RAID 5 Support on SCSI Disk Drives

Rev. 1.1
5-2-94
Seagate Technology

Contents

INTRODUCTION 5

RAID 5 SUPPORT FOR MULTIPLE INTERFACE STRIPING CONFIGURATIONS 6

READ-MODIFY-WRITE Function 7

REGENERATE Function 8

REBUILD Function 9

RAID 5 SUPPORT FOR SINGLE INTERFACE STRIPING CONFIGURATIONS 10

READ-MODIFY-WRITE Function 11

REGENERATE Function 12

REBUILD Function 13

NEW RAID SPECIFIC SCSI COMMANDS 15

XDWRITE Command 15

XPWRITE Command 18

XDREAD Command 20

REBUILD Command 21

REGENERATE Command 24

ADDITIONAL PERFORMANCE ENHANCEMENTS 27

Log Control Page 27

LCLEAR Command 28

Table of Figures

FIGURE 1: PARALLEL READ-MODIFY-WRITE 7
FIGURE 2: PARALLEL REGENERATE 8
FIGURE 3: PARALLEL REBUILD 9
FIGURE 4: SINGLE INTERFACE READ-MODIFY-WRITE 11
FIGURE 5: SINGLE INTERFACE REGENERATE 12
FIGURE 6: SINGLE INTERFACE REBUILD 13

Table of Tables

TABLE 1: XDWRITE COMMAND 17
TABLE 2: XPWRITE COMMAND 19
TABLE 3: XDREAD COMMAND 20
TABLE 4: REBUILD COMMAND 22
TABLE 5: REBUILD COMMAND PARAMETERS 23
TABLE 6: REGENERATE COMMAND 25
TABLE 7: REGENERATE COMMAND PARAMETERS 26

Note:
For Questions or Comments on this document Contact: Mike Miller at (612) 844-5924 or Email-Mike_Miller@notes.seagate.com

Introduction

In recent years advances in processor performance have far outpaced advances in I/O performance. In addition, the requirement to have 100% on-line protection of user data has become more prevalent. Many companies have recognized this and have developed subsystem architectures that attempt to deal with these problems. One of the most prevalent techniques has been to incorporate a RAID 5 architecture into the subsystem. This is usually done with a special disc controller that has additional hardware functionality to support the RAID 5 architectural requirements. These controllers typically incorporate multiple SCSI buses and an XOR engine for parity generation. If an XOR engine were incorporated into the drive the RAID 5 architecture could be accomplished using standard SCSI controllers. Incorporation of an XOR engine into the Disc drives has been done or at least considered many times in the past. With the advent of higher performance interfaces (such as Fibre Channel) it now makes more sense than ever before to include this functionality in the drive.

This document describes the XOR functionality that is designed into the drive and it shows how this functionality can be used by a RAID controller to accomplish some of the tasks it needs to perform to implement a RAID 5 architecture. The first part of the document describes the XOR related tasks that need to be performed in an environment where multiple SCSI busses are used to control the drives. The second part of the document describes how these same functions would be performed in a single bus architecture (such as Fibre Channel). The third part of the document describes the new SCSI commands that are needed to perform the XOR functions. The forth part of the document describes some additional drive functionality that will increase the performance of write operations in a RAID 5 environment.

RAID 5 Support for Multiple Interface Striping Configurations

There are three primary operations that need to be supported to eliminate the XOR engine from the controller. They are, Read-Modify-Write, Regenerate, & Rebuild. The Read-Modify-Write function is used any time a write operation needs to be performed on the array. The Regenerate function is used to regenerate data from the array when a data drive has malfunctioned. In this situation the Regenerate function takes the place of a normal Read operation. The Rebuild function is used to rebuild a drive that has been added to the array in place of a failed drive. The Rebuilt data is written on the new drive as part of this function.

In multiple interface striping configurations the drives do not necessarily have the capability of peer to peer communication. In this environment all data must pass through the host. This section of the document shows how the three basic Functions (described above) are performed using the XOR engine in the drive along with the new SCSI commands.

READ-MODIFY-WRITE Function

(Drives on separate buses or loops)

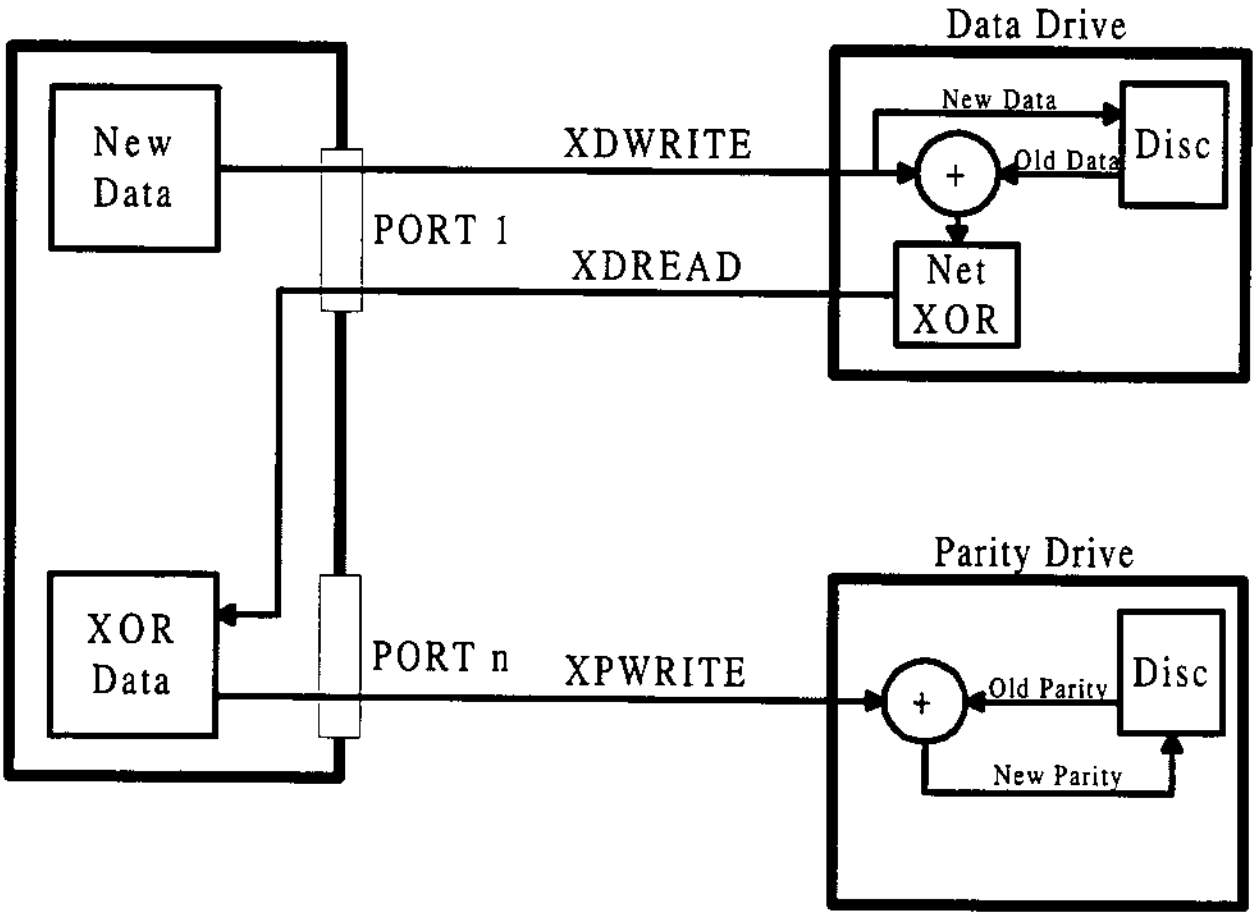
The Read-Modify-Write function is used to write new data to the array. Three new SCSI commands are used to accomplish this. They are XDWRITE, XDREAD, & XPWRITE.

