Date: December 18, 2003

To: T10 Committee (SCSI)

From: Jim Coomes (Seagate)

Subject:SBC, SPC, Obsolete Third Party XOR Commands

1 Revisions

r2 - This revision includes deleteing references to the REGENERATE command in 5.2.39 and 5.2.40.

r1 - This revision includes removing references in SPC-3 to the third party XOR commands.

2 Overview

SBC-2 defines a series of XOR commands, REBUILD, REGENERATE and XDWRITE EXTENDED, that implement a third party mode of operation. In this mode, the primary target takes on the role as a temporary initiator to send commands to secondary targets to perform the operation. The following are some of the reasons why these commands have not been implemented:

    a)   It is difficult to determine the progress of the command execution when a failure occurs;
    b)   The interim XOR data is in volatile memory; and
    c)   The role of temporary initiator increases the complexity of the primary target.

This is a proposal to obsolete the third party XOR commands.

3 Changes to SPC-3 r16

4.5.2.3 Command-specific information sense data descriptor
Further meaning for the COMMAND-SPECIFIC INFORMATION field is defined within the command description in the appropriate command set standard (e.g., see SBC-2 for the MEDIUM SCAN, REASSIGN BLOCKS, ~~REBUILD, REGENERATE, and XPWRITE EXTENDED~~ commands, or see 6.3 for the EXTENDED COPY command).

7.5.3.1 Introduction to EXTENDED COPY protocol specific target descriptors
The target descriptors present in the parameter data of the EXTENDED COPY command (see 6.3) and ~~SBC-2 commands (e.g. REBUILD, REGENERATE, and XDWRITE EXTENDED)~~ may have protocol specific formats described in this subclause. An introduction to EXTENDED COPY target descriptors is provided in 6.3.6.1. ~~The descriptions of EXTENDED COPY target descriptors in this subclause assume usage by the EXTENDED COPY command and may not match the description in SBC-2.~~

4 Changes to SBC-2 r11

~~**3.1.38 third party:** When used in conjunction with exclusive or operations refers to the operations performed by a primary target with a secondary target on behalf of the host storage array controller.~~

4.2.1.9 Reservations

Table 3 — SBC-2 commands that are allowed in the presence of various reservations

Command	Addressed LU has this type of persistent reservation held by another initiator [B]				
	From any initiator		From registered initiator (RR all types)	From initiator not registered	
	Write Excl	Excl Access		Write Excl - RR	Exclusive Access - RR
REBUILD (16)/(32)	Conflict	Conflict	Allowed	Conflict	Conflict
REGENERATE (16)/(32)	Conflict	Conflict	Allowed	Conflict	Conflict
XDWRITE EXTENDED (16)/(32)/(64)	Conflict	Conflict	Allowed	Conflict	Conflict

4.2.3.1 Model for XOR commands overview

During third party XOR operations (see 4.2.3.3) only two data transfer operations are needed:

    a)   a write transfer from the storage array controller to the device containing protected data; and

    b)   a write transfer from the device containing protected data to the device containing check data.

4.2.3.3 Third party XOR operations

4.2.3.3.1 Third party XOR operations overview

Five XOR commands are needed to implement the third party XOR operations: XDWRITE EXTENDED, XPWRITE, XDREAD, REGENERATE, and REBUILD. The storage array controller also uses READ and WRITE commands for certain operations. Third party XOR operations are restricted to systems where all devices involved in the operations are located in the same SCSI domain since direct interaction between those devices is required.

4.2.3.3.2 Update write operation (third party)

The update write operation is used to write user data to a device containing protected data and updates the parity information on a different device containing check data.

    1)   An XDWRITE EXTENDED command is sent to the device containing protected data. This transfers the new write data to that device. The device reads its old user data, performs an XOR operation using the old user data and the received user data, temporarily stores the XOR result, and writes the received user data to the medium;

    2)   The device containing protected data becomes a temporary initiator and sends an XPWRITE command to the device containing check data. This transfers the resulting XOR data from the device containing protected data to the device containing check data. The device containing check data reads its check data, performs an XOR operation using the check data and the received XOR data, and writes the resulting XOR result to the medium; and

    3)   After the device containing protected data receives status for its XPWRITE command, it returns ending status for the XDWRITE EXTENDED command to the storage array controller. This indicates that the operations on both the device containing protected data and the device containing check data have completed.

4.2.3.3.3 Regenerate operation (third party)

The regenerate operation is used to recreate a logical block, including user data and protection information, if any, that is not readable from a block device. This is done by reading the associated logical block from each of the other devices within the redundancy group and performing an XOR operation with each of these logical blocks. The last XOR result is the regenerated logical block that had the error. The number of steps is

dependent on the number of devices in the redundancy group, but the sequence is as follows (since XOR operands are commutable the XOR order is irrelevant, and the order of steps 2 and 3 is interchangeable):

1) A REGENERATE command is sent to a valid device in the redundancy group (a valid device is any device in the group other than the failed device). This transfers the regenerate parameter list from the storage array controller to the device;
2) The device reads the requested data from its own medium. The device retains the requested data for a subsequent XOR operation;
3) The device becomes a temporary initiator and sends a READ command to another device included in the regenerate parameter list. That target device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the read data from the previous step and the read data received in this step. The resulting XOR data is retained for the next step;
4) Step 3 is repeated until all devices listed in the regenerate parameter list have been accessed. The last XOR data result is retained by the temporary initiator. The temporary initiator returns completion status for the REGENERATE command to the storage array controller when this regenerated user data is available; and
5) An XDREAD command is sent from the storage array controller to the device that had been a temporary initiator in the above steps. This transfers the regenerated user data from the last XOR data result from the device to the storage array controller.

4.2.3.3.4 Rebuild operation (third party)

The rebuild operation is similar to the regenerate operation, except that the last XOR result is written to the replacement device. This is used when a failed device is replaced and rebuilt data is written to that replacement device. The sequence is as follows:

1) A REBUILD command is sent to the replacement device in the redundancy group (the device that replaces a failed device). This transfers the rebuild parameter list from the storage array controller to the device;
2) The device becomes a temporary initiator and sends a READ command to a device included in the rebuild parameter list. That target device transfers the requested data to the temporary initiator. The temporary initiator retains this data for a subsequent XOR operation;
3) The temporary initiator sends a READ command to another device included in the rebuild parameter list. That device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the read data from the previous step and the read data received in this step. The resulting XOR data is retained for the next step; and
4) Step 3 is repeated until all devices listed in the rebuild parameter list have been accessed. The last XOR data result is written to the replacement device's medium, then the device returns completion status for the REBUILD command to the storage array controller.

4.2.3.4 Hybrid subsystem XOR operations

4.2.3.4.1 Hybrid subsystem XOR operations overview

In a hybrid subsystem the redundancy group is divided between two or more domains (see 4.2.3.2) and at least one of those domains contains two or more of the devices in the redundancy group. Such a system could do its XOR operations as described in 4.2.3.2 (Storage array controller supervised XOR operations) but it may choose to use third party XOR commands for parts of the XOR operation where all the involved devices are in the same domain. This subclause describes use of the third party XOR operations on a hybrid system with two domains. For illustration the redundancy group has six devices, with three devices in each domain.

4.2.3.4.2 Update write operation (hybrid subsystem)

When the update write operation involves two devices that are in different domains, the storage array controller uses the technique described in storage array controller supervised XOR operations. When the update write operation involves two devices that are in the same domain the storage array controller may use either the storage array controller supervised operation or the third party XOR operation.

