

METHOD AND APPARATUS FOR LOGICALLY REJECTING PREVIOUSLY
RECORDED TRACK RESIDUE FROM MAGNETIC MEDIA

Field of the Invention

The present invention pertains generally to data recording and
5 recovery methods, and more particularly to a method for logically rejecting
previously recorded track residue from magnetic media.

Related Patents

The present invention is related to co-pending U.S. patent application
entitled "Method And System For Monitoring And Adjusting Tape Position
10 Using Control Data Packets", invented by McAuliffe et al., and having an
internal docket number of 9086/107 and a serial number of 09/193,030, filed
concurrently herewith on November 16, 1998, and co-pending U.S. patent
application entitled "A Method Of Reacquiring Clock Synchronization On A
Non-Tracking Helical Scan Tape Device", invented by Blatchley et al., and
15 having an internal docket number of 9086/108 and a serial number of
09/192,808 filed concurrently herewith on November 16, 1998, and co-
pending U.S. patent application entitled "Rogue Packet Detection And
Correction Method For Data Storage Device", invented by McAuliffe et al.,
and having an internal docket number of 9086/105 and a serial number of
20 09/192,809, filed concurrently herewith on November 16, 1998, and co-
pending U.S. patent application entitled "Variable Speed Recording Method
and Apparatus for a Magnetic Tape Drive", invented by Beavers et al., and
having an internal docket number of 9086/101 and a serial number of
09/176,079, filed on October 20, 1998, and co-pending U.S. patent
25 application entitled "Overscan Helical Scan Head for Non-Tracking Tap

Subsystems Reading at up to 1X Speed and Method for Simulation of Same", invented by Blatchley et al., and having an internal docket number of 9086/104 and a serial number of 09/176,013, filed on October 20, 1998, and co-pending U.S. patent application entitled "Fine Granularity Rewrite Method and Apparatus for Data Storage Device", invented by Zaczek, and having an internal docket number of 9086/106 and a serial number of 09/176,015, filed on October 20, 1998, and co-pending U.S. patent application entitled "Multi-level Error Detection and Correction Technique for Data Storage Recording Device", invented by McAuliffe et al., and having an internal docket number of 9086/102 and a serial number of 09/176,014, filed on October 20, 1998, all of which are commonly owned and all of which are hereby incorporated by reference.

Background of the Invention

Tape storage technology is routinely used for routine system back up and long-term data archiving. In the ongoing goal to minimize the cost-to-capacity ratio while still maintaining data integrity, ways are always looked at to reduce drive costs. One way to reduce drive costs is to reduce the costs of the components of the recording device. For example, in the tape drive industry, many tape drive architectures employ an erase head for erasing a magnetic tape before recording data to it. In prior art architectures, this is necessary because portions of tracks written onto the tape that are not precisely overwritten may be unintentionally detected by the read head during a later read session. These previously written portions of tracks,

herein termed "residue", are problematic because although they are not intended to be read, they are still detected by the read logic.

Accordingly, a need exists for a method for logically rejecting previously recorded track residue during a read session to prevent data that is not intended to be read from being included in the recovered data sent to the host system.

U.S. Patent No. 4,910,736 to Tonaka et al. describes the addition of an identification (ID) code and an error check code to each block of the data that makes up a track. The ID code is used to indicate that the block of data belongs to a particular recording session. The error check code is used for error checking of the block of data. In the absence of an error, the original ID code is reconstructed and compared to a reference ID code obtained from a reference ID code setting circuit. If the reconstructed ID code does not match the reference ID code, the block of data is logically rejected as not intended to be read. This approach for logically rejecting residue data reconstructs the original ID code for each block of data recovered from the medium. In addition, a separate reference ID code setting circuit is employed to obtain the reference ID code.

In the interest of reducing the amount of time it takes to determine whether a given block of data belongs to the same recording session as identified by the reference ID code, it would be desirable to have a method for logically rejecting recovered data without having to reconstruct the original ID code for every block of data. In addition, it would be desirable to reduce the amount of hardware, and thus the amount of valuable chip space required to implement the method.