The host begins by sending the data drive new data using an XDWRITE command, with the Buffer bit set to indicate that the host will later be retrieving xor'd data. It also sends the parity drive the XPWRITE command (command phase only at this point - this gets the parity drive started reading old parity from the disc into its buffer). The data drive reads old data from the disc into its buffer, exclusive ors the old data with the new data from the host, stores the xor'd data in its buffer, and writes the new data from the host to the disc. Good XDWRITE ending status is not sent to the host until the exclusive or'd data is available in the buffer.

The host reads the xor'd data by sending the data drive an XDREAD command with the same logical block address and transfer length as in the associated XDWRITE command. (The data drive is required to retain the xor'd data until successfully completing the associated XDREAD command.)

The host then sends the xor'd data to the parity drive during the data phase of the already issued XPWRITE command. The parity drive exclusive ors this data with the old parity data in its buffer. The resulting new parity data is written to the disc.

Figure 1: PARALLEL READ-MODIFY-WRITE



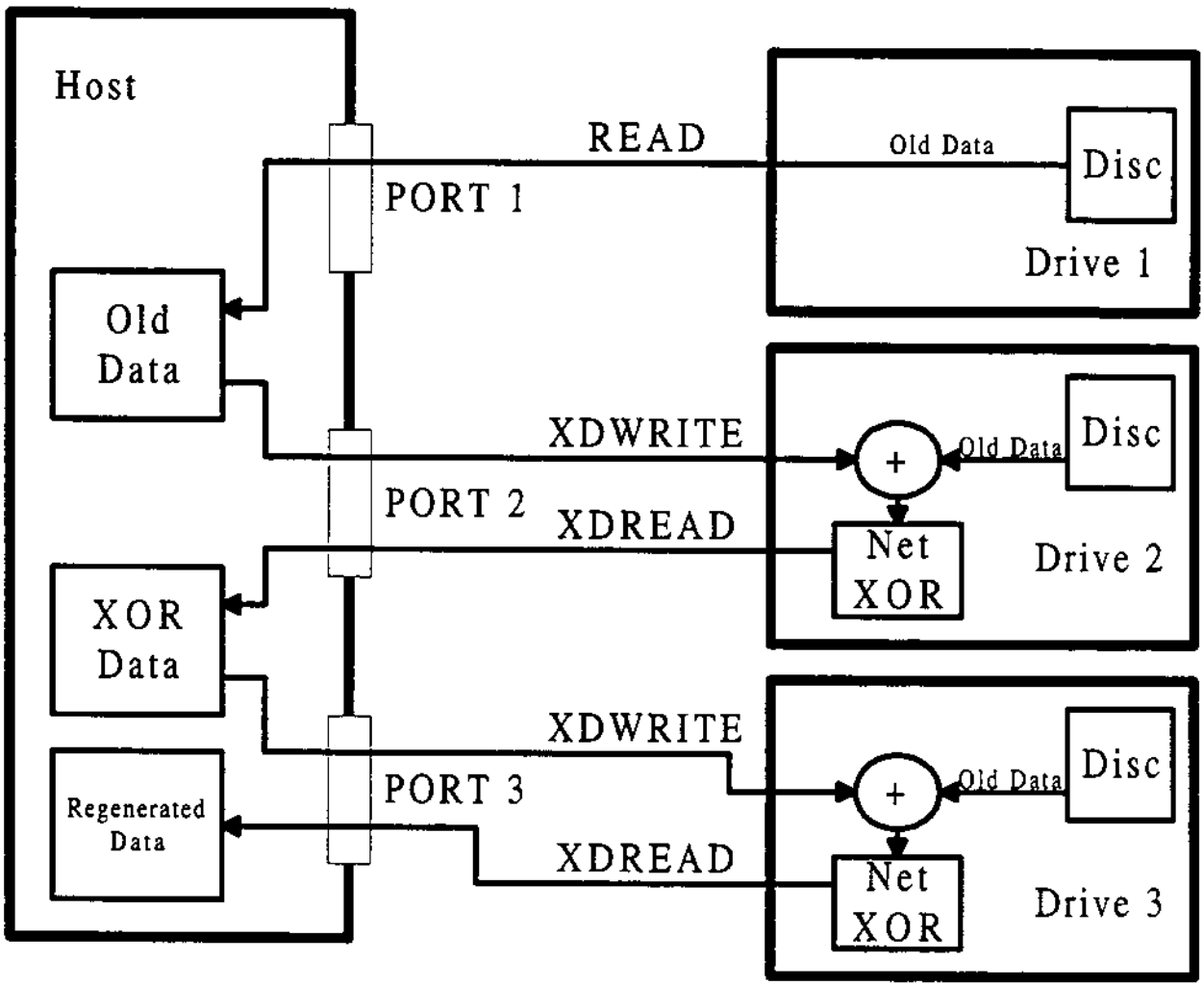
REGENERATE Function

(Drives on separate buses or loops)