~~4.2.3.4.3 Regenerate operation (hybrid subsystem)~~

~~The regenerate operation, for the illustrated case, always involves five XOR devices (all but the failed device) where three devices are in one domain (referred to as domain A) and two devices are in the other domain (referred to as domain B). The sequence is as follows (since XOR operands are commutable the XOR order is irrelevant, and the order of steps 2 and 3 is interchangeable; likewise, the order of steps 7 and 8 is interchangeable):~~

1) ~~A REGENERATE command is sent to a device in domain A. This transfers the regenerate parameter list (containing the other two valid devices and data extents) from the storage array controller to the device;~~
2) ~~The device reads the requested data from its own medium. The device retains the requested data for a future XOR operation;~~
3) ~~The device becomes a temporary initiator and sends a READ command to another device included in the regenerate parameter list. That device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the read data from the previous step and the read data received in this step. The resulting XOR data is retained for the next step;~~
4) ~~Step 3 is repeated until all devices listed in the regenerate parameter list have been accessed. The last XOR data result is retained by the temporary initiator. The temporary initiator returns completion status for the REGENERATE command to the storage array controller when this partially regenerated user data is available;~~
5) ~~An XDREAD command is sent from the storage array controller to the temporary initiator device. This transfers the partially regenerated user data from the last XOR data result from the device to the storage array controller;~~
6) ~~A REGENERATE command with the intermediate data bit set is sent to a device in domain B. This transfers the regenerate parameter list (containing the other valid device and data extent) from the storage array controller to the device. The partially regenerated user data received in step 5 is sent as the intermediate data;~~
7) ~~The device reads the requested data from its own medium. The device performs an XOR operation between the intermediate data and its own data. The resulting XOR data is retained for the next step;~~
8) ~~The device becomes a temporary initiator and sends a READ command to the other device included in the regenerate parameter list. That target device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the XOR data from step 7 and the read data received in this step. The last XOR data result is retained by the temporary initiator. The temporary initiator returns completion status for the REGENERATE command to the storage array controller when this regenerated user data is available; and~~
9) ~~An XDREAD command is sent from the storage array controller to the device that had been a temporary initiator device in the above steps. This transfers the regenerated user data from the last XOR data result from the device to the storage array controller.~~

~~4.2.3.4.4 Rebuild operation (hybrid subsystem)~~

~~The rebuild operation, for this illustration, involves five valid devices and the replacement device where three valid devices are in one domain (referred to as domain A) and two valid devices and the replacement device are in the other domain (referred to as domain B). The sequence is as follows (since XOR operands are commutable the XOR order is irrelevant, and the order of steps 2 and 3 is interchangeable):~~

1) ~~A REGENERATE command is sent to a device in domain A. This transfers the regenerate parameter list (containing the other two valid devices and data extents) from the storage array controller to the device;~~
2) ~~The device reads the requested data from its own medium. The device retains this data for a future XOR operation;~~
3) ~~The device becomes a temporary initiator and sends a READ command to another device included in the regenerate parameter list. That device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the read data from the previous step and the read data received in this step. The resulting XOR data is retained for the next step;~~
4) ~~Step 3 is repeated until all devices listed in the regenerate parameter list have been accessed. The last XOR data result is retained by the temporary initiator. The temporary initiator returns completion status for the REGENERATE command to the storage array controller when the data is available;~~

5) An XDREAD command is sent from the storage array controller to the temporary initiator device. This transfers the partially regenerated data from the device to the storage array controller;

6) A REBUILD command with the intermediate data bit set is sent to the replacement device in domain B. This transfers the rebuild parameter list (containing the two valid devices and data extents) from the storage array controller to the device. The partially rebuilt data received in step 5 is sent as the intermediate data;

7) The device becomes a temporary initiator and sends a READ command to a device included in the rebuild parameter list. That target device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the intermediate data and the read data received in this step. The resulting XOR data is retained for the next step;

8) The temporary initiator sends a READ command to the other device included in the rebuild parameter list. That device transfers the requested data to the temporary initiator. The temporary initiator performs an XOR operation between the read data from the previous step and the read data received in this step; and

9) The last XOR data result is written to the replacement device's medium, then the device returns completion status for the REBUILD command to the storage array controller.

4.2.3.5.3 Access to an inconsistent stripe

<b>)When an XDWRITE EXTENDED command has been issued, the device containing protected data and the device containing check data are updated at different times during the course of the command. The stripe should be treated as inconsistent between the time the storage array controller issues the XDWRITE EXTENDED command and the completion status for the command is received; and

4.2.3.6.2 Errors during third party XOR operations

Third party operations involve the processing of several commands exchanged among three or more devices. For the purposes of this clause, a command that causes the recipient of that command to generate one or more other commands to another device (or devices) is referred to as a primary command and the recipient of a primary command is referred to as a primary target. All commands generated by a primary target (based on the receipt of a primary command) are referred to as secondary commands and are sent to a secondary target or secondary targets. The definitions of primary command, primary target, secondary command, and secondary target are temporary, for this clause only, and should not be associated with the more general "SCSI Primary Command" or "SCSI Target" usage.

The primary target of an XDWRITE EXTENDED primary command generates an XPWRITE secondary command; the primary target of a REBUILD primary command generates one or more READ secondary commands. The primary command shall not be completed until the primary target is prepared to report the completion status or implied completion status of the secondary commands. Two classes of exception conditions may occur during these commands. One class consists of those resulting directly from the primary command. The other class consists of those resulting from a secondary command. Either or both of these classes of exception may occur during a third party operation.

4.2.3.6.3 Primary errors - errors resulting directly from the primary command

The first class of errors consists of exception conditions that are detected by the device that received the primary command (primary target) and are not due to the failure of a resulting secondary command. These conditions include, but are not limited to, invalid parameters in the primary command, inability of the primary target to continue operating, and parity errors while transferring the primary command, data, or status byte. In the event of such an exception condition, the primary target shall:

1) terminate the primary command with CHECK CONDITION status; and
2) build sense data according to the exception condition.

4.2.3.6.4 Secondary errors - errors resulting from the secondary command

The second class of errors consists of exception conditions resulting from the failure of a secondary command. The sense data for such errors shall be passed to the initiator of the primary command in the additional sense code field of the sense data.

If the primary target detects the exception (i.e., by some means other than receiving CHECK CONDITION status from the secondary target) it shall:

1) terminate the primary command with CHECK CONDITION status;
2) set the sense key to ABORTED COMMAND if there are no primary errors to report. Otherwise, the sense key shall be set according to the primary error;
3) set the first byte of the COMMAND SPECIFIC INFORMATION field of the sense data to the starting byte number, relative to the first byte of sense data, of an area that contains the primary target's sense data for the secondary error. A zero value in this byte indicates no secondary error has been detected by the primary target. The secondary sense data shall be built in the standard sense data format (see SPC-3); and
4) in the case of a REBUILD or REGENERATE primary command, set the third byte of the COMMAND SPECIFIC INFORMATION field of the sense data to an index value indicating the target identifier of the failing secondary target. This value shall be an index into the source descriptor entries of the parameter data of the primary command, and shall point to the entry containing the target identifier of the failing device (i.e., 00h points to the first entry, 01h points to the second entry, etc.). This byte shall be ignored if the primary command is not a REBUILD or REGENERATE.

If the secondary target detects the exception, the primary target receives CHECK CONDITION status from the secondary target. The primary target shall recover the sense data associated with the exception condition, clear any exception conditions associated with the CHECK CONDITION status, and shall:

1) terminate the primary command with CHECK CONDITION status;
2) set the sense key to ABORTED COMMAND if there are no primary errors to report. Otherwise, the sense key shall be set according to the primary error;
3) set the second byte of the COMMAND SPECIFIC INFORMATION field of the sense data to the starting byte number, relative to the first byte of sense data, of an area that contains (unchanged) the secondary target's status byte followed by its sense data. A zero value in this byte indicates no secondary error has been reported by the secondary target; and
4) in the case of a REBUILD or REGENERATE (primary) command, set the third byte of the COMMAND SPECIFIC INFORMATION field of the sense data to an index value indicating the target identifier of the failing secondary target. This value shall be an index into the source descriptor entries of the parameter data of the primary command, and shall point to the entry containing the target identifier of the failing device; 00h points to the first entry, 01h points to the second entry, etc. This byte is invalid and shall be ignored if the primary command is not a REBUILD or REGENERATE.

