Date: December 18, 2000

To: T10 Committee (SCSI)

From: Jim Hafner (IBM) (hafner@almaden.ibm.com)

Subject: Long Identifiers in SPC-3, SAM-2, SBC-2 and other XOR issues

**ABSTRACT**:

As noted in T10/00-248r2 (“SBC-2 issues” by Robert Elliot), there are problems with the specification of logical units and targets in some of the XOR commands in SBC-2. Additionally, as noted in T10/00-279r0 (“Large SCSI Device Identifiers” by Ed Gardner), there are limitations in SAM-2 and SPC-3 that fix device identifiers to a maximum of 64bits and these are insufficient for both of the forthcoming iSCSI and SRP protocols. This proposal is aimed at resolving these problems. The next section details some of the existing problems and limitations. The subsequent sections provide an outline of the proposed solution and then the details of that solution.

The specific formats for long identifiers in parameter data is roughly sketched in this proposal, within a designed framework (see 3.6.1, especially Table xx5). Editorial and technical proposals in this area are welcome.

## 1.0 Summary existing problems and limitations of 64bit identifiers

The following summarizes the current state of affairs:

a) SBC-2 REBUILD and REGENERATE commands (both 16 and 32 byte versions) use only a 64bit identifier for a third party target device but make no explicit reference to a logical unit (e.g., by LUN) at that target.

b) SBC-2 XDWRITE EXTENDED (both 16 and 32 byte versions) explicitly restrict their use to LUN0 and provide either a 1 byte third party address format or a pointer to an undefined table format to resolve the address; this has the dual problems of being unspecified (table) and restricting to only LUN0 which does not necessarily have a consistent meaning in the presense of LUN Mapping.

c) SPC-2 EXTENDED COPY leaves only16 bytes in a target descriptor for a target identifier; 16bytes is insufficient for some long identifiers such as IPv6 address with IPport qualifier.

d) SAM-2 specifies that device identifiers shall be at most 64bits and this is also insufficient to deal with IPv6 style addresses, for example.

## 2.0 Summary of proposed changes

The following summarizes the proposed changes:

a) SAM-2: remove any specification that target identifiers are 64 bit and leave that to the protocols.

b) SBC-2: Add language to REBUILD (16) and REGENERATE (16) commands to specify that the LUN value should be zero (that is, add language similar to that in XDWRITE EXTENDED).

c) SBC-2: Change the parameter data format for the REBUILD (32) and REGENERATE (32) to replace the 8 byte SERVICE DEVICE ADDRESS field with a 32 byte TARGET DESCRIPTOR field that is defined as in the EXTENDED COPY command of SPC-3.

d) SBC-2: Convert TABLE ADDRESS bit in the XDWRITE EXTENDED (16) and (32) to reserved (it can't have been implemented as the table reference is explicitly undefined).

e) SBC-2: Define a new XDWRITE EXTENDED (64) which contains a 32 byte TARGET DESCRIPTOR field that is defined as in the EXTENDED COPY command of SPC-3.

f) SPC-3: Define a new pair of commands called REPORT ALIASES and CHANGE ALIASES that allow an initiator to define a mapping of an 8 byte alias to a longer formatted device identifier. This proposal identifies a new Service Action in each of the MAINTENANCE IN and MAINTENANCE OUT commands for this purpose.

g) SPC-3: Define a new EXTENDED COPY target descriptor called an Alias Target Descriptor. In this descriptor, the target device is identified by an 8 byte value that should be found in the alias list of the target (as defined by CHANGE ALIASES commands); the logical unit is identified as is done in (all but one) existing target descriptors as either a LUN value or a Proxy Token.

This would result in the following:

a) clear up the ambiguity in existing REBUILD and REGENERATE commands;

b) leverage the extensible and defined third party target descriptors of EXTENDED COPY to the other third party commands (REBUILD, REGENERATE, and XDWRITE EXTENDED);

c) provide a generic and extensible method to map long addresses or identifiers to shorter identifiers that can fit in existing target descriptors.

We propose the following properties of the alias list and the REPORT ALIASES and CHANGE ALIASES commands.

a) The alias list is by logical unit (that is each device server maintains its own alias list).

b) The alias list is either shared by all initiators or is initiator-specific (analogous to mode pages in this regard only); the choice is implementation dependent and the implementation choice is noted in the returned data of the REPORT ALIASES command.

c) The CHANGE ALIASES command should be blocked by reservations; there is no particular requirement for the REPORT ALIASES to also be blocked by reservations, however.