Summary of the Invention

The present invention is a novel method for logically rejecting previously recorded track residue from magnetic media by encoding an identical recording session ID code into the track packet error correction code of every track packet recorded during a single write session. Error correction is performed on the first few track packets to acquire a reference session ID. The reference session ID is used to preload the ECC syndrome and CRC generators prior to processing each recovered packet to determine whether the recovered packet is valid for the current session. Those packets that are recovered but which have a session ID that does not match the acquired session ID are inherently rejected as unintended to be read because the ECC syndromes and CRC generators will indicate an error. As a result, there is no need to actually reconstruct the packet's original session ID. Accordingly, the invention provides an efficient way of determining whether a recovered track packet was recorded during the recording session identified by the acquired session ID.

In accordance with the invention, a session ID unique to a given recording session is generated at the beginning of the recording session and encoded into an error correction code redundancy field of a track packet. When the track packet is written to tape, the session ID itself is not actually sent to the tape, thereby reducing the amount of overhead data required for each track and increasing the actual data-capacity-to-overhead ratio of the media. During a data recovery session, the session ID encoded in the ECC redundancy field of the track packet is acquired by assuming that the entire contents of the first few track packets are good and performing packet error

correction on a number of packets to reconstruct the current session ID. If several consecutively reconstructed session IDs match, the session ID is considered acquired. Once the session ID is acquired, it is preloaded into the CRC/ECC error detection and correction hardware prior to processing
5 each subsequent recovered track packet. Only recovered track packets that are encoded with the acquired session ID are considered good and therefore allowed to be sent to the data buffer; packets that are encoded with a different session ID other than the acquired session ID are rejected by the CRC/ECC error detection and correction hardware.

10 Accordingly, the technique of the invention eliminates the need for a costly erase head yet requires no additional track packet overhead to be recorded onto the magnetic media. In addition, track packet rejection is more efficient since once the reference session ID is acquired, the original session ID need not be reconstructed for every recovered track packet.
15 Finally, the same error correction hardware used to correct packet errors is used to acquire the session ID, thereby eliminating the requirement for separate hardware to do each function.

Brief Description of the Drawing

20 The invention will be better understood from a reading of the following detailed description taken in conjunction with the drawing in which like reference designators are used to designate like elements, and in which:

FIG. 1(a) is a side view of a portion of a magnetic tape illustrating a first track pattern that results in a previously recorded track residue zone;

FIG. 1(b) is a side view of a portion of a magnetic tape illustrating a second track pattern that results in a previously recorded track residue zone;

FIG. 2(a) is a flowchart illustrating a method in accordance with the invention for generating and encoding a session ID in a track packet;

5 FIG. 2(b) is a diagram illustrating the construction of a track packet in accordance with the method of FIG. 2(a);

FIG. 3(a) is a flowchart illustrating one embodiment of a method in accordance with the invention for acquiring the current session ID at the start of a data recovery session when the tape drive is in a normal read mode;

FIG. 3(b) is a diagram illustrating the construction of a track packet in accordance with the method of FIG. 3(a);

FIG. 4(a) is a flowchart of one embodiment of a high-speed search session ID acquisition method;

FIG. 4(b) is a diagram illustrating the high-speed search session ID acquisition process;

FIG. 5(a) is a flowchart of one embodiment of a method in accordance with the invention for logically rejecting residue data during a data recovery session; and

FIG. 5(b) is a diagram illustrating the process of FIG. 5(a).

20 Detailed Description

A method and apparatus for logically rejecting previously recorded track residue during a read session of data recorded over the previously recorded tracks is described in detail hereinafter.

In the recording of data onto a magnetic medium that contains
25 previously recorded tracks, previously recorded track residue can result

unless the portion of the media on which the new data is recorded is erased. Previously recorded track residue can result for a number of reasons, including recording/recovery device or media interchange issues, track pitch discontinuity, variable speed recording capability, track curvature, or head clogs.

FIG. 1(a) is a side view of a portion of a magnetic tape 2 illustrating previously recorded track residue zones 6 resulting from interchange between recording/recovery devices or media. More recently recorded tracks 4 overwrite most but not all of any previously recorded tracks.

However, when magnetic media is recorded on one device and then re-recorded to on a different device, a slight difference in recording head alignment prevents the previously recorded tracks from being precisely overwritten and thus results in residue zones, for example those shown at 6 on magnetic tape 2.