This function is used in place of a Read command when one data drive in the array has malfunctioned. The host begins by sending the first source drive a READ command. The data received from this drive is sent to the second source drive using an XDWRITE command with the NDISC bit set (to prevent data from being written to disc). The second source drive reads old data from the disc, exclusive ors this old data with the data sent from the host, and stores the result in its buffer. The host retrieves the xor'd data by sending the second drive an XDREAD command with the same logical block address and transfer length as in the associated XDWRITE command. (The second drive is required to retain the xor'd data until successfully completing the associated XDREAD command.) The host issues the XDWRITE and XDREAD commands in a likewise manner for each successive source drive.

The xor'd data from the last source drive is the regenerated data.

Figure 2: PARALLEL REGENERATE



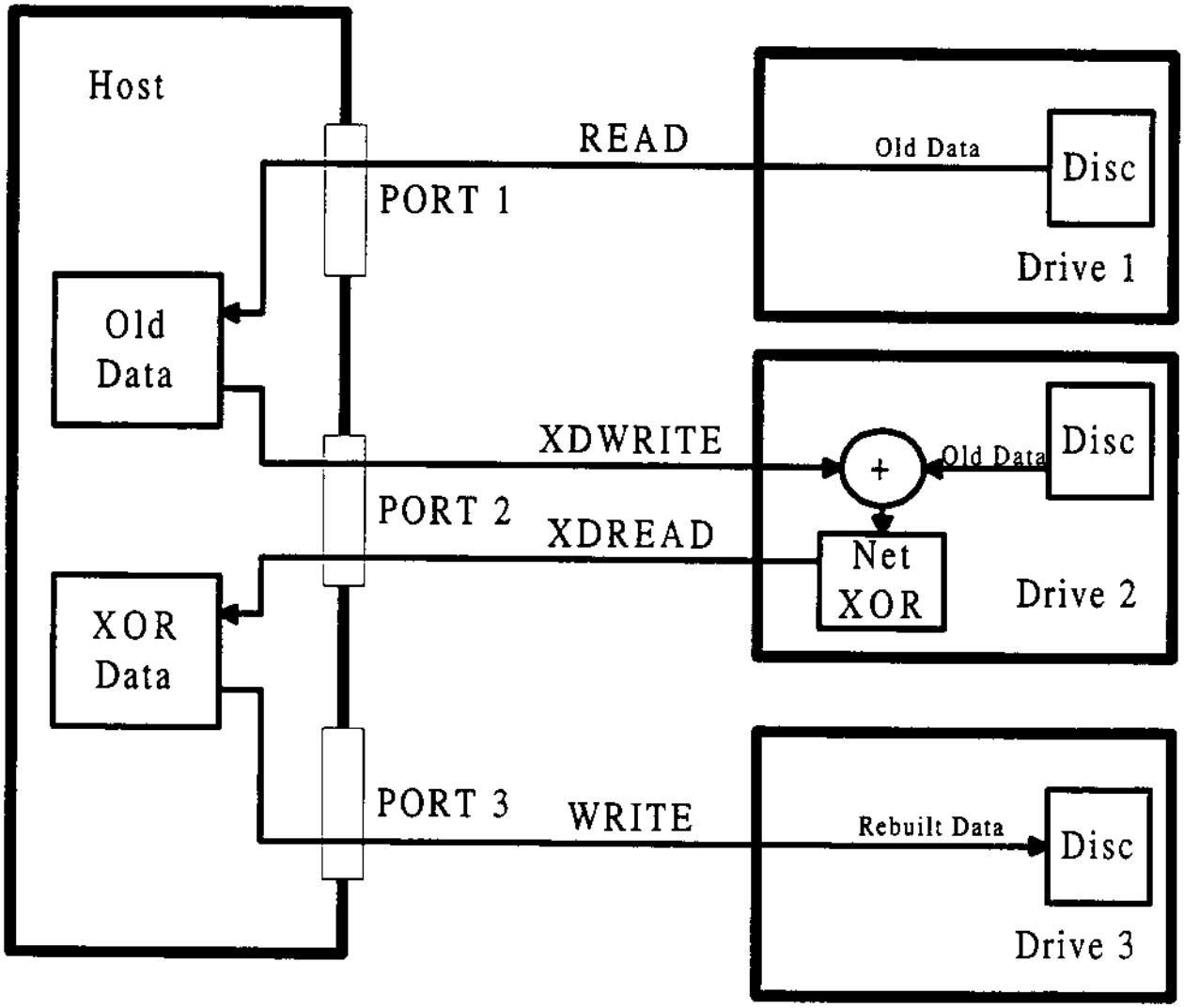
REBUILD Function

(Drives on separate buses or loops)

The host begins by sending the first source drive a READ command. The data received from this drive is sent to the second source drive using an XDWRITE command with the NDISC bit set (to prevent data from being written to disc). The second source drive reads old data from the disc, exclusive ors this old data with the data sent from the host, and stores the result in its buffer. The host retrieves the xor'd data by sending the second drive an XDREAD command with the same logical block address and transfer length as in the associated XDWRITE command. (The second drive is required to retain the xor'd data until successfully completing the XDREAD command.) The host issues the XDWRITE and XDREAD commands in a likewise manner for each successive source drive.

The xor'd data from the last source drive is the "Rebuilt" data, and is sent to the drive being rebuilt using a WRITE command.

Figure 3: PARALLEL REBUILD



RAID 5 Support for Single Interface Striping Configurations

As with the Multiple Interface Striping Configuration the Single Interface Striping Configuration needs to support three basic operations to eliminate the XOR engine from the controller. They are, Read-Modify-Write, Regenerate, & Rebuild. The Read-Modify-Write function is used any time a write operation needs to be performed on the array. The Regenerate function is used to Build data from the array when a data drive has malfunctioned. In this situation the Regenerate function takes the place of a normal Read operation. The Rebuild function is used to rebuild a drive that has been added to the array in place of a failed drive. The Rebuilt data is written on the new drive as part of this function.