NOTE: if the target does not support initiator-specific alias lists, then the use of reservations can help to guarantee that the state of a shared list doesn't change for an initiator during completion of a series of tasks requiring a specific state for that list.

d) The alias list is volatile and is cleared under any event that resets the logical unit.

e) Long identifiers in the list can be variable length, up to a maximum of 65532 bytes. The specific format is defined by a protocol code and type code. The protocol code specifies the transport protocol and the type code specifies a format specific to that protocol. [This allows each protocol to have a private name space where they can define additional long identifiers without changes to SPC-x.]

f) The REPORT ALIASES command will always report the entire alias list applicable to the requesting initiator (the shared list or that initiator's specific list, depending on the implementation). There is no defined mechanism for requesting any single entry in the list (this just cluttered the design with no major functional gain).

g) For opcodes, we propose new service actions in MAINTENANCE IN and MAINTENANCE OUT (of SCC-2); we also propose that these be documented in SPC-3 in a manner analogous to SET DEVICE IDENTIFIER and REPORT DEVICE IDENTIFIER).

## 3.0 Detailed description of proposed changes

### 3.1 Changes for SAM-2 (sam2r15)

Change the glossary definition 3.1.84 of "SCSI device identifier" to include the phrase "or name" and to remove any reference to the size of the identifier. The proposed wording is:

> **3.1.84 SCSI device identifier:** An address or name by which an SCSI device is referenced within a domain. Depending on the device model in use, the SCSI device identifier is either an Initiator Identifier or a Target Identifier (see 4.7.3).

The first change allows identifiers to be WWUniqueNames or for names such as IP Names, neither of which are really addresses from the transport protocol's point of view.

Change all references to "64 bit" size specifications from the document:

> **Clause 4.7.1**: An Initiator Identifier is a field containing ~~up to 64 bits that is~~ a SCSI device identifier for the initiator device...
>
> **Clause 4.7.2**: A Target Identifier is a field containing ~~up to 64 bits that is~~ a SCSI device identifier for the target device...

### 3.2 Changes for SBC-2 (sbc2r02) REBUILD (16) and REGENERATE (16)

Add the underlined words to the indicated paragraph of clause 5.1.13 that deals with the specification of the source device (though this is only the clause for REBUILD (16), it covers the clause for REGENERATE (16) by reference):

> The source device address field specifies an ANSI X3.270 SAM compliant target identifier (of no more than 64 bits) of a device that is a data source. The implied LUN at the target shall be zero.

Change the title of Table 37 from “REBUILD and REGENERATE parameter data” to “REBUILD (16) and REGENERATE (16) parameter data”.

## 3.3 Changes for SBC-2 (sbc2r02) REBUILD (32) and REGENERATE (32)

### 3.3.1 REBUILD (32), clause 5.1.14

Replace the paragraph immediately following the current Table 39 which reads

> See the REBUILD (16) command (5.1.13), Table 40, and SPC-2 for a description of the fields in this command.

with

> See the REBUILD (16) command (5.1.13) and SPC-2 for a description of the fields in this command. Tables 40 and 41 define the parameter data format for the REBUILD (32) and REGENERATE (32) commands.

and replace Table 40 with the following two tables and additional text.

[EDITOR’S NOTE: Table 41 as specified in the above paragraph is a new table, so the appropriate renumbering of subsequent tables will have to be performed.

]

Table 40. 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 | | | | | | | |
| 3 | SOURCE DESCRIPTOR /PAD LENGTH | | | | | | | LSB |
| | SOURCE DESCRIPTOR(S) (if any) | | | | | | | |
| 4<br>43 | SOURCE DESCRIPTOR (first) | | | | | | | |
| | | | | | | | | |
| 40*x*-36<br>40*x*+3 | SOURCE DESCRIPTOR (last) | | | | | | | |
| 40*x*+4<br>40*x*+*y*+3 | PAD, if any (length *y*) | | | | | | | |
| 40*x*+*y*+4 | MSB | | | | | | | |
| 40*x*+*y*+*z*+3 | INTERMEDIATE DATA, if any (length *z*) | | | | | | | LSB |

The SOURCE DESCRIPTOR format is specified in Table 41. All other fields in the parameter data are as defined in 5.1.13.

