waveforms continue through this sequence of 262.5 conventional scan lines or traces, as illustrated in FIG. 2(b).

In addition to the conventional field of horizontal trace information for displaying one video field, the present invention introduces audio information on all horizontal scans or traces of a video field in the one embodiment illustrated in FIG. 2(b) (or as compressed audio information on only a selected few horizontal scans or traces such as lines 10 and 11 in another embodiment).

FIG. 2(b) shows how a standard NTSC video field is modified in accordance with the present invention. In accordance with NTSC standards a video field contains 262½ scan lines. In one embodiment of the present invention, the beginning of each and every scan line contains an audio sync pulse 28 followed by a variable number of audio sample tracks 30. Note that where VCR head switching occurs, which is normally at line 257 in the video field, there is a horizontal skew that occurs. The purpose of the audio sync pulse is to locate the audio samples even during the period when the horizontal lines are skewed. In the video field, lines 1 through 3 and lines 7 through 9 contain equalizing pulses, and the vertical sync pulse occurs during lines 4 through 6. Lines 12 through 21 contain header data, as illustrated in FIG. 2(c), which indicates the position of this field in a logical sequence of fields. The header data is also relevant to the program material, including the spacing of successive fields of the same program material. Additional data may be included in lines 22 through 256 in lieu of normal video.

FIG. 2(c) illustrates the format for header data. Each line contains two 8-bit bytes or one 16-bit word. Lines 12 and 13 contain 4 bytes or two words of utility data which can be used to control program flow, synchronized with the current program material. Line 14 is a tier identifier number that indicates which tier or program material this field is a member of. The first byte of line 15 contains a tier field number. This is a sequential number beginning with 0 for the first field of a new tier or portion of program material. A value of 255 indicates the end of this tier or program material. The second byte of line 15 contains a field type identifier. Field type has 256 possibilities and indicates the format of video versus digital data contained within this field. A type of 0 indicates that lines 22 through 262 contain video information only. Other values indicate the type and location of digital data within the normal video period of lines 22 through 256. For example, a value of 1 may indicate that lines 22 through 256 contain digital data in the form of program header. A value of 2 may indicate that lines 2-256 contain digital data (not for display) which represents program logic. Line 16 contains the byte check sum of lines 12 through 15. Lines 17 through 21 inclusively, may contain a duplicate of lines 12 through 16.

FIG. 2(d) shows the format of a horizontal scan line used for digital data. As with a conventional NTSC signal, a typical horizontal scan line begins with the horizontal sync pulse, a colorburst and then, according to the present invention, an audio sync pulse followed by samples of audio tracks (in this case, four shown). At a point in the video area which is approximately 20 microseconds past the leading edge of the horizontal sync pulse, a stream of 16 databits occurs. The width of each bit is approximately 1.4 microseconds. The value of 0 is indicated by a video level below the 40% mark of black to white video, and the value of 1 is indicated by a level greater than 60% of the range between black and white. The 0.71 megahertz timing signal and the 89 kilohertz timing signal are both referenced to the leading edge of the horizontal sync pulse and are used to show the relative position of the databits with respect to these timing signals.

FIG. 3(a) illustrates the scheme by which video material from several sources are combined and interleaved to form a video signal compatible with the present invention. In the example shown, field 1 of frame 1 of colorframe 1 is presented as the first field. Field 2 is

selected from field 2 of frame 1 of colorframe 1 of video source 2. Continuing with the example shown, field 3 of video source number 3 is selected for the third field. Field 4 of video source number 4 is selected for the fourth field. The fifth field is selected from field 5 of video source number 1. Field 6 is selected from video source N, and field 7 is selected from video source 1. The order in which video sources contribute to the resultant stream of video fields may be fixed or variable according to the present invention. It should be noted, however, that each field in the resultant stream of video fields corresponds to the like numbered field of the tier or program source from which the video field is selected.

Referring now to FIG. 3(a), there is shown a pictorial illustration of video signals #1 through #N of displayable programs assembled into a selected sequence of video fields 6. Each of the video signals #1 through #N may be referred to for convenience as a 'tier' and, as illustrated in FIG. 3(a), the selected video fields of the tiers (whether color, or black and white, or non displayable data) are assembled in a sequence to form the composite multi program (or multi tier) video signal 6, according to the present invention. Specifically, if the sequence of video fields in video signal 1 is associated with scene 1 (e.g. camera 1 providing one camera angle), then that tier 1 is of scene 1. Similarly, tiers 2, 3, n provide scenes 2, 3, n, respectively.