For a given primary command, if errors are generated by more than one secondary command, the sense data shall contain error information for the secondary error first obtained by the primary target.

Since, for secondary errors, the sense key is set to ABORTED COMMAND only if there are no primary errors to report (see item 2 above), the first and second bytes of the COMMAND SPECIFIC INFORMATION field should be checked, even when the sense key is a value other than ABORTED COMMAND, to determine if any secondary errors have occurred.

NOTE 1 - All three of the above error types might occur during the same third party operation. If this happens, there are three unique pieces of error information contained in the sense data: one for the primary error (starting at byte 0), and two for the secondary errors (in the additional sense code).

4.5.5 Protected data commands

<e>)REBUILD (16)/(32);
<f>)REGENERATE (16)/(32);
<o>)XDWRITE EXTENDED (16)/(32)/(64);

5.1 Opcodes for variable length CDB opcodes

Table 12 — Variable length command service action code assignments

Service action code	Description	Reference
0001h	REBUILD (32)	5.2.19
0002h	REGENERATE (32)	5.2.21
0005h	XDWRITE EXTENDED (32)	5.2.44
0008h	XDWRITE EXTENDED (64)	5.2.47

5.2.1 Commands for direct-access devices overview.

Table 15 — Commands for direct-access devices

Command name	Operation code	Type	Reference
REBUILD (16)	81h	O	5.2.18
REBUILD (32)	7Fh/0001h	O	5.2.19
REGENERATE (16)	82h	O	5.2.20
REGENERATE (32)	7Fh/0002h	O	5.2.21
XDWRITE EXTENDED (16)	80h	O	5.2.45
XDWRITE EXTENDED (32)	7Fh/0005h	O	5.2.46
XDWRITE EXTENDED (64)	7Fh/0008h	O	5.2.47

5.2.18 REBUILD (16) Command

Editor's Note 1: The REBUILD commands might be obsoleted

The REBUILD (16) command (see table 52) requests that the target write to the medium the XOR data generated from the specified source devices. The target, acting as a temporary initiator, issues READ commands to retrieve the specified logical blocks. Logical blocks include user data and protection information,

if any. READ (10) should be used for accesses to SCSI devices supporting less than 2 Terabytes, and READ (16) should be used for accesses to SCSI devices supporting greater than or equal to 2 Terabytes.

Table 52 — REBUILD (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (81h)							
1	Reserved			DPO	FUA	INTDATA	PORT CONTROL	
2	(MSB)							
5	LOGICAL BLOCK ADDRESS							(LSB)
6	(MSB)							
9	REBUILD LENGTH							(LSB)
10	(MSB)							
13	PARAMETER LIST LENGTH							(LSB)
14	Reserved							
15	CONTROL							

See 4.2.1.9 for reservation requirements for this command. See the READ (10) command (see 5.2.8) for a definition of the DPO and FUA bits.

Editor's Note 2: this command needs a description of when the temporary initiator sets the RDPROTECT field to non zero in its READ commands (probably if its own medium is formatted with protection information). Which value it chooses needs to be controlled (probably with a RDPROTECT field in this CDB). Rules are needed controlling if the temporary initiator checks the protection information it receives.

If the intermediate data (INTDATA) bit is zero, then intermediate data is not sent with the rebuild parameter list (see table 54). If the bit is one, the rebuild parameter list includes intermediate data. The length of the intermediate data may be calculated by multiplying the REBUILD LENGTH times the block length. This data shall be treated as an additional source, and an XOR operation performed with it and the data from the specified sources.

The PORT CONTROL field is defined in table 53. If the PORT CONTROL field has a value of 01b and the target is not a multiple port device the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

Table 53 — PORT CONTROL field

Value	Description
00b	The target transfers the data using the same port that received the command.
01b	The target transfers the data using a different port than the one that received the command.
10b	The target transfers the data using one port of the target's choice.
11b	The target transfers the data using one or more ports of the target's choice.

NOTE 2   The target that receives the REBUILD command is not one of the source devices. If only one source is specified, then an XOR operation does not occur. This case may occur in disk mirroring applications.

If the command terminates with CHECK CONDITION status the INFORMATION field in the sense data shall contain the LBA of the failed logical block with the lowest LBA. All logical blocks affected by the command and having an LBA lower than that of the reported failing logical block shall have been rebuilt and written to the medium.

Editor's Note 3: is it the INFORMATION field or the COMMAND SPECIFIC INFORMATION field above?

The LOGICAL BLOCK ADDRESS field specifies the starting LBA where the target shall write the XOR result data on its own medium. The REBUILD LENGTH field specifies the number of logical blocks to be written to the medium. It also specifies the number of logical blocks that are read from each source.

The PARAMETER LIST LENGTH field specifies the length in bytes of the parameter list that shall be transferred from application client. The REBUILD (16) parameter data is described in table 54.

Table 54 — REBUILD (16) and REGENERATE (16) parameter data

Byte\Bit	7	6	5	4	3	2	1	0
0	NUMBER OF SOURCE DESCRIPTORS (x)							
1	Reserved							
2	(MSB)	SOURCE DESCRIPTOR/PAD LENGTH						
3								(LSB)
Source descriptor(s) (if any)								
4	Source descriptor 0							
19								
...	...							
16x − 12	Source descriptor x							
16x + 3								
Pad (if any)								
16x + 4	PAD (length y)							
16x+y+3								
Intermediate data (if any)								
16x+y+4	INTERMEDIATE DATA (length z)							
16x+y+z+3								

The NUMBER OF SOURCE DESCRIPTORS field specifies the number of source descriptors in the parameter data.

The SOURCE DESCRIPTOR/PAD LENGTH specifies the sum of the lengths in bytes of all of the source descriptors and the PAD.

The SOURCE DESCRIPTORs identify the source device target identifiers and starting LBAs on the devices for the regenerate or rebuild operation. See table 55 for the source descriptor format.

Table 55 — REBUILD (16) and REGENERATE (16) source descriptor format

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7				SOURCE DEVICE ADDRESS				(LSB)
8								
11				Reserved				
12	(MSB)							
15				SOURCE STARTING LOGICAL BLOCK ADDRESS				(LSB)

The SOURCE DEVICE ADDRESS field specifies a SAM-3 compliant target identifier of the device that is the data source. The target identifier is limited to 64 bits in this command; REBUILD (32) supports longer target identifiers. The implied LUN is zero.

The SOURCE STARTING LOGICAL BLOCK ADDRESS field indicates the starting LBA to use when reading data from the source specified in the SOURCE DEVICE ADDRESS field.

The PAD field accommodates initiators that require the INTERMEDIATE DATA to be aligned on a particular memory boundary. The PAD field shall be ignored.

The INTERMEDIATE DATA field contains data that shall be used in the XOR operation with the data from the specified source devices. The length of the data is equal to the rebuild/regenerate length multiplied by the block length.

5.2.19 REBUILD (32) Command

Editor's Note 4: The REBUILD commands might be obsoleted

The REBUILD (32) command (see table 56) requests that the target write to the medium the XOR data generated from the specified source devices. The target, acting as a temporary initiator, issues READ

commands to retrieve the specified logical blocks. Logical blocks include user data and protection information, if any.

Table 56 — REBUILD (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH							
8	(MSB)	SERVICE ACTION (0001h)						
9								(LSB)
10	Reserved			DPO	FUA	INTDATA	PORT CONTROL	
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	Reserved							
23								
24	(MSB)	REBUILD LENGTH						
27								(LSB)
28	(MSB)	PARAMETER LIST LENGTH						
31								(LSB)

See 4.2.1.9 for reservation requirements for this command. See the REBUILD (16) command (see 5.2.18) table 57, table 58, and SPC-3 for a description of the fields in this command.

