

- 1 -

TITLE OF THE INVENTION

A Method and Apparatus for Management of Faulty Data
in a RAID System

Background of the Invention

The invention pertains to the field of fault tolerant arrays of hard disks that are known as Redundant Arrays of Inexpensive Disks (RAID). More specifically the invention relates to the management and correction of undetected read errors in a RAID system that are discovered when a disk failure requires the replacement and build-up of a disk other than that containing the undetected read error.

Summary of the Invention

This invention provides a faulty data management subsystem that avoids the potential generation of spurious data when two bad data blocks exist within the same RAID data group because of a disk recording medium error in one channel of the array and the failure, replacement, and rebuild of a disk drive on another channel. The faulty data management subsystem of the invention includes:

- a) a disk rebuild routine for rebuilding a failed drive that checks for a bad data block amongst the channels being used to reconstruct the data on the failed disk and, if found, scrubbing the bad data block by writing a filler code in that block and in the corresponding block of the disk that is being rebuilt;
- b) computing a new parity block corresponding to the bad data block by using all channels including those with filler code;
- c) updating a bad data table that lists all bad data blocks; and

- 2 -

d) checking each disk array access request to determine if the data address of that request is listed in the bad data table, and, if not, allowing the access to proceed, otherwise
5 checking if the access request is for a write and, if so, allowing the write, deleting the block address from the bad data table and generating a new parity block, and, if not, generating an error signal.

10 The present invention provides means for managing faulty data in RAID systems without incurring the potential problems described above.

Because a byte stripe is a single address block, it may be seen that a RAID-3 and RAID-4 system are the
15 same except for block length. Consequently, all following references to a block will be understood to include the RAID-3 byte stripe unless otherwise indicated.

Also, "RAID system" will be used to indicate RAID-
20 3, -4 and -5 systems unless indicated otherwise.

RAID systems, based on magnetic disk technology developed for personal computers, offer an attractive alternative to single large expensive disk memories by providing improved performance, reliability, and power
25 consumption. The manufacturers of small disks can offer such performance because of the efforts at standardization in defining higher level peripheral interfaces such as the ANSI X3.131-1986 Small Computer Synchronous Interface (SCSI). This has led to the
30 development of arrays of inexpensive disks which are organized in an interleaved manner for large block transfers of data, or arranged for independent parallel access for small transaction processing.

The formation of large disk arrays introduces
35 reliability problems which result from the use of a large

- 3 -

multiplicity of electromechanical devices. The mean-time-to-failure (MTTF) for an array has been estimated to increase with the number of disks in an array (Patterson, D.A., Gibson, G., and Katz, R.H., "A Case for Redundant
5 Arrays of Inexpensive Disks (RAID), Report No. UCB/CSD 87/391, Dec. 1987, Computer Science Division (EECS), University of California, Berkeley, CA 94720).

Brief Description of the Drawings

The present invention may be more fully understood
10 from the detailed description given below and from the accompanying drawings of the preferred embodiments of the invention, which, however should not be taken to limit the invention to the specific embodiment but are for explanation and understanding only.

15 FIG. 1 is a block diagram of a prior art MxN disk array, including a disk array controller.

FIG. 2 shows the prior art logical disk area configuration based on the array of FIG. 1.

FIG. 3(a) shows the memory map of a prior art
20 RAID-1 system.

FIG. 3(b) shows the memory map of a prior art RAID-3 system.

FIG. 3(c) shows the memory map of a prior art RAID-4 system.

25 FIG. 3(D) shows the memory map of a prior art RAID-5 system.

FIGS. 4(a), 4(b), and 4(c) show an initial RAID-1 memory map, a double faulted memory RAID-1 memory map, and a rebuilt, but double faulted RAID-1 memory map,
30 respectively, according to the prior art.

FIGS. 5(a), 5(b), and 5(c) show an initial RAID-5 memory map, a double faulted RAID-5 memory map, and a rebuilt, but double faulted, RAID-5 memory map, respectively, according to the prior art.

- 4 -

FIG. 6 is a flow diagram of the disk rebuild and reconstruct method of the invention.

FIG. 7 is a flow diagram of the method of the invention for accessing the array.