FIG. 1(b) illustrates a second track pattern that results in a previously recorded track residue zone. In this embodiment, residue zone 6 results from variable recording speeds. In drives that support variable speed recording, a speed change results in an angular gap between tracks 4a recorded at one speed and tracks 4b recorded at another speed. When a recording speed change occurs over a portion of previously recorded media, previously recorded track residue zones 6 result in the angular gap between tracks 4a and 4b.

Referring back to FIG. 1(a), data is formatted into track packets 8 before being recorded to the magnetic media 2. Multiple track packets 8 exist on each track 4. Each track packet 8 includes framing information 10

used by the read logic of the recording/recovery device for detecting the leading edge of a packet 8, a local packet address field 12 used by the read logic in constructing the global address of the data, a data field 14 for storing user data if the track packet is a data field, a CRC field 16 used by the read logic to detect an error in data field 14, and an error correction code ECC field 18 used in conjunction with the CRC field 16 to detect and correct errors across packet 8. As described hereinafter, the session ID is encoded in the CRC field 16 and ECC field 18 and may be recovered via error correction techniques. However, the session ID itself is not separately stored in the packet on the media.

FIG. 2(a) is a flowchart illustrating a method in accordance with the invention for generating and encoding a session ID in a track packet 8. FIG. 2(b) is a diagram illustrating the construction of a track packet 8 in accordance with the method of FIG. 2(a). At the beginning of a recording session, a session ID 52 is generated by a session ID generator 50 and assigned to the current recording session in step 202. The tape drive firmware keeps track of the last session ID used as part of its logical formatting on tape and always assigns the next one in sequence when a new write session begins. This is accomplished by storing system data containing the last used session ID in the end-of-data (EOD) area and/or the system partition area on the tape that contains system data used by the drive. Thus, when data is appended to previously recorded data during a new recording session, the tape drive firmware extracts the last used session ID from the system data contained in the EOD area, and increments

it by one to generate the current session ID. The session ID remains constant during the entirety of the recording session.

During a recording session, user data is partitioned into data packets 15, which include data field 14 and local packet address field 12. Data packets are formatted into track packets 8 before being recorded to a magnetic media. In step 204, a CRC generator 54 generates a CRC 56 across the assigned session ID 52 and the data packet 15. The CRC 56 is appended as CRC field 16 to the data packet 15. In step 206, an ECC generator 58 generates an error correction code (ECC) 60 over the assigned session ID 52, local packet address field 12, data field 14, and CRC field. The ECC 58 is appended as ECC field 18 to track packet 8. Framing information 10 is added ahead of track packet 8, which is then modulated and sent to the write channel of the recording device to be recorded onto the magnetic media in step 208.

FIG. 3(a) is a flowchart illustrating one embodiment of a method in accordance with the invention for acquiring the current session ID at the start of a data recovery session when the tape drive is in a normal read mode (i.e., when the tape is moving at normal read speed). FIG. 3(b) is a diagram illustrating the construction of a track packet in accordance with the method of FIG. 3(a).

During the acquisition of a session ID, all data packets recovered are assumed to contain no errors. The method 300 begins with a step 302 in which a possible session ID 76, a total packet count 80, and a good packet count 88 are each cleared. In a step 304, a track packet 8 is recovered from tape and copied (minus the framing information 10) into a temporary buffer

location 72, and the total packet count is incremented. In a step 306, a packet session ID 20 associated with the track packet 8 is cleared (i.e., reset to all zeros) in preparation for error correction by packet error correction logic 74. In step 308, error correction is performed on track packet 8 by error

5 correction logic 74. This step corrects the erased packet session ID 20 using the contents of the track packet 8 located in temporary buffer location 72, which are assumed to be correct, to reconstruct the original packet session ID of recovered track packet 8. The reconstructed packet session ID is captured as a possible session ID in step 310. CRC and ECC

10 generators 82 and 84 perform error detection on the corrected track packet 8 in step 312. If an error is detected, as determined in step 314, the possible session ID and track packet are rejected, and the process repeats beginning at step 302. If no errors are detected in the track packet 8, the good packet count is incremented in step 316. The reconstructed packet session ID is

15 then compared against the possible session ID 76 in step 318. After the first pass, a mismatch will occur since the possible session ID 76 is cleared at the beginning of the acquisition process. Accordingly, the reconstructed packet session ID is captured as the possible session ID 76 in step 320. At the point in the method reach in step 320, a single good possible session ID

20 has been detected. In accordance with the illustrative embodiment, the method is always looking for two identical session IDs in a row as a criteria for considering a reconstructed session ID as a real possible session ID. Accordingly, in step 320 the good packet count and total packet count are set to a count value of one if they are not already set to one. Steps 304

through 318 are repeated to attempt to capture two identical reconstructed packet session IDs for two contiguous track packets 8.