In accordance with the present invention, the video fields of tier 1 are alternated in sequence with the video fields of tier 2 (and tier 3 and tier n) to produce the resulting composite signal 6, as illustrated in FIG. 3(a). The sequence of video fields selected from the tiers #1 through #N may be cyclic or according to an arbitrary sequence. Each video field associated with a tier may correlate with the next video field in the sequence of that tier, so that the related video fields may be re-assembled during play back and display according to the present invention as continuous portions of arbitrary length or duration of the initial tiers, as illustrated pictorially in FIG. 3(b).

In order to assure that video fields associated with a given tier may be properly correlated, each video field 8 includes header or identification information, which as previously described, starts at line 10 and continues to line 21, for example in the following sequence:

lines 10 and 11-	Compressed Audio Cell (in one embodiment);
lines 12 and 13-	Four bytes of utility streamer data;
line 14-	Tier identifier number
line 15-	Tier field number and field type
line 16	Checksum for lines 12-15
lines 17-21	May duplicate lines 12 through 16.

The tier identifier number identifies each of the initial tiers of displayable program material (up to 65,535), and the tier field number designates the field sequence in the designated tier, as illustrated in FIG. 3(a). The video field type identification identifies a normal video field from a digital video field, as previously described, since a digital video field will normally not be displayed. With the header information in place, the transfer rate of streamer data is 240 bytes per second, or about 1900 BAUD. Each video field contains 2 bytes of data that indicate the tier number of which this field is a number to facilitate the skipping from field to successive field in a non-cyclic sequence for re-assembling the correlated,

sequential video fields of a selectable tier, as illustrated in FIG. 3(b), according to the present invention. The interleave factor between video fields identifies the distance (or interleaved number of video fields) between correlated video fields of the same tier and can be arbitrarily changed at any video field according to the needs of the current program material. Of course, if the composite video signal 6 includes successive fields cyclically assembled from the tiers #1 through #N, then simple iterative counting may be used to re-assemble successive fields of a selected tier without the need for header information. Frequent, non-cyclic selections of one tier over another tier, however, may be required to improve the visual display of rapid image movements in such one tier of program material, compared with slower image movements (for example, a scoreboard) in another tier of program material. The typical tier of program material is a displayable video scene of moving objects. In order to maintain reasonably acceptable display of motion, it is believed that compilation of four tiers of program material is a subjective limit, although many more tiers are possible according to the present invention. In addition to typical tiers of displayable program material, the present invention also accommodates data tiers which can transfer digital data (normally not for display but for control or other purposes) at the rate of approximately 28K bytes per second. The composite video signal 6, as illustrated in FIG. 3(a), may therefore be selectively and interactively reassembled into portions 10, 12, 14, 16 of arbitrary duration of the original tiers of successive video fields, as illustrated in FIG. 3(b). Of course, the composite signal illustrated in FIG. 3(a) may occur in real time, for example, for cable transmission or be recorded on, for reproduction from, a recording medium such as conventional video disks or video cassette recording tape.

Referring now to FIG. 4, there is shown a pictorial representation of the composite signal for multi program material compiled on video tape 9. All the image information for a single video field of color or black and white program material, including header and audio samples, as previously described, is contained in the trace 11 of one head scan that is oriented in skew relationship to the longitudinal dimension of the tape 9 for operation on conventional helical-scan video tape recorder equipment. The traces 11 include scan synchronizing information and field header information, as previously described, for proper operation of decoding equipment, later described herein. The video field traces are interleaved along the length of the tape (i.e. in 'time' in cyclic manner, as illustrated) such that the field (n) for program material A is followed by field (k) for program material B, followed by field (x) for program material C, followed by field (R) for program material D, followed by field (n+1) for program material A, followed by field (k+1) for program material B, and so on for the duration of the composite signal in time (or length of tape 9). Thus, one field of a selected program is followed by a successive field of information for the same program only in the fourth successive trace with fields of independent information for three other programs presented in the three intervening traces. The correlated video fields that comprise one complete program may therefore be reconstructed in accordance with one embodiment of the invention by using every fourth trace in the sequence, while each of the other complete programs may be reconstructed using every

fourth trace displaced by one or two or three traces, respectively. Of course, it should be understood that there may be two or many dozens of individual programs assembled with successive traces of video field information for each program cyclically (or non-cyclically) interleaved to form a composite multi-program or multi-tier video signal in this manner.

As later described herein, the display image associated with each video field of information of a given program remains on display until updated by the next video field of information of the same program. Therefore, increasing the number of interleaved, individual programs increases the delays between updates, with concomitant appearance of jerky or stroboscopic movements of displayed video images. In accordance with one embodiment of the present invention, program materials including rapidly moving images may include higher density of successive fields of information (i.e. greater number of fields per unit time) in non-cyclical distribution in the composite signal to reduce the delay time between updates of fixed-field displays to thereby reduce the jerky or stroboscopic appearance of moving images being displayed. Of course, it should also be understood that such composite signal need not be assembled on a video tape, as illustrated in FIG. 4, but may be assembled and transmitted in real time, for example, via cable television network, with successive fields of multiple programs interleaved in cyclic or non-cyclic succession, as previously described.