5 FIG. 8 shows the hardware configuration of the invention for a RAID faulty data management system.

Description of the Preferred Embodiments

FIG. 6 is a flow diagram of the disk rebuild and reconstruction method 100 used to circumvent the problem
10 of generating spurious data because a disk read fault has occurred on a channel of a RAID system while reconstructing a corresponding data block on another channel. The reconstruction of a data block or rebuild of a disk drive is accomplished by bit-by-bit EXORing of
15 the corresponding bits of the remaining blocks belonging to the same group that share a common parity block.

Disk rebuild and reconstruction method 100 begins at step 101 in which the EXOR method of reconstruction begins. During reconstruction, step 102 checks for an
20 indication that one of the blocks needed for reconstruction is bad. If the reconstruction is completed without detecting a bad block, the process terminates at step 112. Otherwise, step 105 checks if the bad data block needed for reconstruction is a parity
25 block. If so, step 104 writes a filler block to the block that is to be reconstructed and step 105 updates the bad data table (BDT) by recording that the data block is non-recoverable (bad). (The filler block may contain any convenient alphanumeric code.) If the bad data block
30 of step 103 is not a parity block, step 106 scrubs the bad data block by writing a filler block to replace the bad data block and, in step 107, also writes a filler block to the corresponding block that was to be reconstructed. Step 107 updates the BDT by recording
35 that both the block in step 103 and the corresponding

- 5 -

block of the channel being reconstructed is bad. Each entry in the BDT identifies a block as containing artificial data and hence non-recoverable data. Step 109 computes new parity data by EXORing all of the data
5 channel blocks (including the filler blocks). Step 100 writes a new parity block and step 100 resumes the reconstruction process by returning to step 102.

The procedure of Fig. 6 ensures that

(1) if a read error occurs during reconstruction
10 in one of the data blocks, all of the remaining "good" data blocks are recoverable, and

(2) the artificial data of the filler blocks can not be confused with real data.

Step 109 updates a "bad data table", or BDT, that
15 is located in disk controller 12, and may also be located in a reserved section of each of the disks. The BDT contains a list of all bad data blocks (filler blocks) so that any read request to the disk array first compares the target address with the addresses listed in the BDT
20 and returns an appropriate error signal to the requesting agent if the address is listed. After the BDT update, the process of reconstruction continues in step 110, and the monitoring in step 104 continues until completed. If additional bad data blocks are found in step 104, steps
25 105-110 are repeated as required.

The method for operating a comparator for checking array access 200, shown in the flow diagram of FIG. 7, is used to manage the use of a BDT in conjunction with the RAID memory system. The method is invoked whenever an
30 access is requested and begins in step 201 by checking if an access request is to a bad data block whose address is listed in the BDT, and, if not, proceeds with the access in the normal operating mode. If the address is listed in the BDT, step 203 determines if the access is for a
35 write operation, and, if not, an error flag is generated

- 6 -

in step 204. The error flag advises the host system that the read request is to a non-recoverable data block. If the request is for a write access, step 205 deletes the block address from the BDT and permits the write

5 circuitry to write the new data to the block address in step 206, thus scrubbing the bad data block status. Step 207 is for computing the new block parity and writing it into the corresponding parity block associated with the accessed data group.

10 FIG. 8 is a block diagram that shows the architecture of a RAID system of this invention using the methods outlined in FIGS. 6 and 7. Typically, RAID system 20 with faulty data management of the type described above is interfaced through its controller 22
15 from a SCSI bus 21. (SCSI stands for a well known industry standard bus as described in "Small Computer System Interface-2", American National Standard X3T9.2/86-109). Bus 21 connects RAID system 20 to the host system. Disk array controller 22 provides the
20 necessary logic to map any logical address associated with a memory access into the corresponding physical address, directs data traffic, controls the operation of each disk drive and provides status information to the host. BDT 23 is shown coupled to controller 22 but may
25 also be an integral part of an integrated chip disk array controller. Also, each disk 11 of the array is shown having a local BDT 24 in which the bad data information for each disk is optionally stored so that an access address can be checked at the disk level rather than at
30 the array level in disk controller 22.

As will be understood by those skilled in the art, many changes in the methods and apparatus described above may be made by the skilled practitioner without departing from the spirit and scope of the invention, which should
35 be limited only as set forth in the claims which follow.