Once two contiguous track packets 8 are detected to have been recorded during the same recording session (by having identical packet session ID's), subsequent recovered track packets 8 are monitored for a predetermined number of total packets to determine the ratio of good packets to rejected packets. This stage of the acquisition process begins in step 318 if the reconstructed session ID matches the possible session ID for a second or greater consecutive good track packet with an identical reconstructed session ID. For every subsequent error-free track packet with an identical reconstructed packet ID up to a predetermined number of total subsequent tracks, steps 304 through 316 are repeated. Then in step 322, a determination is made as to whether the predetermined total number of consecutive packets have been processed. If not, steps 304 through 322 are repeated. Once the predetermined total number of consecutive packets have been processed, the ratio of good packets to total processed packets, based on the value of the good packet count 88, is determined in step 324. If the value of good packet count 88 is not acceptable, for example by comparing it to a minimum threshold level, the possible session ID is discarded and the entire acquisition process beginning with step 302 is repeated. If, however, the value of good packet count 88 is acceptable, the session ID is considered acquired, and the possible session ID 76 is adopted as the acquired session ID 78 in step 326.

In the illustrative embodiment, the session ID acquisition logic 90 waits to process two identical session ID 76 values in a row during a



reconstruction portion of the acquisition process, determines whether an acceptable number of good packets are detected during a predetermined number of total subsequent recovered track packets during a verification portion of the acquisition process, and adopts the session ID 76 as the
5 acquired session ID 78 if an acceptable number of good packets are detected during this period. It will be appreciated by those skilled in the art that the number of required identical reconstructed packet session ID's detected during the reconstruction portion of the acquisition process, and the number of required good packets recovered during the verification portion of
10 the acquisition process, are adjustable.

In a preferred embodiment, the session ID may also be acquired when the drive is operating in a high-speed search mode – that is, the tape is being transported at greater than normal read speed and data is not actually being read from the tape. FIG. 4(a) is a flowchart of one
15 embodiment of a high-speed search session ID acquisition method 400. FIG. 4(b) is a diagram illustrating the high-speed search session ID acquisition process. According to the preferred embodiment for high-speed search mode session ID acquisition, session ID acquisition logic 90 monitors only control packets recovered from tape at high speed. Each control packet
20 40 includes a control packet address 30, a control packet body 32 that contains control information including the session ID 34 that matches that of the recording session when the control packet was recorded, a packet CRC field 16 and a packet ECC field 18. Packet detection logic 70 detects control packets 40 and places them into a temporary packet buffer location 72.
25 Packet correction logic 74 reconstructs the packet session ID 34 for each

recovered control packet 40. Session ID acquisition logic 90 processes the recovered control packets 40 and associated reconstructed packet session ID's 32 to acquire the original session ID 78 of a particular portion of tape.

In the preferred embodiment, a first control packet is recovered and
5 its associated packet session ID is reconstructed in a step 402. The reconstructed packet session ID is maintained as a possible session ID 76. In a step 404, a second control packet is recovered and its associated packet session ID is reconstructed. The second reconstructed packet session ID is compared to the possible session ID 76 in a step 406. If the
10 two session ID's do not match, the process starts over again with step 402. If the two session ID's do match, however, the session ID acquisition logic 90 waits a predetermined (preferably programmable) number of drum revolutions in step 408. A third control packet 40 is recovered in step 410 and its associated packet session ID is reconstructed. The third packet
15 session ID is compared to the possible session ID in step 412. If the third packet session ID does not match the possible session ID, the process starts over again with step 402. If the third session ID does match the possible session ID, a fourth control packet 40 is recovered in step 414 and its associated packet session ID is reconstructed. The fourth packet session ID
20 is compared to the possible session ID in step 416. If the third packet session ID does not match the possible session ID, the process starts over again with step 402. If the fourth session ID does match the possible session ID, the possible session ID is adopted as the acquired session ID in step 418 and the method 400 is complete. Once a session ID is acquired, it
25 is used until a programmable number of control packets have failed to be

read. When this condition occurs, the logic repeats the session ID acquisition.