Editor's Note 5: this command needs a description of when the temporary initiator sets the RDPROTECT field to non-zero in its READ commands (probably if its own medium is formatted with protection information). Which value it chooses needs to be controlled (probably with a RDPROTECT field in this CDB). Rules are needed controlling if the temporary initiator checks the protection information it receives.

The REBUILD (32) parameter data is described in table 57.

Table 57 — REBUILD (32) and REGENERATE (32) parameter data

Byte\Bit	7	6	5	4	3	2	1	0
0	NUMBER OF SOURCE DESCRIPTORS (x)							
1	Reserved							
2	(MSB)		SOURCE DESCRIPTOR/PAD LENGTH					
3								(LSB)
Source descriptor(s) (if any)								
4	Source descriptor 0							
43								
...	...							
40x - 36	Source descriptor x							
40x + 3								
Pad (if any)								
40x + 4	PAD (length y)							
40x+y+3								
Intermediate data (if any)								
40x+y+4	INTERMEDIATE DATA (length z)							
40x+y+z+3								

The source descriptor format is specified in table 58. All other fields in the parameter data are as defined in 5.2.18.

Table 58 — REBUILD (32) and REGENERATE (32) source descriptor format

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)		SOURCE DEVICE ADDRESS					
31								(LSB)
32	(MSB)		SOURCE STARTING LOGICAL BLOCK ADDRESS					
39								(LSB)

The SOURCE DEVICE ADDRESS specifies the third party logical unit to use as the data source. The format of this conforms to one of the target descriptor formats of the EXTENDED COPY command specified in SPC-3.

5.2.20 REGENERATE (16) command

Editor's Note 6: The REGENERATE commands might be obsoleted

The REGENERATE (16) command (see table 59) requests that the target write to the buffer the XOR data generated from its own medium and the specified source devices. The target, acting as a temporary initiator, issues READ commands to retrieve the specified logical blocks. Logical blocks include user data and

protection information, if any. The resulting XOR data is retained in the target's buffer until it is retrieved by an XDREAD command with an LOGICAL BLOCK ADDRESS field and TRANSFER LENGTH field that match, or are a subset of, the LOGICAL BLOCK ADDRESS field and REGENERATE LENGTH field specified by this command.

Table 59 — REGENERATE (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (82h)							
1	Reserved			DPO	FUA	INTDATA	PORT CONTROL	
2	(MSB)							
5	LOGICAL BLOCK ADDRESS							(LSB)
6	(MSB)							
9	REGENERATE LENGTH							(LSB)
10	(MSB)							
13	PARAMETER LIST LENGTH							(LSB)
14	Reserved							
15	CONTROL							

See 4.2.1.9 for reservation requirements for this command. See 5.2.8 for a definition of the DPO and FUA bits and 5.2.18 for a definition of the INTDATA and PORT CONTROL fields.

> Editor's Note 7: this command needs a description of when the temporary initiator sets the RDPROTECT field to non-zero in its READ commands (probably if its own medium is formatted with protection information). Which value it chooses needs to be controlled (probably with a RDPROTECT field in this CDB). Rules are needed controlling if the temporary initiator checks the protection information it receives.

The LOGICAL BLOCK ADDRESS field specifies the starting LBA for the target to read data from its own medium. This data is a source for the regenerate operation.

The REGENERATE LENGTH field indicates the length in logical blocks of the resulting XOR data. It also specifies the length in logical blocks that is transferred from each of the specified sources.

The parameter data for the REGENERATE command is defined in table 57. The parameter data describes the other devices that are sources for the regenerate operation. The target receiving the REGENERATE command is implicitly a source, and is not included in the parameter data.

5.2.21 REGENERATE (32) command

> Editor's Note 8: The REGENERATE commands might be obsoleted

The REGENERATE (32) command (see table 60) requests that the target write to the buffer the XOR data generated from its own medium and the specified source devices. The target, acting as a temporary initiator, issues READ commands to retrieve the specified data.

Table 60 — REGENERATE (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2 6	Reserved							
7	ADDITIONAL CDB LENGTH (18h)							
8 9	(MSB) SERVICE ACTION (0002h)							(LSB)
10	Reserved			DPO	FUA	IDATA	PORT CONTROL	
11	Reserved							
12 19	(MSB) LOGICAL BLOCK ADDRESS							(LSB)
20 23	Reserved							
24 27	(MSB) REGENERATE LENGTH							(LSB)
28 31	(MSB) PARAMETER LIST LENGTH							(LSB)

See 4.2.1.9 for reservation requirements for this command. See the REGENERATE (16) command (see 5.2.20), table 57, table 58, and SPC-3 for a description of the fields in this command.

> Editor's Note 9: this command needs a description of when the temporary initiator sets the RDPROTECT field to non-zero in its READ commands (probably if its own medium is formatted with protection information). Which value it chooses needs to be controlled (probably with a RDPROTECT field in this CDB). Rules are needed controlling if the temporary initiator checks the protection information it receives.

5.2.39 XDREAD (10) command

The XDREAD (10) command (see table 84) requests that the target transfer to the initiator the XOR data generated by an XDWRITE or REGENERATE command. Data transferred from the device server includes user data and includes protection information as required by the XORPINFO bit and the medium format.

The XOR data transferred is identified by the LOGICAL BLOCK ADDRESS field and the TRANSFER LENGTH field. The LOGICAL BLOCK ADDRESS field and TRANSFER LENGTH field shall be the same as, or a subset of, those specified in a prior XDWRITE or REGENERATE command. If a match is not found the command is terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.1.4.2).

5.2.40 XDREAD (32) command

The XDREAD (32) command (see table 85) requests that the target transfer to the initiator the XOR data generated by an XDWRITE ~~or REGENERATE~~ command. Data transferred from the device server includes user data and includes protection information as required by the XORPINFO bit and the medium format

~~5.2.45 XDWRITE EXTENDED (16) command~~

~~Editor's Note 10: The XDWRITE EXTENDED commands might be obsoleted~~

~~The XDWRITE EXTENDED (16) command (see table 90) requests that the target XOR the data transferred from the application client with the data on the medium. Data transferred from the application client includes user data and includes protection information as required by the WRPROTECT bit and the medium format. The resulting XOR data may subsequently be sent to a secondary device using an XPWRITE (10) or XPWRITE (32) command. The target, acting as a temporary initiator, issues XPWRITE commands to retrieve the specified data. XPWRITE (16) should be used for access to SCSI devices supporting less than 2 Terabytes, and XPWRITE (32) should be used for accesses to SCSI devices supporting greater than or equal to 2 Terabytes.~~

~~**Table 90 — XDWRITE EXTENDED (16) command**~~

~~Byte\Bit~~	~~7~~	~~6~~	~~5~~	~~4~~	~~3~~	~~2~~	~~1~~	~~0~~
~~0~~	~~OPERATION CODE (80h)~~							
~~1~~	~~WRPROTECT~~			~~DPO~~	~~FUA~~	~~DISABLE WRITE~~	~~PORT CONTROL~~	
~~2~~	~~(MSB)~~							
~~5~~		~~LOGICAL BLOCK ADDRESS~~						~~(LSB)~~
~~6~~	~~(MSB)~~							
~~9~~		~~SECONDARY LOGICAL BLOCK ADDRESS~~						~~(LSB)~~
~~10~~	~~(MSB)~~							
~~13~~		~~TRANSFER LENGTH~~						~~(LSB)~~
~~14~~	~~SECONDARY ADDRESS~~							
~~15~~	~~CONTROL~~							

~~See 4.2.1.9 for reservation requirements for this command. See 5.2.8 for a definition of the DPO and FUA bits. See the WRITE (10) command (see 5.2.8) for a definition of the WRPROTECT field.~~

~~Editor's Note 11: How is XORPINFO chosen for the XPWRITE commands the temporary initiator issues?~~

~~The SECONDARY ADDRESS field contains the target identifier of the target that will receive the XOR data transfer. The implied LUN of the secondary target shall be zero. If the transport protocol requires more than one byte for the target identifier, the SECONDARY ADDRESS field specifies the least significant byte of the secondary target identifier, and the upper bytes of the secondary target identifier shall be equal to the upper bytes of the target identifier of the XDWRITE EXTENDED target.~~

A DISABLE WRITE bit of zero indicates that the data transferred from the application client shall be written to the medium after the XOR operation is complete. A DISABLE WRITE bit of one indicates that the data shall not be written to the medium.