In Single Interface Striping Configurations the drives have the capability of peer to peer communication. In this environment some of the data passes directly from drive to drive. This technique greatly reduces the amount of work that must be performed in the host as well as reduces the amount of data that must be transferred over the interface. For instance for Read-Modify-Write operations the number of data transfers is reduced from 4 to 2. This section of the document shows how the three basic Functions (described above) are performed using the XOR engine in the drive along with the new SCSI commands.

READ-MODIFY-WRITE Function

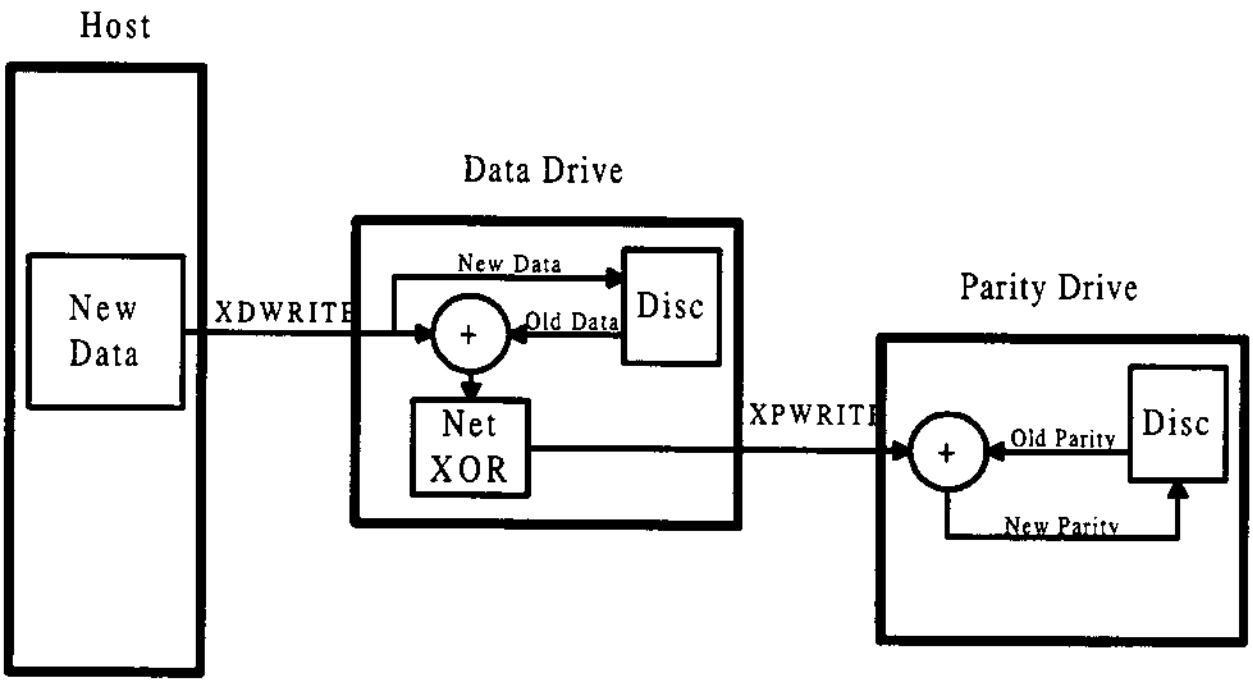
(Drives on <u>same</u> bus or loop)

The host begins by sending the data drive new data using an XDWRITE command, with the Buffer bit clear to indicate a secondary target (parity drive) will be involved. Included in the command descriptor block for the XDWRITE command is the address of the parity drive associated with the data block(s) being written. The data drive immediately disconnects, takes on the role of initiator, and sends an XPWRITE command to the parity drive (command phase only at this point - this gets the parity drive started reading old parity from the disc into its buffer). The data drive reads old data from the disc into its buffer, exclusive ors the old data with the new data from the host, sends the xor'd data to the parity drive (data out phase of XPWRITE), and writes the new data from the host to the disc.

The parity drive receives the exclusive or'd data from the data drive, and exclusive ors this data with the old parity data in the buffer. The resulting new parity data is written to the disc.

Upon successful completion of the XPWRITE command good XDWRITE ending status is sent by the data drive to the host. The XPWRITE command is thus "nested" within the XDWRITE command.