In the preferred embodiment, the tape drive has the capability for allowing a user to set the session ID, thereby overriding an automatically
5 acquired session ID.

FIG. 5(a) is a flowchart of one embodiment 500 of a method in accordance with the invention for logically rejecting residue data during a data recovery session. The method 500 begins by first acquiring a reference session ID in step 502. The reference session ID is acquired, for example,
10 using the method of FIG. 3(a) or 4(a). In a step 504, the reference session ID is preloaded into the CRC and ECC syndrome generators 82 and 84. A track packet 8 is recovered from tape during a step 506. In step 508, the track packet 8 is simultaneously copied into a temporary packet buffer location 72 and sent to the CRC and ECC syndrome generators 82 and 84
15 to detect any errors in the local packet address field 12 or data field 14 of the packet. If an error is detected in the address or data field of the packet, it is corrected by packet correction logic 74 in step 510. Error correction is performed on the corrected packet in step 512 in case the packet was miscorrected due to being encoded with a residue session ID. If an error is
20 detected, the packet is discarded in step 516 and not sent to the data buffer because it would have been corrected correctly by the packet correction logic 74 if its original packet session ID as encoded in the packet CRC field 16 and packet ECC field 18 matched that of the reference session ID. Accordingly, the packet is inherently rejected as unintended to be read and
25 is not copied from the temporary packet buffer location 72 for further

processing. However, if the CRC and ECC syndrome generators 82 and 84 do not initially detect any errors in step 508, or after the packet has been corrected in step 512, the packet is considered good and sent to the data buffer in step 514. Steps 506 through 516 are repeated for each subsequent
5 recovered track packet 8 of the current data recovery session.

FIG. 5(b) is a diagram illustrating the process of FIG. 5(a). During a data recovery session, once the session ID for the current session is acquired, data packets 8 and control packets 40 are checked by packet detection logic 70 while they are written into temporary packet buffer location
10 72. Packet detection logic 70 notifies the packet correction logic 74 when a packet 8 is available in the temporary buffer location 72. Packet correction logic 74 corrects any error in the address field 12 or data field 14 of packets 8 that fail checks by the CRC generator 82 and Reed-Solomon ECC generator 84 that were performed by the packet detection logic 70. If error
15 correction was performed on a packet 8, CRC and ECC error detection is performed on the corrected packet by CRC generator 82 and ECC generator 84 as it is being sent to its location in the data buffer 96 in case the packet was miscorrected due to being encoded with a residue session ID. If the packet 8 was miscorrected, the error detection circuitry 82 and 84 will detect
20 this condition and the miscorrected packet 8 is discarded.

It will be appreciated by those skilled in the art that the present invention provides several advantages over the prior art. First, the need for a costly erase head and associated control hardware is eliminated. Secondly, track packet rejection is more efficient since, once the reference
25 session ID is acquired at the start of a data recovery session, the packet

session ID need not be reconstructed for every subsequent recovered track packet. Finally, the same error detection hardware used to detect packet errors and same error correction hardware used to correct packet errors is used to acquire the session ID, thereby eliminating the requirement for
5 separate session ID acquisition hardware.

Although the invention has been described in terms of the illustrative embodiments, it will be appreciated by those skilled in the art that various changes and modifications may be made to the illustrative embodiments without departing from the spirit or scope of the invention. It is intended that
10 the scope of the invention not be limited in any way to the illustrative embodiment shown and described but that the invention be limited only by the claims appended hereto.

**WHAT IS CLAIMED IS:**

1. A method for logically rejecting previously recorded recovered track packets from a magnetic media, comprising:

recovering a track packet (8) from said magnetic media, said track packet comprising a data field (14), an error detection field (16, 18), and a packet session identifier (ID) (52) encoded in said error detection field (16, 18);

preloading (504) an error detection code generator (82, 84) with a reference session ID (78), said error detection code generator for generating an error detection code based on said reference session ID across said track packet, said error detection code indicating a packet error if said packet session ID (52) does not match said reference session ID (78); and

rejecting (516) said track packet (8) as invalid if said error detection code indicates a packet error.