See 5.2.18 for a definition of the PORT CONTROL field. If the PORT CONTROL field has a value of 01b and the target is not a multiple port device the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

The TRANSFER LENGTH field specifies the number of logical blocks that shall be transferred from the application client, and to the secondary device. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page (see 6.1.4.2).

The XOR data transfer to the secondary target is performed using an XPWRITE command. The XPWRITE command shall be sent to the device specified in the SECONDARY ADDRESS field. The SECONDARY LOGICAL BLOCK ADDRESS field value shall be placed in the LOGICAL BLOCK ADDRESS field of the XPWRITE command. The TRANSFER LENGTH field value shall be placed in the TRANSFER LENGTH field of the XPWRITE command. The completion status of the XDWRITE EXTENDED command shall not be returned to the initiator until the completion status of the XPWRITE command has been received.

> NOTE 3 - The XOR data transfer to the secondary target may be broken into multiple XPWRITE commands. If this is done, the XDWRITE EXTENDED target calculates the LBAs and transfer lengths for the individual XPWRITE commands. Also, the completion status of the XDWRITE EXTENDED command is not returned to the initiator until the completion status of all XPWRITE commands have been received.

If the prior XPWRITE command terminates with a CHECK CONDITION status and the sense key is not set to RECOVERED ERROR the XDWRITE EXTENDED command shall return CHECK CONDITION status.

5.2.46 XDWRITE EXTENDED (32) command

> Editor's Note 12: The XDWRITE EXTENDED commands might be obsoleted

The XDWRITE EXTENDED (32) command (see table 91) requests that the target XOR the data transferred from the application client with the data on the medium. Data transferred from the application client includes

user data and includes protection information as required by the WRPROTECT bit and the medium format. The resulting XOR data may subsequently be sent to a secondary device using an XPWRITE (32) command.

Table 91 — XDWRITE EXTENDED (32) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)			SERVICE ACTION (0005h)				
9								(LSB)
10	WRPROTECT			DPO	FUA	DISABLE WRITE	PORT CONTROL	
11	SECONDARY ADDRESS							
12	(MSB)			LOGICAL BLOCK ADDRESS				
19								(LSB)
20	(MSB)			SECONDARY LOGICAL BLOCK ADDRESS				
27								(LSB)
28	(MSB)			TRANSFER LENGTH				
31								(LSB)

See 4.2.1.9 for reservation requirements for this command. See the XDWRITE EXTENDED (16) command (see 5.2.45) and SPC 3 for a description of the fields in this command.

Editor's Note 13: How is XORPINFO chosen for the XPWRITE commands the temporary initiator issues?

5.2.47 XDWRITE EXTENDED (64) command

The XDWRITE EXTENDED (64) command (see table 92) requests that the target XOR the data transferred from the application client with the data on the medium. Data transferred from the application client includes

user data and includes protection information as required by the WRPROTECT bit and the medium format. The resulting XOR data may subsequently be sent to a secondary device using an XPWRITE (32) command.

Table 92 — XDWRITE EXTENDED (64) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
6								
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB) SERVICE ACTION (0005h)							
9								(LSB)
10	WRPROTECT			DPO	FUA	DISABLE WRITE	PORT CONTROL	
11	Reserved							
12	SECONDARY ADDRESS DESCRIPTOR							
43								
44	(MSB) LOGICAL BLOCK ADDRESS							
51								(LSB)
52	(MSB) SECONDARY LOGICAL BLOCK ADDRESS							
59								(LSB)
60	(MSB) TRANSFER LENGTH							
63								(LSB)

See 4.2.1.9 for reservation requirements for this command.

> Editor's Note 14: How is XORPINFO chosen for the XPWRITE commands the temporary initiator issues?

The SECONDARY ADDRESS DESCRIPTOR field contains the logical unit identifier of the logical unit that will receive the XOR data transfer. The format of this field conforms to one of the target descriptor formats of the EXTENDED COPY command as specified in SPC-3.

See the XDWRITE EXTENDED (16) command (see 5.2.45) and SPC-3 for a description of the other fields in this command.

6.1.3.10 XOR Control mode page

Table 134 — XOR Control mode page

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	Reserved	PAGE CODE (10h)					
1	PAGE LENGTH (16h)							
2	Reserved						XORDIS	Reserved
3	Reserved							
4	(MSB)		MAXIMUM XOR WRITE SIZE					
7								(LSB)
8	Reserved							
11								
12	(MSB)		Obsolete MAXIMUM REGENERATE SIZE					
15								(LSB)
16	(MSB)		Obsolete MAXIMUM REBUILD TRANSFER SIZE					
19								(LSB)
20	Reserved							
21								
22	(MSB)		Obsolete REBUILD DELAY					
23								(LSB)

The MAXIMUM XOR WRITE SIZE field specifies the maximum transfer length in blocks that the target accepts for a single XDWRITE, XDWRITE EXTENDED, or XPWRITE command.

The MAXIMUM REGENERATE SIZE field specifies the maximum REGENERATE LENGTH in blocks that the target accepts for the REGENERATE command.

The MAXIMUM REBUILD TRANSFER SIZE field specifies the maximum transfer length in blocks that the target shall use for commands issued as a temporary initiator (e.g., READ and XPWRITE commands during a rebuild operation). This field does not limit the rebuild size.