Table 41. REBUILD (32) and REGENERATE (32) source descriptor format (32 Byte source device address and 8 Byte LBA version).

| Byte | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | MSB | | | | | | | |
| 31 | SOURCE DEVICE ADDRESS (32 bytes) | | | | | | | LSB |
| 32 | MSB | | | | | | | |
| 40 | SOURCE STARTING LOGICAL BLOCK ADDRESS (8 bytes) | | | | | | | LSB |

The SOURCE DEVICE ADDRESS specifies a third party logical unit as a data source. The format of this conforms to one of the target descriptor formats of the EXTENDED COPY command as specified in SPC-2.

[EDITOR'S NOTE: should that be SPC-3, both here and in the paragraph above?]

[EDITOR'S NOTE: Should there be additional language here to describe the rules for when target descriptors cannot be resolved by the target?]

### 3.3.2 REGENERATE (32), clause 5.1.16

Add the underlined words to the indicated paragraph at the end of 5.1.16.

> See the REGENERATE (16) command (5.1.15), Table 40 and Table 41, and SPC-2 for a description of the fields in this command.

[EDITOR'S NOTE: as above, should this be SPC-2 or SPC-3?]

### 3.4 Changes for SBC-2 (sbc2r02) for XPWRITE EXTENDED (16) and (32)

Convert the TABLE ADDRESS bit in the current Table 63 and Table 64 to Reserved. This bit was used to indicate that the target should consult some table to resolve the SECONDARY ADDRESS field.

[EDITOR'S NOTE: The current text specifically says "The lookup table is reserved for future definition" so changing to Reserved should not be a problem. If that is not the opinion of the committee, then converting to Obsolete would be acceptable.]

Change the paragraph immediately following Table 63 as indicated:

> See 4.2.1.9 for reservation requirements for this command. ~~A TABLE ADDRESS bit of zero indicates that t~~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 ~~and the Table Address bit is zero~~, the SECONDARY ADDRESS field specifies the least significant byte of the secondary target identifier. The upper bytes of the secondary target identifier shall be equal to the upper bytes of the target identifier of the XDWRITE EXTENDED target.

Delete the paragraph following in that clause that specifies the meaning of the TABLE ADDRESS bit of one.

### 3.5 Addition to SBC-2 (sbc2r02) of new XDWRITE EXTENDED (64)

In the Service action code assignments Table 1, add an entry for the XDWRITE EXTENDED (64) service action code (0007h) and a reference to a new clause 5.1.38, and fix the reference to existing clause 5.1.39 in the XPWRITE (32) command of Table 1. Insert the following clause as a new clause 5.1.38.

> #### 5.1.38 XDWRITE EXTENDED (64) command
>
> The XDWRITE EXTENDED (64) command (see Table xx) requests that the target XOR the data transferred with the data on the medium. The resulting XOR data may be subsequently sent to a secondary device using an XPWRITE (32) command.

Table xx. XDWRITE EXTENDED (64) command.

| Byte | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (7F) | | | | | | | |
| 1 | CONTROL | | | | | | | |
| 2<br>4 | RESERVED | | | | | | | |
| 5 | ENCRYPTION IDENTIFICATION | | | | | | | |
| 6 | RESERVED | | | | | | | |
| 7 | ADDITIONAL CDB LENGTH (38h) | | | | | | | |
| 8 | MSB | | | | | | | |
| 9 | SERVICE ACTION (0007h) | | | | | | | LSB |
| 10 | RESERVED | | | DPO | FUA | DISABLE WRITE | PORT CONTROL | |
| 11 | RESERVED | | | | | | | |
| 12 | MSB | | | | | | | |
| 43 | SECONDARY ADDRESS DESCRIPTOR (32 bytes) | | | | | | | LSB |
| 44 | MSB | | | | | | | |
| 51 | LOGICAL BLOCK ADDRESS (8 bytes) | | | | | | | LSB |
| 52 | MSB | | | | | | | |
| 59 | SECONDARY LOGICAL BLOCK ADDRESS (8 bytes) | | | | | | | LSB |
| 60 | MSB | | | | | | | |
| 63 | TRANSFER LENGTH (4 bytes) | | | | | | | LSB |

[EDITOR'S NOTE: In this and in the 32 byte commands, should the Encryption Identification field be converted to Reserved, per the changes accepted for SPC-2?]