Figure 4: SINGLE INTERFACE READ-MODIFY-WRITE

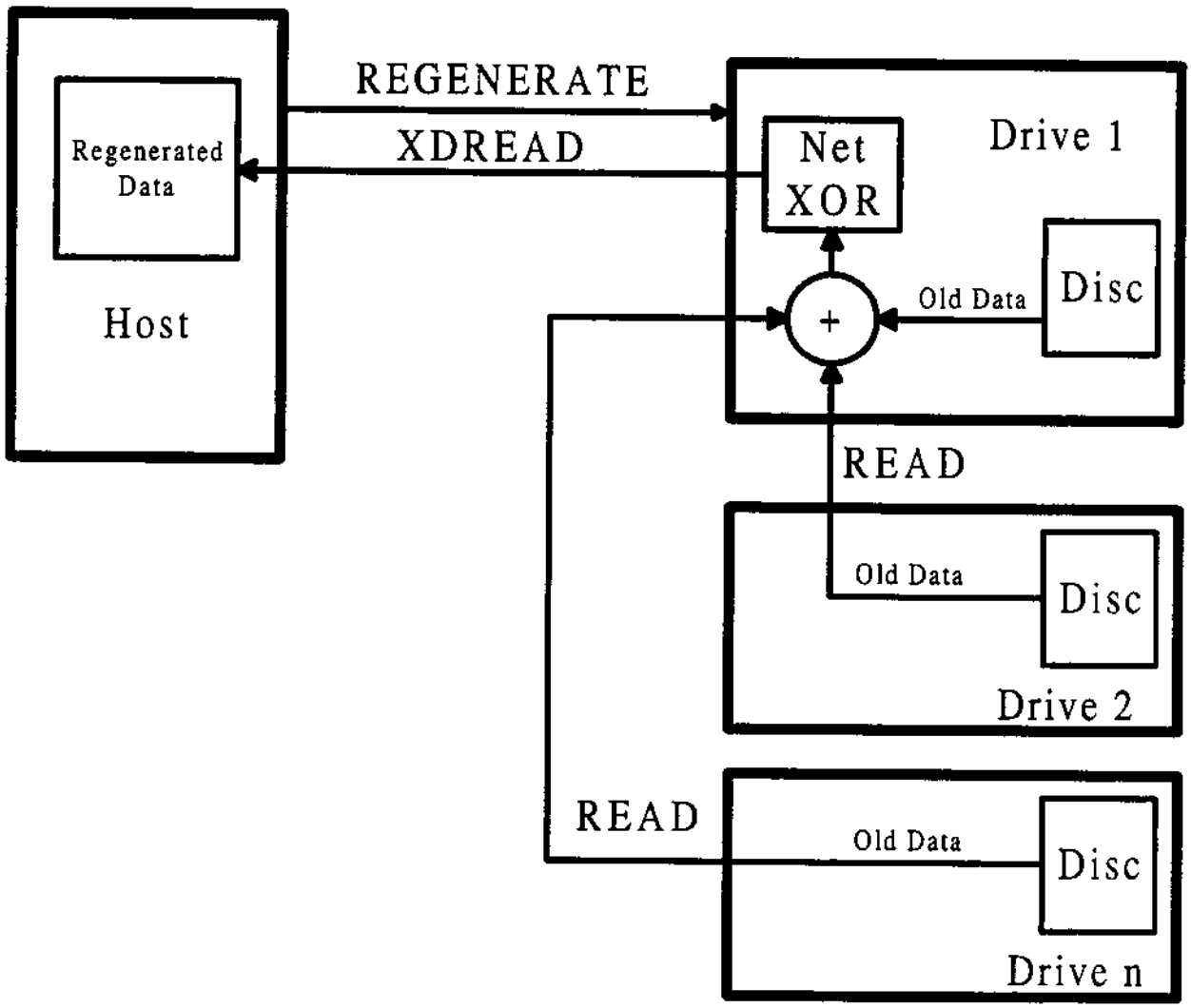


REGENERATE Function

(Drives on same bus or loop)

This function is used in place of a Read command when one data drive in the array has malfunctioned. The host begins by sending a known good drive the REGENERATE command. The addresses of the source drives, as well as the logical block address and regenerate length are passed during the data out phase of the REGENERATE command. The drive then takes on the role of initiator and sends READ commands to all source drives. It also concurrently reads the appropriate data from its own disc. The data from all drives is exclusive or'd and written to the drive's buffer. The host retrieves this built data by sending the drive an XDREAD command with the same logical block address and build length as in the associated REGENERATE command. (The drive is required to retain the xor'd data until successfully completing the associated XDREAD command.)

Figure 5: SINGLE INTERFACE REGENERATE

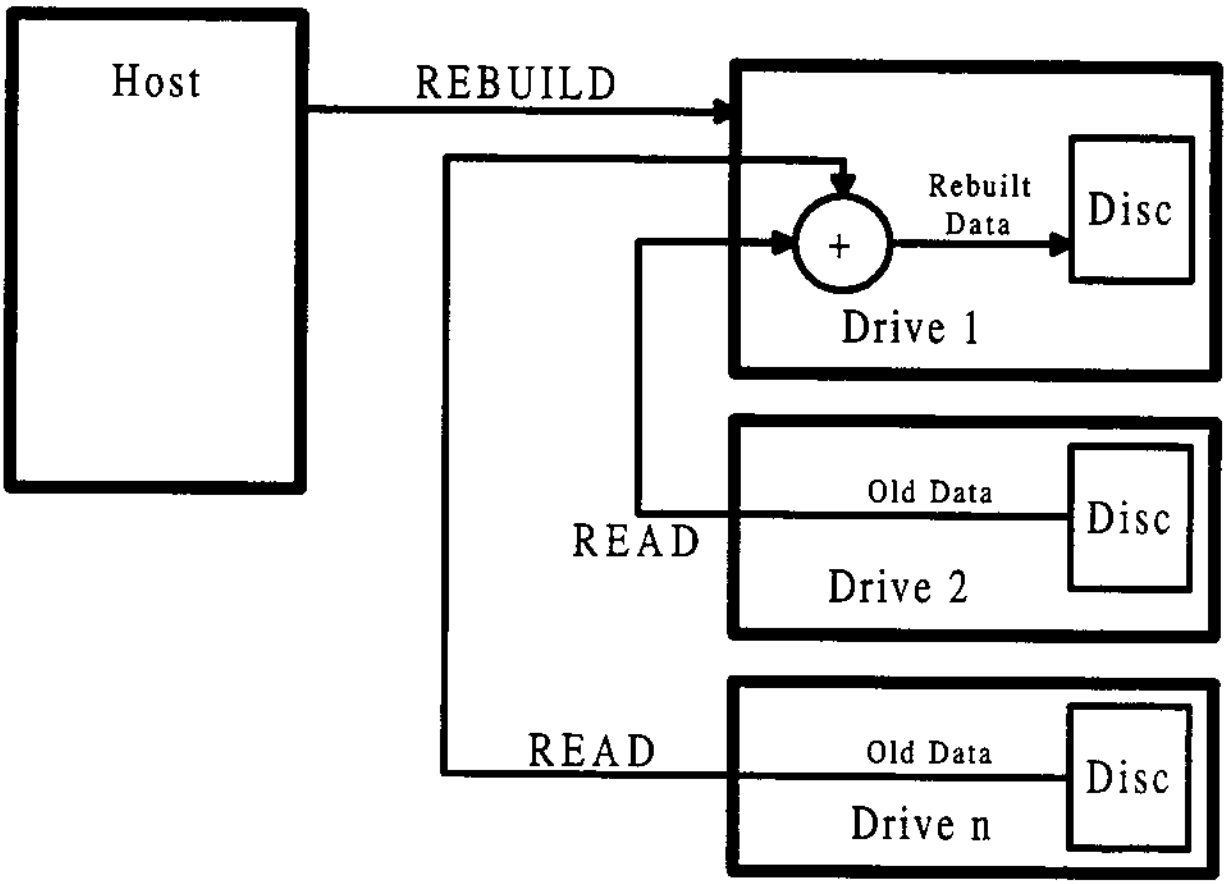


REBUILD Function

(Drives on <u>same</u> bus or loop)

The host begins by sending the REBUILD command to the drive to be rebuilt. The addresses of the source drives, as well as the logical block address and rebuild length are passed during the data out phase of the REBUILD command. The drive then takes on the role of initiator and issues READ commands to all source drives. The data from all source drives is exclusive or'd and written to the disc which is being rebuilt.