In accordance with one embodiment of the present invention, each video field of information includes the header data or coded signals previously described to designate the correlated or associated program material and the field number in the succession of video fields for that program. In addition, each video field includes audio signals that are introduced into the horizontal scans or traces of each video field in various ways according to alternative embodiments of the present invention. In one embodiment, as illustrated in FIG. 2(d), the initial portion of the video waveform that follows the horizontal synchronizing pulse 16 by a selected time interval 18 includes four distinct timed intervals 20 (for four tracks of audio programs) preceded by an audio synchronizing pulse 22. In each of the four timed intervals occurring in the interval 20 that follows the audio sync pulse 22, individual samples of analog audio signals for each audio signal channel are imposed on the video intensity level at the start of the horizontal scan or trace. Since this initial sector of the horizontal trace is normally not displayed, the audio signals imposed on the video intensity levels according to this embodiment of the present invention does not significantly alter compatibility with standard NTSC television waveforms. Such individual channels of audio signal samples occurring on each of 262 horizontal traces or scan lines per 1/60 of a second per video field, as illustrated in FIG. 2(b), yields approximately 7.8 KHZ of audio signal bandwidth in each audio signal channel. A greater number than four independent audio channels (for a greater number than four tracks of audio programs) tends to extend the audio-samples interval 20 into visible, displayable portions of horizontal traces 22 through 256, and may therefore require masking to prevent their appearance on the television display screen.

In accordance with another embodiment of the present invention, audio signal information may be selectively introduced into the horizontal traces of a video field by allocating, say, horizontal scan or trace lines 10

and 11 for compressed audio signalling. In this embodiment, the individual channels of audio information are compressed into the video interval of horizontal trace line 10, or line 11, or both, as illustrated in FIG. 1(b), for high-speed transfer into a buffer register. Then, the buffer register can be clocked at low speed to yield the slow-speed stream of audio signal samples over the time interval until horizontal scan line 10 next appears in the next video field. Audio signals introduced into video fields of displayable programs remain compatible with standard NTSC television waveforms, and additional audio channels can be readily introduced with higher speed transfers of audio samples into buffer registers, or with additional horizontal scan lines allocated for compressed audio signalling.

In each of the embodiments described above for introducing audio signal information into the video field information, it should be noted that audio information correlated with all tiers of displayable programs is included in each video field of each tier. In this manner, the on-going audio signal information correlated with any one tier is always available from the video fields of such one or other tiers, even though the video fields for such one tier may be infrequently sampled (for example, to display a scoreboard with continuing background music).

Referring now to FIGS. 5 and 6 there are shown simplified block schematic diagrams of apparatus according to one embodiment for producing the composite video field signal 6 for multi-tier or multi-program interactive operation on multiple video inputs 24.

In FIG. 6, the video inputs 24 are illustrated to be derived from multiple sources 13, 15, 17, 19 of program material (such as video tape recorders or video cameras). Each source is connected via conventional time-base correctors 21, 23, 25, and 27 to a video encoder 29 which produces the composite video signal 6 by interleaving the selected video fields from the multiple program sources. The interval encoder 37 generally introduces header data on the composite video signal under control of a computer or processor 39, and the audio encoder 41 that follows generally introduces audio information correlated with the multiple programs into the video field output information 43.

Specifically, with reference to FIG. 6, the multiple program sources 13-19 (e.g. video tape recorders or video cameras) in association with the synchronization generator 31 assure that the program sources 13-19 operate in synchronism to provide individual video field inputs 24 from the multiple program sources 13-19 that are synchronized in time, as illustrated in FIG. 3(a). Auxiliary video information 35, for example, background scenery that is normally intended to be updated only over long intervals, or filler scenes for display during transitions between program sources, may also be supplied to the video encoder 29.

A minicomputer or microprocessor 33 is connected to control the signal sources with respect to time codes (such as status and position information that are associated with the fields of taped program materials). The computer 33 thus controls the encoder 29 which may be a conventional video switch to select particular, successive fields of selected program sources in accordance with an edit decision list, later described herein, to produce a composite video signal 6, as previously described in connection with FIG. 3(a). The selection of particular video fields at the inputs 24 from the program sources 13-19 can be made at every video field by the

A COVERBINDER COVER 1-800-366-5060  
1 1/4" Classic PTO 1683-9 for 2#1 to 300 sheets