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-2.

[EDITOR'S NOTE: as always, should this be SPC-3 both above and below?]

See the XDWRITE EXTENDED (16) command (5.1.36) and SPC-2 for a description of all other fields in this command.

## 3.6 Addition to SPC-3 of REPORT ALIASES and CHANGE ALIASES commands

[EDITOR'S NOTE: all clause references to SPC-3 are actually from the SPC-2 rev 18 (spc2r18) version as no copy of SPC-3 was available at the time of this writing.]

### 3.6.1 A clause for REPORT ALIASES command for SPC-3

The following clause should be added to SPC-3.

## 7.xx REPORT ALIASES command

The REPORT ALIASES command (see Table xx1) requests that the device server send a report of the alias list applicable to the requesting initiator. The alias list itself is managed by the initiator via the CHANGE ALIASES command (7.yy). The REPORT ALIASES command is the REPORT ALIASES service action of the MAINTENANCE IN command. This service action is optional for all device types, however, if the CHANGE ALIASES sercice action of the MAINTENANCE OUT command is supported then the REPORT ALIASES service action of the MAINTENANCE IN command shall also be supported. Additional MAINTENANCE IN service actions (that apply to SCC-2 devices and devices that set the SCCS bit in their Standard Inquiry data) are defined in SCC-2. Only those service actions that are defined in this standard concern all SCSI devices. SCC-2 defines specific usage for bytes 4 and 5 and bit 1 in byte 10, however, these fields are reserved for the REPORT ALIASES command as defined in this standard.

Table xx1. REPORT ALIASES command.

| | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | OPERATION CODE (A3h) | | | | | | | |
| 1 | RESERVED | | | SERVICE ACTION (0Bh) | | | | |
| 2<br>3 | RESERVED | | | | | | | |
| 4<br>5 | SEE SCC-2 | | | | | | | |
| 6<br>9 | MSB<br>ALLOCATION LENGTH | | | | | | | LSB |
| 10 | RESERVED | | | | | | see SCC-2 | RSRVD |
| 11 | CONTROL | | | | | | | |

SCC-2 defines specific usages for bytes 4 and 5, and bit 1 in byte 10, however these fields are reserved for the REPORT ALIASES command defined by this standard.

The ALLOCATION LENGTH field indicates how much space has been allocated for the returned parameter data. If the length is not sufficient to contain all the parameter data, the first portion of the data shall be returned. This shall not be considered an error. The actual length of the parameter data can be computed from the ADDITIONAL LENGTH field in the parameter data. If the remainder of the parameter data is required, the application client should send a new REPORT ALIASES command with an ALLOCATION LENGTH field large enough to contain all the data.

The format of the returned parameter data is specified in Table xx2. It contains a header that indicates the amount of additional data, an indication as to whether

this alias list applies only the requesting initiator or is a shared alias list and the number of entries in the alias list.

Table xx2. REPORT ALIASES and CHANGE ALIASES parameter list.

| Byte | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | MSB | | | | | | | |
| 3 | | ADDITIONAL LENGTH (n-4) | | | | | | LSB |
| 4 | | RESERVED | | | | | | SHARED |
| 5 | | RESERVED | | | | | | |
| 6 | MSB | | | | | | | |
| 7 | | NUMBER OF ALIASES | | | | | | LSB |
| 8 | | | | | | | | |
| n | | ALIAS LIST ENTRIES | | | | | | |

The ADDITIONAL LENGTH field specifies the number of bytes following this field.

The SHARED bit is reserved for the CHANGE ALIASES command. A SHARED bit of one indicates that the list can be read, edited and referenced by all initiators. A SHARED bit of zero indicates that this list is specific to the requesting initiator. In this case, the list cannot be read, edited or referenced by another initiator.

The NUMBER OF ALIASES field indicates the number of ALIAS LIST ENTRIES. If there are no alias entries in the requesting initiator's applicable alias list, then the ADDITIONAL LENGTH field and the NUMBER OF ALIASES field shall be zero.