Figure 6: SINGLE INTERFACE REBUILD



New RAID Specific SCSI Commands

XDWRITE Command

(Exclusive Or Data Write)

The EXCLUSIVE-OR DATA WRITE command (Table 1) requests that the target read old data from the medium, write to the medium the data transferred by the initiator, exclusive-or the old data with the data transferred by the initiator, and send the xor'd data to the destination specified by the initiator. The destination shall be either 1) another target (secondary target), or 2) the target's buffer (so it can be retrieved with a subsequent command). The exclusive-or operation shall be bit for bit; byte 0, bit 0 of the transferred data shall be xor'd with byte 0, bit 0 of the old data, etc.

> IMPLEMENTOR'S NOTE: The XDWRITE command is well suited for the read-modify-write operation in a RAID 5 (Redundant Array of Inexpensive Discs, level 5) environment. The XDWRITE is typically used on the data drive, and allows the xor function to be handled in the device rather than in the controller.

If the Mirror bit is set to one, the target does not perform the exclusive-or operation. Instead, the data transferred by the initiator is sent directly to the destination specified by the initiator. If the Mirror bit is set to zero, the exclusive-or operation is performed, and the resulting data is sent to the destination specified by the initiator.

> IMPLEMENTOR'S NOTE: The Mirror bit is typically set to one in a RAID 1 environment, where all data written to the device is duplicated on a second "mirror" device.

If the NDisc bit is set to zero, the data transferred by the initiator shall be written to the medium. If the NDisc bit is set to one, the data shall not be written to the medium.

> IMPLEMENTOR'S NOTE: The NDisc bit is typically set to one during an initiator supervised regenerate or rebuild operation in the RAID configuration. The data transferred by the initiator is xor'd with data stored on the medium. The result is then retrieved by the initiator (without being written to the medium) to be similarly processed by successive targets in the RAID configuration.

If the Disable Page Out (DPO) bit is set to one, no data is cached. The DPO bit is invalid and shall be ignored if the RCD (Read Cache Disable) bit of Mode Select Page 8 is set true (Caching disabled).

A Force Unit Access (FUA) bit of one indicates that the write command shall not return GOOD status until the logical blocks transferred by the initiator have been written on the media. The FUA bit is invalid and shall be ignored if the WCE (Write Cache Enable) bit of Mode Sense page 8 is set false (Write caching disabled). The FUA bit shall be set to zero if the NDisc bit is set to one (data not written to medium). If both the FUA and NDisc bits are set to one the target shall

return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

If the Buffer bit is set to zero, then the xor'd data shall be sent to a secondary target whose address is specified in the Secondary Address field. Disconnection is required in this case. The target takes on the initiator role and sends an XPWRITE command to the secondary target. This secondary target operation is "nested" within the XDWRITE command; the ending status for the XDWRITE command is not sent to the initiator until the secondary operation has completed. If the XPWRITE command ends with CHECK CONDITION status and the sense bytes indicate an unrecoverable error, the XDWRITE command shall end with CHECK CONDITION status. Targets that are capable of accepting the XDWRITE command with the Buffer bit set to zero shall also be capable of sending the XPWRITE command (i.e. assume an initiator role).

> IMPLEMENTOR'S NOTE: The Buffer bit is typically set to zero if all devices in the RAID configuration are on the same bus or loop and can directly access one another. The secondary target is normally the parity device, to which the xor'd data is transferred.

If the Buffer bit is set to one, then the xor'd data shall be retained in the target's buffer until it is successfully transferred via a received associated XDREAD command. The xor'd data shall not be sent to a secondary target. The associated XDREAD command "satisfies" the XDWRITE command; the XDWRITE command remains "unsatisfied" until an associated XDREAD command is successfully completed. Good XDWRITE status implies, at a minimum, that the xor'd data is available in the buffer. Depending on the caching configuration of the target and the state of the NDisc bit, good XDWRITE status may also imply that data has been written to the medium (see NDisc and FUA bits above, and WCE bit of Mode Sense page 8). Targets that are capable of accepting the XDWRITE command with the Buffer bit set to one shall also be capable of accepting the XDREAD command.

> IMPLEMENTOR'S NOTE: The Buffer bit is typically set to one if the devices in the RAID configuration are on isolated busses or loops. In this case, the RAID controller (initiator) must supervise the read-modify-write operation, since the devices cannot directly access one another. The controller is normally the initiator of the subsequent associated XDREAD command.

If the Alternate Port bit is set to one, the secondary transfer (XPWRITE) shall take place on the alternate port from that which received the XDWRITE command. If the Alternate Port bit is set to zero the secondary transfer shall take place on the same port that received the XDWRITE command. The Alternate Port bit shall be set to zero if the Buffer bit is set to one. If both the Alternate Port and Buffer bits are set to one the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB. The Alternate Port bit shall be set to zero if the target is not a two port device. If the Alternate Port bit is set to one and the target is not a two port device the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

The logical block address field specifies the first logical block of the range of logical blocks to be operated on at the target.

The transfer length field specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred. The value contained in this field shall be used in the transfer length field of any associated XPWRITE command sent by the target.

The secondary address field specifies the address of the secondary target. This field is not valid and shall be ignored if the Buffer bit is set to one.

> IMPLEMENTOR'S NOTE: The secondary address may be the address of the initiator of the XDWRITE command. In this case, the initiator must be capable of becoming a target and accepting the nested XPWRITE command that will be sent by the XDWRITE target. This provides an alternate approach to using the XDREAD command in the case where the RAID devices are isolated from one another.

The secondary logical block address field specifies the value which shall be used in the logical block address field of any associated XPWRITE command sent by the target. This field is not valid and shall be ignored if the Buffer bit is set to one.