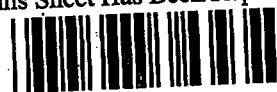
15

2. A method in accordance with claim 1, comprising:
acquiring said reference session ID (502).

3. A method in accordance with claim 1 or 2, wherein:
said error detection field comprises a cyclical redundancy code (CRC) (16) with said packet session ID (52) encoded in said CRC.

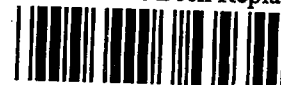
4. A method in accordance with claim 1, 2, or 3, wherein:
said error detection field comprises an error correction code (ECC) (18) with said packet session ID (52) encoded in said ECC.

5. A method in accordance with claim 2, 3 or 4, wherein:
said acquiring step (502) comprises:
recovering a predetermined number of consecutive track packets that have identical packet session ID's (52); and
adopting said identical packet session ID (52) as said reference session ID (78).





6. A method in accordance with claim 5, wherein:
said recovering step comprises:
recovering (304) a first track packet;
reconstructing (308) said packet session ID (52) of said first track
5 packet (8);
holding said reconstructed packet session ID (52) of said first track packet as
a possible session ID (76);
recovering (304) a next track packet;
reconstructing (308) said packet session ID (52) of said next track
10 packet;
comparing said reconstructed packet session ID (52) of said next
track packet to said possible session ID (76);
repeating said first track packet recovery step through said
comparison step if said reconstructed packet session ID of said next track
15 packet does not match said possible session ID;
determining (324) whether a predetermined number of consecutive
packets have been recovered since said first track packet was recovered;
repeating said next track packet recovering step through said
determining step if said predetermined number of consecutive packets have
20 not yet been recovered; and
adopting (326) said possible session ID (76) as said acquired session
ID (78) if said predetermined number of consecutive packets have not yet
been recovered.
- 25 7. A method in accordance with claim 1, 2, 3, 4, 5 or 6, comprising:
recording said track packet onto said magnetic media during an
earlier recording session, said recording session comprising the steps of:
generating (202) a recording session ID at a start of said
recording session;
30 formatting data into said data field (14);
generating (204) said error detection field (16, 18) over said
recording session ID (52) and said data field (14), said recording





session ID (52) becoming said packet session ID of said track packet (8);

formatting (204) said data field (14) and said error detection field (16, 18) into said track packet (8); and

5 recording (208) said track packet (8) onto said magnetic media.

8. An apparatus for logically rejecting previously recorded recovered track packets from a magnetic media, comprising:

10 a read channel which recovers a track packet (8) from said magnetic media;

a packet detector (70) which extracts from said track packet (8) a data field (14), an error detection field (16, 18), and a packet session identifier (ID) (78) encoded in said error detection field;

15 an error detection code generator (84, 82) preloaded with a reference session ID (78) which uses said reference session ID to detect an error in said track packet (8) and generates an error detection code indicating whether or not said packet session ID matches said reference session ID; and

20 wherein said packet detector (70) rejects said track packet (8) as invalid if said error detection code indicates a packet error.

9. An apparatus in accordance with claim 8, comprising:

a reference session identifier (ID) encoder which encodes a reference session ID in each of a plurality of track packets recorded onto said magnetic media during a single recording session, said reference session ID encoder comprising:

a recording session ID generator (50) which generates a current recording session ID (52), said current recording session ID (52) being unique from prior recording session IDs of prior recording sessions to said magnetic media; and





an error correction code (ECC) generator (54, 58) which
generates an error correction field (16, 18) for each said plurality of
track packets (8), wherein each of said error correction field (16, 18) is
generated across said current recording session ID (52) and said
5 respective track packet (8), and is then appended to said respective
track packet (8).

10. An apparatus in accordance with claim 8 or 9, comprising:
- 10 a reference session ID acquisition circuit (90) which generates a
possible recording session ID (76) by erasing a packet session ID (20) of a
first track packet (8), performing error correction on said first track packet (8)
to reconstruct said packet session ID (52), using said reconstructed packet
session ID as said possible recording session ID (76), setting said acquired
session ID (78) to said possible recording session ID (76) if a predetermined
15 number of subsequent recovered track packets have a packet session ID
(52) that matches said acquired session ID (78).

This Sheet Has Been Replaced