The REBUILD DELAY field is provided to allow allocation of the SCSI interconnect subsystem bandwidth. The REBUILD DELAY field specifies the minimum time in milliseconds between successive READ commands during a rebuild operation.
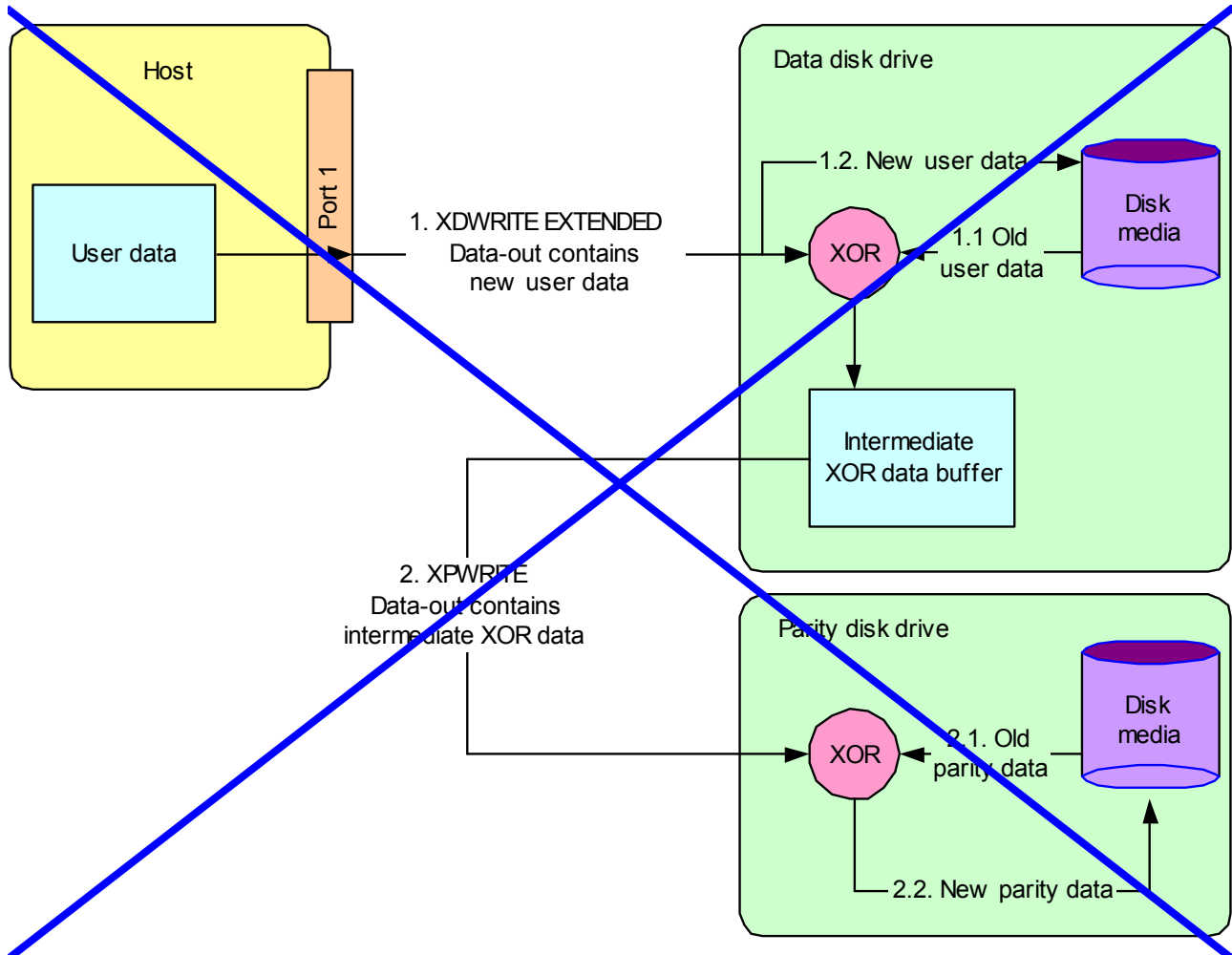
0.1 Third-party XOR operations

0.1.1 Update write operation

Figure 0.1 illustrates a third party read-modify-write operation. The example uses a host, a data disk device (holding protected user data), and a parity disk device (holding check data). In this example, the data and parity devices are on the same SCSI physical interconnect, and thus are capable of peer-to-peer interaction. Two SCSI commands are used: XDWRITE EXTENDED and XPWRITE.

The host begins by sending user data to the data disk device using an XDWRITE EXTENDED command. The data disk device assumes initiator role and sends an XPWRITE command to the parity disk device (the data disk device does not yet have the intermediate XOR data for this command; the purpose of issuing the XPWRITE command at this time is to cause the parity disk device to begin reading XOR data from its medium to its buffer memory).

The data disk device reads old user data from its medium, performs an XOR operation using the old user data and the user data from the host, stores the resulting intermediate XOR data in its buffer memory, and writes the user data from the host to its medium.

The data disk device makes the resulting intermediate XOR data (from its buffer memory) available to the parity disk device for the already issued XPWRITE command. The parity disk device performs an XOR operation using the intermediate XOR data and the XOR data in its buffer memory. The resulting new XOR data is written to the medium.
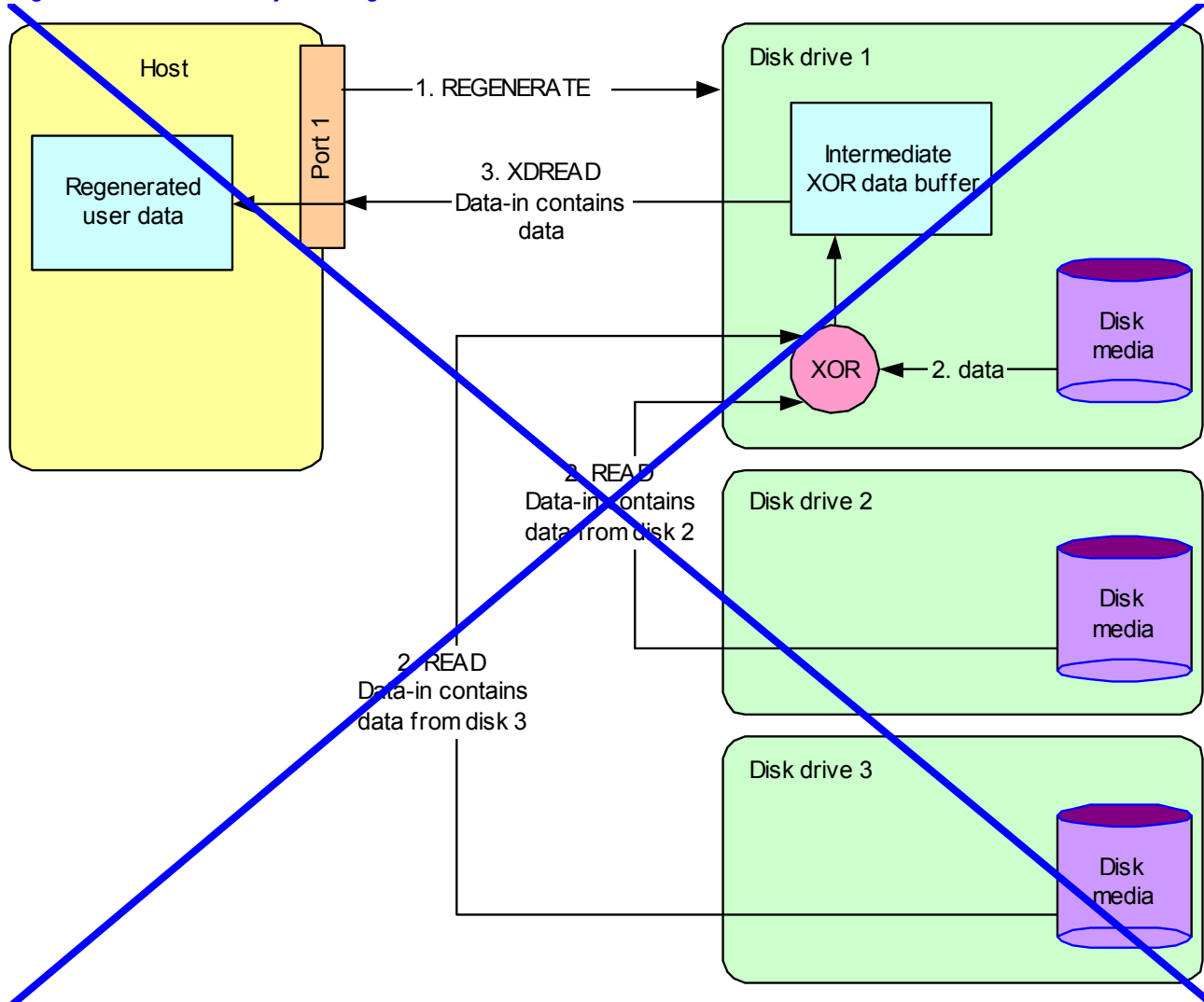

Figure 0.1 — Update write operation (third-party)

0.1.2 Regenerate operation

Figure 0.2 illustrates a third party regenerate operation. The example uses a host and three disk devices. In this example, all three disk devices are on the same SCSI physical interconnect, and thus are capable of peer-to-peer interaction. Three SCSI commands are used: REGENERATE, READ, and XDREAD.

The host begins by issuing a REGENERATE command to disk device 1. Disk device 1 assumes initiator role and sends READ commands to disk device 2 and disk device 3. It also concurrently reads data from its own medium. Disk device 1 performs an XOR operation on the data from all three disk devices and stores the

resulting intermediate XOR data (regenerated user data) in its buffer memory. The host retrieves this regenerated user data by sending an XDREAD command to disk device 1.