Note: there is no Logical Unit Number specified since it is always assumed to be zero for this command.

Table 1: XDWRITE COMMAND

BIT BYTE(S)	7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
1	Reserved	Mirror	NDisc	DPO	FUA	Buffer	Alt Port	Reserved
2 - 5	LOGICAL BLOCK ADDRESS							
6 - 7	TRANSFER LENGTH							
8 - 10	SECONDARY ADDRESS							
11 - 14	SECONDARY LOGICAL BLOCK ADDRESS							
15	CONTROL						Flag	Link

XPWRITE Command

(Exclusive Or Parity Write)

The EXCLUSIVE-OR PARITY WRITE command (Table 2) requests that the target read old data from the medium, and write to the medium the old data xor'd with the data transferred by the initiator. The exclusive-or operation shall be bit for bit; byte 0, bit 0 of the transferred data shall be xor'd with byte 0, bit 0 of the old data, etc.

> IMPLEMENTOR'S NOTE: The XPWRITE command is well suited for the read-modify-write operation in a RAID 5 (Redundant Array of Inexpensive Discs, level 5) environment. The XPWRITE is typically used on the parity drive, and allows the xor function to be handled in the device rather than in the controller. The data transferred by the initiator is typically the result of the previous associated XDWRITE command issued to a data drive.

If the Disable Page Out (DPO) bit is set to one, no data is cached. The DPO bit is invalid and shall be ignored if the RCD (Read Cache Disable) bit of Mode Select Page 8 is set true (Caching disabled).

A Force Unit Access (FUA) bit of one indicates that the write command shall not return GOOD status until the logical blocks transferred by the initiator have been written on the media. The FUA bit is invalid and shall be ignored if the WCE (Write Cache Enable) bit of Mode Sense page 8 is set false.

The logical block address field specifies the first logical block of the range of logical blocks to be operated on by this command.

The transfer length field specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

IMPLEMENTOR'S NOTE: The logical block address and transfer length fields typically contain values that were passed in the associated XDWRITE command.

Table 2: XPWRITE COMMAND

BIT BYTE(S)	7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
1	RESERVED			DPO	FUA	Reserved		
2 - 5	LOGICAL BLOCK ADDRESS							
6	RESERVED							
7 - 8	TRANSFER LENGTH							
9	CONTROL						Flag	Link

XDREAD Command

(Exclusive Or Data Read)

The EXCLUSIVE-OR DATA READ command (Table 3) requests that the target transfer to the initiator data stored in the targets buffer as a result of a previous associated XDWRITE command with the Buffer bit set to one, or a previous associated REGENERATE command. The successful completion of this command "satisfies" the associated command, relieving the target of having to retain the data in its buffer. Until the target receives this subsequent associated XDREAD command, an XDWRITE or REGENERATE command remains "unsatisfied" (see XDWRITE and REGENERATE commands).

The logical block address field shall contain the same value as was used in the associated command, and is used by the target to associate the two commands. If this field contains a value which does not match any unsatisfied command's logical block address, the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

The transfer length field specifies the number of contiguous logical blocks of data that shall be transferred, and shall contain the same value that was used in the transfer length field of the associated command. If this field contains a value which is different from the transfer length used in the associated command, the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

Table 3: XDREAD COMMAND

BIT BYTE(S)	7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
1	RESERVED							
2 - 5	LOGICAL BLOCK ADDRESS							
6	RESERVED							
7 - 8	TRANSFER LENGTH							
9	CONTROL						Flag	Link

REBUILD Command

(Rebuild)

The REBUILD command (Table 4) requests that the target write to the medium data which has been built by xor-ing data from the source(s) specified by the initiator (the target of the REBUILD command is not one of the sources, as it is with the REGENERATE command). The target assumes the role of initiator and reads the specified data from each source specified.

> IMPLEMENTOR'S NOTE: The REBUILD command is used in a RAID configuration to restore data which has been previously lost. Because of the redundancy nature of RAID, the lost data can be built by xor-ing data from the other devices in the configuration.

The exclusive-or operation shall be bit for bit; byte 0, bit 0 of the data from one specified source shall be xor'd with byte 0, bit 0 of the data from the next specified source, etc., until all specified data from the two sources has been xor'd. The data resulting from this operation shall be xor'd in a likewise manner with the next specified source. This shall continue until the data from all specified sources has been xor'd. If only one source is specified, no exclusive-or operation shall take place.

If the Alternate Port bit is set to one, the source data transfers shall take place on the alternate port from that which received the REBUILD command. If the Alternate Port bit is set to zero the source data transfers shall take place on the same port that received the REBUILD command. The Alternate Port bit shall be set to zero if the target is not a two port device. If the Alternate Port bit is set to one and the target is not a two port device the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

The parameter list length field specifies the length in bytes of the parameters that shall be sent during the DATA OUT phase of the command. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered an error, and no reconstruction or write operation shall take place.

The REBUILD parameter list is described in Table 5.

Table 4: REBUILD COMMAND

BIT BYTE(S)	7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
1	RESERVED						Alt Port	Reserved
2 - 4	PARAMETER LIST LENGTH							
5	CONTROL						Flag	Link

Table 5: REBUILD COMMAND PARAMETERS

BIT BYTE(S)	7	6	5	4	3	2	1	0
0 - 3	LOGICAL BLOCK ADDRESS							
4 - 7	REBUILD LENGTH							

SOURCE ADDRESS ENTRIES

0 - 3	SOURCE ADDRESS 0
4 - 7	SOURCE 0 LOGICAL BLOCK ADDRESS
	⋮
0 - 3	SOURCE ADDRESS n
4 - 7	SOURCE n LOGICAL BLOCK ADDRESS

The logical block address field specifies the logical block address at which the target is to begin data reconstruction.

The rebuild length field indicates the length in logical blocks of the rebuild operation. This corresponds to the length in logical blocks of the transfers from the specified sources. All F's in the rebuild length field indicates that the target shall rebuild through the last existing logical block on the medium.

The source address field indicates the address of the source of the data to be xor'd during the data reconstruction. This field is repeated for each source.

The source logical block address field indicates the logical block address at which to begin transferring data from the source. This field is repeated for each source.