The ALIAS LIST ENTRIES contains a list of alias entries as specified in Table xx3. Each alias entry contains an 8 byte ALIAS value that the device server associates to the value in the DEVICE IDENTIFIER/ADDRESS field. The structure of the DEVICE IDENTIFIER/ADDRESS field is described in Table xx4. The number of bytes in each alias entry can be determined from the length field in the DEVICE IDENTIFIER/ ADDRESS field.

Table xx3. REPORT ALIASES and CHANGE ALIASES alias entry.

| Byte | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | MSB | | | | | | | |
| 7 | | ALIAS | | | | | | LSB |
| 8 | MSB | | | | | | | |
| n | | DEVICE IDENTIFIER/ADDRESS | | | | | | LSB |

.

Table xx4. DEVICE IDENTIFIER/ADDRESS format.

| Byte | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | PROTOCOL IDENTIFIER | | | | | | | |
| 1 | TYPE CODE | | | | | | | |
| 2 | MSB | | | | | | | |
| 3 | IDENTIFER/ADDRESS LENGTH (n-4) | | | | | | | LSB |
| 4 | MSB | | | | | | | |
| n | IDENTIFIER/ADDRESS | | | | | | | LSB |

The PROTOCOL IDENTIFIER field defines the transport protocol to which the IDENTIFIER/ADDRESS applies; the defined values of the PROTOCOL IDENTIFIER field are defined in Table 172 (of SPC-2 rev 18). The TYPE CODE defines the format of the IDENTIFIER/ADDRESS field with respect to the specific transport protocol. Table xx5 summarizes TYPE CODE values and IDENTIFIER/ADDRESS formats for some existing transport protocols. The IDENTIFIER/ADDRESS LENGTH specifies the number of bytes of the IDENTIFIER/ADDRESS field. This value shall be a multiple of 4. An IDENTIFIER/ADDRESS that is not a multiple of 4 bytes long shall be padded in the least significant bytes by zeros to the next largest multiple of 4 bytes long. The IDENTIFIER/ADDRESS field specifies a SCSI Target Identifier per the specific protocol.

Table xx5: PROTOCOL IDENTIFIER, TYPE CODE values and IDENTIFIER/ADDRESS formats.

| Protocol Identifier | Protocol Description | Type Code | Type Description | Format/length |
|---|---|---|---|---|
| 00h | FCP-n | 00h | WWPortName | 8 bytes |
| 00h | FCP-n | 01h | WWNodeName | 8 bytes |
| 01h | SPI-n | 00h | SCSI Address | 2 bytes, decimal address |
| 02h | SSA-SnP | 00h | Reserved | |
| 03h | SBP-n | 00h | Reserved | |
| 04h | SRP-n | 00h | Reserved | |
| 05h | iSCSI-n | 00h | IPv4 plus port | 12 bytes, binary: IPv4 address (4 bytes) + Rsvd (2 bytes) + port (2 bytes) + Rsvd (2 bytes) + Internet Protocol Number (2 bytes) |
| 05h | iSCSI-n | 01h | IPv6 plus port | 24 bytes, binary: IPv6 address (16 bytes) + Rsvd (2 bytes) + port (2 bytes) + Rsvd (2 bytes) + Internet Protocol Number (2 bytes) |
| 05h | iSCSI-n | 02h | iSCSI WWUI | TBD (might be variable length) |
| 05h | iSCSI-n | 03h | IPaddress:port | m+8 bytes, mixed: Address (UTF-8, m bytes, including zero pad) + Rsrv (2 bytes) + port (binary, 2 bytes) + Rsvd (2 bytes) + Internt Protocol Number (binary, 2 bytes) |
| 05h | iSCSI-n | 00h | URI | m bytes, UTF-8, including zero pad |
| 06h-FFh | Reserved | n/a | | n/a |

[EDITOR'S NOTE: By using a PROTOCOL IDENTIFIER and TYPE CODES, we are allowing the independent protocol documents to extend the naming conventions used in the IDENTIFIER/ADDRESS field without resorting to changes to SPC-3.]

[EDITOR'S NOTE: Table xx5 above is admittedly not well done. It is not clear whether the protocol documents should define these formats or whether they should be defined here. This can get very long-winded if defined here, however, some protocol documents are complete and can't get this information included. Editorial advice on this matter is welcome.]

[EDITOR'S NOTE: This proposal assumes that Table 172 of SPC-2 will be expanded to include SRP and iSCSI. What about SST both here and in Table 172?]