Figure 0.2 — Regenerate operation (third-party)

0.1.3 Rebuild operation

Figure 0.3 illustrates a third-party rebuild operation. The example uses a host, two disk devices as the source devices, and one disk device as the device being rebuilt. In this example, all three disk devices are on the same SCSI physical interconnect, and thus are capable of peer to peer interaction. Two SCSI commands are used: REBUILD and READ.

The host begins by issuing a REBUILD command to the device being rebuilt (disk device 1). Disk device 1 assumes the initiator role and issues READ commands to the source devices (disk device 2 and disk device

3). Disk device 1 performs an XOR operation on the data received from these two commands and writes the resulting rebuilt data to its medium.
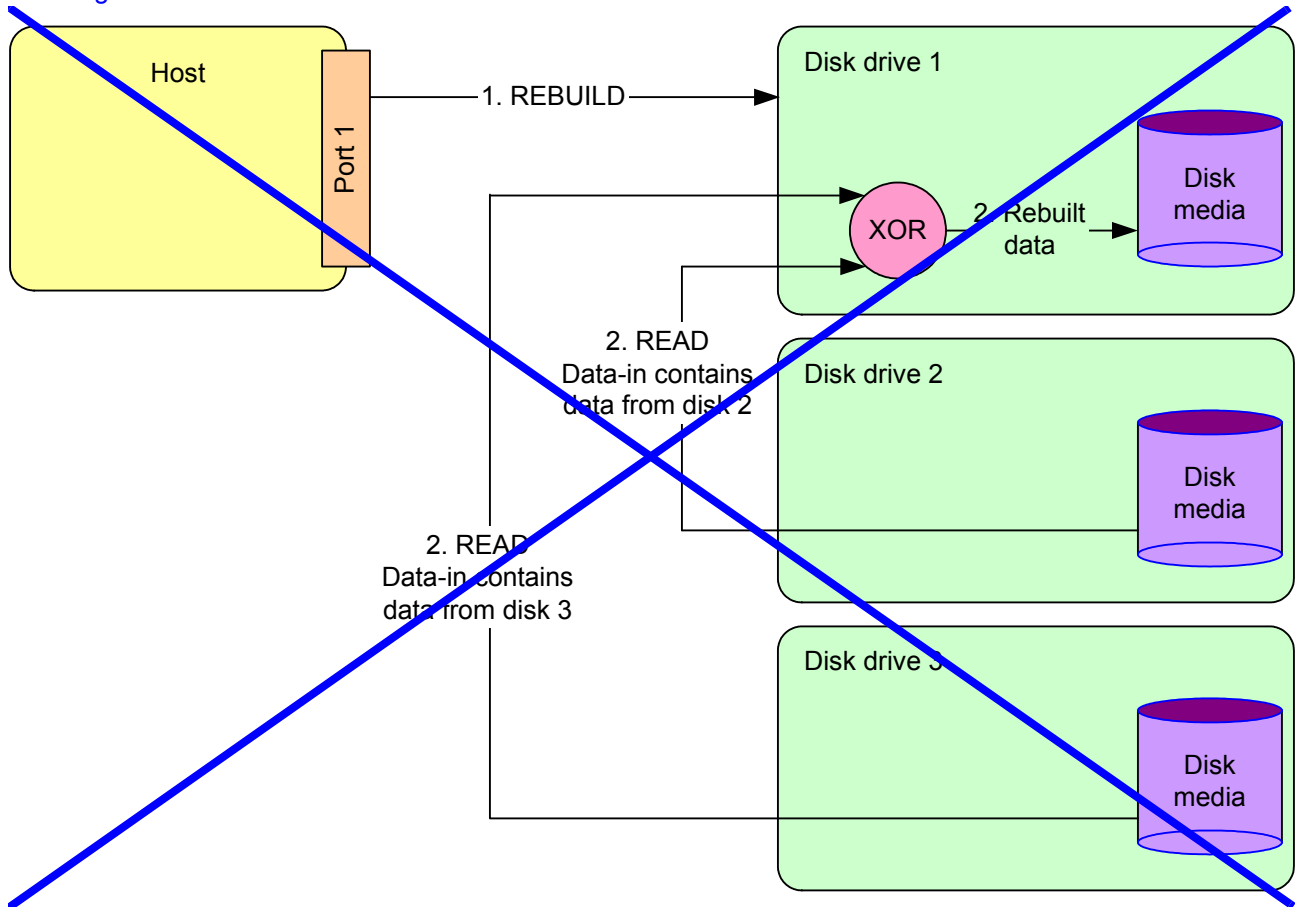

Figure 0.3 — Rebuild operation (third-party)

0.2 Hybrid subsystem XOR operations

0.2.1 Regenerate operation

Figure 0.5 illustrates a regenerate operation on a hybrid system. The example uses a host and four disk devices as source devices. In this example, two of the disk devices are on one SCSI physical interconnect, and thus are capable of peer-to-peer interaction between themselves. The two other disk devices are on a different SCSI physical interconnect and are also capable of peer-to-peer interaction between themselves. Three SCSI commands are used: REGENERATE, READ, and XDREAD.

The host begins by issuing a REGENERATE command to disk device 1. Disk device 1 assumes initiator role and sends a READ command to disk device 2. Disk device 1 also concurrently reads data from its own medium. Disk device 1 performs an XOR operation on the data from the two disk devices and stores the resulting partially regenerated user data in its buffer memory. The host retrieves this partially regenerated user data by sending an XDREAD command to disk device 1.

The host then issues a REGENERATE command to disk device 3 with the partially regenerated user data that was obtained from disk device 1 (with the XDREAD command). Disk device 3 assumes the initiator role and sends a READ command to disk device 4. Disk device 3 also concurrently reads data from its own medium. Disk device 3 performs an XOR operation on the data from disk device 3, disk device 4, and the partially

regenerated user data (from disk device 1). The host retrieves this regenerated user data by sending an XDREAD command to disk device 3.
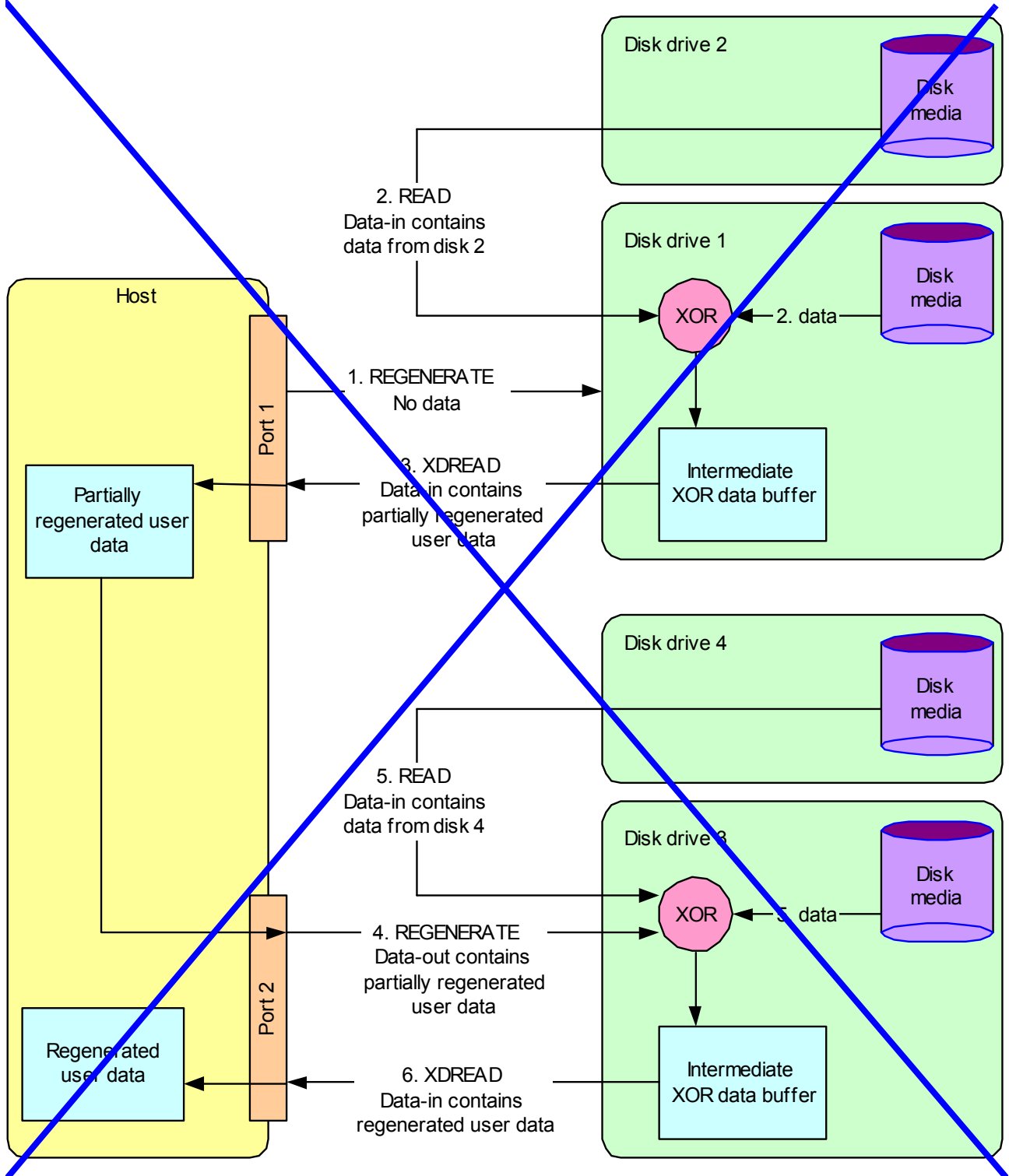