REGENERATE Command

(Regenerate)

The REGENERATE command (Table 6) requests that the target write to its buffer data which has been built by xor-ing data from the source(s) specified by the initiator (the target of the REGENERATE command is inherently one of the sources). The target assumes the role of initiator and reads the specified data from each source specified. The result of the exclusive-or operation is placed in the buffer so it can be retrieved by a subsequent command.

> IMPLEMENTOR'S NOTE: The REGENERATE command is typically used for read operations in a RAID configuration when one data device has malfunctioned. Because of the redundancy nature of RAID, the data from the malfunctioning device can be regenerated by xor-ing data from the other devices in the configuration.

The exclusive-or operation shall be bit for bit; byte 0, bit 0 of the data from one specified source shall be xor'd with byte 0, bit 0 of the data from the next specified source, etc., until all specified data from the two sources has been xor'd. The data resulting from this operation shall be xor'd in a likewise manner with the next specified source (note: since xor operands are commutable, the source order is irrelevant). This shall continue until the data from all specified sources has been xor'd. The result is written to the target's buffer.

The xor'd data shall be retained in the target's buffer until it is successfully transferred via a received associated XDREAD command. The subsequent XDREAD command "satisfies" the REGENERATE command. The REGENERATE command remains "unsatisfied" until an associated XDREAD command is successfully completed. Good REGENERATE status is not returned until the xor'd data is available in the buffer. Targets that are capable of accepting the REGENERATE command shall also be capable of accepting the XDREAD command.

> IMPLEMENTOR'S NOTE: The XDREAD command is typically sent by the initiator that sent the REGENERATE command.

If the Alternate Port bit is set to one, the source data transfers shall take place on the alternate port from that which received the REGENERATE command. If the Alternate Port bit is set to zero the source data transfers shall take place on the same port that received the REGENERATE command. The Alternate Port bit shall be set to zero if the target is not a two port device. If the Alternate Port bit is set to one and the target is not a two port device the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

The parameter list length field specifies the length in bytes of the parameters that shall be sent during the DATA OUT phase of the command. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered an error.

The REGENERATE parameter list layout is shown in Table 7.

Table 6: REGENERATE COMMAND

BIT BYTE(S)	7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
1	RESERVED						Alt Port	Reserved
2 - 4	PARAMETER LIST LENGTH							
5	CONTROL							

Table 7: REGENERATE COMMAND PARAMETERS

BIT BYTE(S)	7	6	5	4	3	2	1	0
0 - 3	LOGICAL BLOCK ADDRESS							
4 - 7	REGENERATE LENGTH							

SOURCE DEVICE ADDRESSES

0 - 3	SOURCE ADDRESS 0
4 - 7	SOURCE 0 LOGICAL BLOCK ADDRESS
	⋮
0 - 3	SOURCE ADDRESS n
4 - 7	SOURCE n LOGICAL BLOCK ADDRESS

The logical block address field specifies the target's own logical block address at which to begin the regenerate operation.

The regenerate length field indicates the length in logical blocks of the regenerate operation. This corresponds to the length in logical blocks of the transfers from the specified sources.

The source address field indicates the address of the source of the data to be xor'd during the regenerate operation. This field is repeated for each source.

The source logical block address field indicates the logical block address at which to begin transferring data from the source. This field is repeated for each source.

Additional Performance Enhancements

Notice to Reader: This section of the document is still be worked on and does not yet fully describe how the Logging Function works.

The single biggest impact of a RAID 5 architecture is the performance degradation that occurs as a result of the Read-Modify-Write operation required for all writes. Typically the only way to improve on this situation has been to use a non-volatile write cache so that completion status can be returned to the OS before the Read-Modify-Write is complete on the disc drives. One way to improve write performance without a non-volatile write cache is to implement a special Logging mode in a spare disc drive. Many RAID 5 systems include a hot spare for system availability reasons. If this spare drive were used as a logging drive (when it isn't needed to replace a downed drive) system write performance could be significantly improved.

To implement this functionality the drive needs to be placed in a special mode called Logging Mode. This is done using a new mode page as described below. Once the drive is in logging mode the host writes data to the drive using standard write commands.

Log Control Page

The log control page (Table 8) provides the initiator the means to enable the target as a high speed "logging" device, which causes normal write and read commands to be executed as logged commands. In this mode write data is written to the next "available" location on the media. Once written, a location becomes "unavailable" until the successful completion of a subsequent LCLEAR command for that location.

The Log Mode bit, when set to one, enables the target as a logging device. When enabled as a logging device, the target shall be capable of accepting the LCLEAR command.

TABLE 8: LOG CONTROL PAGE

BIT BYTE(S)	7	6	5	4	3	2	1	0
0			PAGE CODE (XXh)					
1	PAGE LENGTH (02)							
2	RESERVED							LOG MODE
3	RESERVED							

IMPLEMENTOR'S NOTE: Configuring a target as a logging device causes a reduction in the average time for command complete status to be returned following a write operation. The logging device is normally used for temporary write data storage while the same data is being written to a more permanent location. Once the more permanent write is complete the LCLEAR command is normally issued to the logging device, causing the occupied location to once again become available for writes. Typically, data is not read from the logging device except for data recovery purposes - e.g. there was a failure during the write to the more permanent location.

The logging mode target will need to maintain a non-volatile record which links the logical block address of received commands to the physical location on the media, since there is otherwise no correlation between the two. To illustrate, if an initiator issues a write command specifying "n" as the logical block address, the next available block on the media becomes logical block "n", regardless of its physical location.

LCLEAR Command

(Log Clear)

The Log Clear command requests that the logging mode target (see log control mode sense page) make the specified blocks of data available for future write commands.

The logical block address field specifies the first logical block of the range of logical blocks to be operated on by this command.

The length field specifies the number of contiguous logical blocks of data that shall be cleared. A transfer length of zero indicates that no logical blocks shall be cleared. This condition shall not be considered an error.

In order to accept this command the target shall have been placed in logging mode via the log control page of the Mode Select command. If this command is received by a target which is not in logging mode, the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

IMPLEMENTOR'S NOTE: This command is typically sent when the data stored in the specified blocks has been written to a more permanent location, and the need to retain it on the logging device no longer exists.