- 7 -

Claims

What is claimed is:

1. Apparatus for managing faulty data in a multichannel memory system having a memory array
5 controller for controlling access to the memory system, for memory failure detection, for memory error detection and for single channel correction, each channel having at least one memory module, each module having a failure and read error detection means, and each memory module being
10 separately replaceable upon failure, the apparatus, accessible to the memory controller, comprising:
 - a) a bad data table (BDT) for storing addresses of non-recoverable data due to the occurrence of concurrent faults in more than one channel;
 - 15 b) a write circuit for writing filler data to a bad block location for replacing a bad data block and for writing to the BDT for storing addresses of bad data blocks representing non-recoverable data; and
 - c) detection circuitry for detecting memory
20 access requests to addresses stored in the BDT, for returning a non-recoverable data error signal to a host system if the access request is a read request, and, if the access request is a write request, permitting the write to the BDT listed address and deleting the listed
25 address from the BDT.
2. The apparatus of claim 1 wherein the multichannel memory system is a RAID-3 memory array.
3. The apparatus of claim 1 wherein the multichannel memory system is a RAID-4 memory array.
- 30 4. The apparatus of claim 1 wherein the multichannel memory system is a RAID-5 memory array.
5. The apparatus of claim 1 wherein the BDT is part of the memory array controller.
6. The apparatus of claim 1 wherein the BDT is
35 stored in a reserved section of each memory module.



- 8 -

7. The apparatus of claim 1 wherein the BDT is stored in a memory in the memory array controller and in a reserved section of each memory module.

8. A method for managing faulty data in a
5 multichannel memory system having memory error detection and single channel correction means, each channel having at least one memory module capable of detecting failure and read errors, each memory module being separately replaceable upon failure, the method comprising:
- 10 a) detecting a read error in a block of a given channel;
- b) determining if the read error is due to a failure of a memory module and, if so, replacing the failed memory module, otherwise
15 using the single channel correction means for reconstructing data in the given channel;
- c) monitoring all other channels for a second read error that would prevent reconstructing data, and, if such a second read error is
20 found and, if neither the first nor second block is a parity block, entering the second block address, and the corresponding address of the first block into a bad data table (BDT), writing a filler block to both the
25 first and the second blocks, computing and replacing a corresponding parity block using the first and the second blocks together with all corresponding data blocks, otherwise, writing a filler block to whichever of the
30 first block and the second block is a data block, computing and writing a corresponding parity block using the filler block with all corresponding data blocks, and then continuing reconstructing data in the given
35 channel until finished; and



- 9 -

d) monitoring all memory access requests by comparing the requested address to the BDT listed addresses, and, if the address is not listed in the BDT, allowing the access request to proceed, otherwise checking if the request is a write request, and, if not, returning a non-recoverable data error signal, otherwise allowing the write request to proceed so that the addressed filler block is replaced by a valid data block, and then computing and entering a new corresponding parity data block.

9. The method of claim 8 wherein the multichannel memory system is a RAID-3 memory.

10. The method of claim 8 wherein the multichannel memory system is a RAID-4 memory array.

11. The method of claim 8 wherein the multichannel memory system is a RAID-4 memory array.

12. A multichannel memory array system with a faulty data management system handling multiple faults, comprising:

- a) a memory array wherein each channel has at least one memory module, each memory module being separately replaceable upon failure, and each module having a failure and read error detector;
- b) a memory array controller for coupling the memory array to a memory bus, for controlling access to the memory array, for memory array failure detection and single channel correction and for communicating memory status over the memory bus, the memory array controller further having,
 - i) a bad data table (BDT) for storing addresses of non-recoverable data blocks



- 10 -

due to the occurrence of concurrent faults in more than one channel,

- 5 ii) write circuitry for writing filler data blocks to bad data blocks corresponding to non-recoverable data blocks and for storing non-recoverable data block addresses in the BDT, and
- 10 iii) a comparator for detecting memory access requests to bad data blocks by comparing a memory access request address with the BDT stored addresses, for returning a non-recoverable error signal to a requesting agent if the access request is a read request and the address is
- 15 stored in the BDT, and for permitting the access request to proceed if the request is for a write access.

13. The multichannel memory array system of claim 12 wherein each memory module has a reserved memory area
20 for storing the BDT.

14. The multichannel memory array system of claim 12 wherein the multichannel memory array system is a RAID-3 memory system.

25 15. The multichannel memory array system of claim 12 wherein the multichannel memory array system is a RAID-4 memory system.

16. The multichannel memory array system of claim 12 wherein the multichannel memory array system is a RAID-5 memory system.