Figure 0.4 — Regenerate operation (hybrid subsystem)

0.2.2 Rebuild operation

Figure 0.5 illustrates a rebuild operation on a hybrid system. In this example, two of the disk devices are on one SCSI physical interconnect, and thus are capable of peer to peer interaction between themselves. The two other disk devices are on a different SCSI physical interconnect and are also capable of peer to peer

SBC, SPC, Obsolete Third Party XOR Commands

interaction between themselves. Four SCSI commands are used: REGENERATE, READ, XDREAD, and REBUILD.

The host begins by issuing a REGENERATE command to a source device (disk device 1). Disk device 1 assumes the initiator role and issues a READ command to disk device 2. Disk device 1 also concurrently reads data from its own medium. Disk device 1 performs an XOR operation on the data from the two disk devices and stores the resulting partially rebuilt data to its buffer memory. The host retrieves this partially rebuilt data by sending disk device 1 an XDREAD command.

The host then issues a REBUILD command to disk device 3 (device being rebuilt) with the partially rebuilt data that was obtained from disk device 1 (with the XDREAD command). Disk device 3 assumes the initiator role and sends a READ command to disk device 4. Disk device 3 also concurrently reads data from its own

medium. Disk device 3 performs an XOR operation on the data from disk device 3, disk device 4, and the partially rebuilt data (from disk device 1). The resulting rebuilt data is written to the medium in disk device 3.

Disk drive 2

Disk media

2. READ
Data-in contains
data from disk 2

Disk drive 1

XOR ← 2. data — Disk media

Host

Port 1

1. REGENERATE
No data

Partially rebuilt
user data

3. XDREAD
Data-in contains
partially rebuilt
user data

Intermediate
XOR data buffer

Disk drive 4

Disk media

5. READ
Data-in contains
data from disk 4

Disk drive 3

XOR → 5.1 Rebuilt data → Disk media

Port 2

4. REBUILD
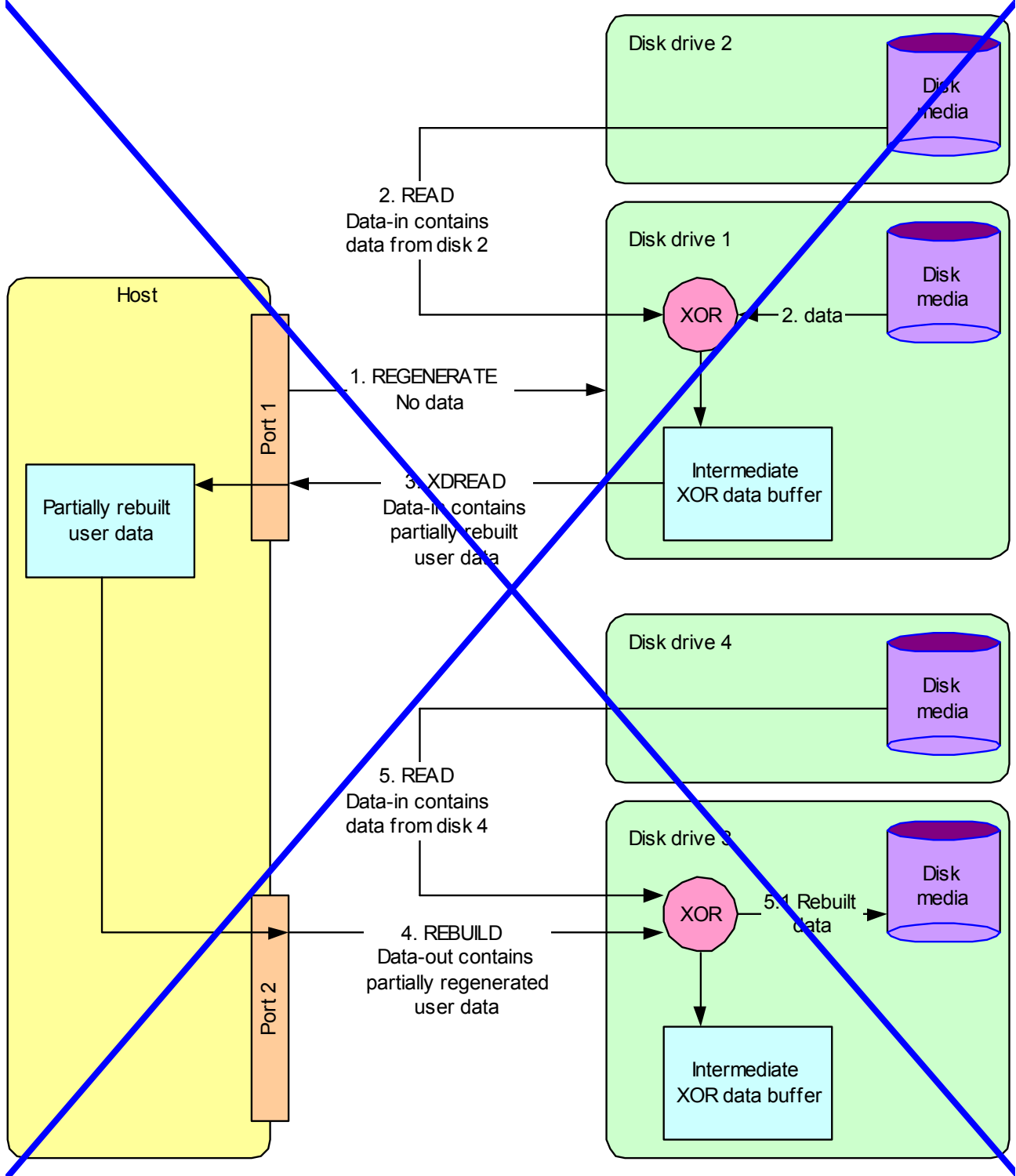Data-out contains
partially regenerated
user data

Intermediate
XOR data buffer

Figure 0.5 — Rebuild operation (hybrid subsystem)