### 3.6.2 A clause for CHANGE ALIASES command for SPC-3

The following clause should be added to SPC-3.

#### 7.yy CHANGE ALIASES command

The CHANGE ALIASES command (see Table yy1) requests that the device server maintain and make changes to a table of associations between short (8 byte) alias identifiers and (possibly) longer SCSI device identifiers. The alias list can be queried by the initiator via the REPORT ALIASES command (7.xx). The CHANGE ALIASES command is the CHANGE ALIASES service action of the MAINTENANCE OUT command. This service action is optional for all device types, however, if the REPORT ALIASES sercice action of the MAINTENANCE IN command is supported then the CHANGES ALIASES service action of the MAINTENANCE OUT command shall also be supported. Additional MAINTENANCE OUT service actions (that apply to SCC-2 devices and devices that set the SCCS bit in their Standard Inquiry data) are defined in SCC-2. Only those service actions that are defined in this standard concern all SCSI devices. SCC-2 defines specific usage for bytes 4 and 5 and bit 1 in byte 10, however, these fields are reserved for the CHANGE ALIASES command as defined in this standard.

On successful completion of an CHANGE ALIASES command, the device server shall maintain a mapping or association of each assigned 8 byte alias value to the paired (possibly) longer protocol-specific SCSI device identifier. This association shall be cleared under any event that includes a logical unit reset. The device server may optionally maintain a separate list for each initiator or may maintain a common, shared list for all initiators. The choice of shared or initiator-specific alias list is implementation dependent, but shall be reported via the SHARED bit in the REPORT ALIASES parameter data as described in 7.xx.

A CHANGE ALIASES command may add, change or remove entries from the applicable alias list. Changes to the applicable alias list indicated by a CHANGE ALIASES command affect only those entries that are specified in the parameter

data for the command. Alias list entries not specifically referenced in the parameter data remain unchanged.

NOTE: An application client may use alias values to reference SCSI devices in third party commands (e.g., the EXTENDED COPY command). The alias list provides a mechanism for eight byte third party device identifier fields to reference or indicate a third party device whose protocol-specific SCSI device identifier is longer than eight bytes. For example, an application client may use the CHANGE ALIASES command to establish an association between an alias value and a SCSI target device identifier. Then it may send an EXTENDED COPY command that use the alias target descriptor format (see 7.5.6.z) that includes this alias value. At the completion of the EXTENDED COPY command the application client should clear this entry from the device server's alias list by sending a subsequent CHANGE ALIASES command that requests association of the alias value to a null identifier as described below.

Table yy1. CHANGE ALIASES command.

| | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | OPERATION CODE (A4h) | | | | | | | |
| 1 | RESERVED | | | SERVICE ACTION (0Bh) | | | | |
| 2<br>3 | RESERVED | | | | | | | |
| 4<br>5 | SEE SCC-2 | | | | | | | |
| 6 | MSB | | | | | | | |
| 9 | PARAMETER LIST LENGTH | | | | | | | LSB |
| 10 | RESERVED | | | | | | see SCC-2 | RSRVD |
| 11 | CONTROL | | | | | | | |

SCC-2 defines specific usages for bytes 4 and 5, and bit 1 in byte 10, however these fields are reserved for the REPORT ALIASES command defined by this standard.

The PARAMETER LIST LENGTH field specifies the length in bytes of the parameter data that shall be transferred from the application client to the device server. A parameter list length of zero indicates that no data shall be transferred and no changes are requested in the initiator's applicable alias list.

The format of the parameter data is specified in Table xx2. It contains a header that indicates the amount of additional data, the number of alias entries in the parameter data and the list of new, changed or removed entries.

Besides the DEVICE IDENTIFIER/ADDRESS formats specified in 7.xx, an additional null identifier format may be used in the parameter data of the CHANGE ALIASES

command. This format shall consist of the DEVICE IDENTIFIER/ADDRESS header only with identifier length set to zero and no additional data. The application client may use this format to remove an alias entry from its initiator's applicable alias list.

If the device server has insufficient resources to make all requested changes to the applicable alias list, the device server shall make no changes to the applicable alias list and shall return CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INSUFFICIENT RESOURCES,

The device server shall not attempt to verify or validate any SCSI device identifier mapped to an alias value at the time of processing the CHANGE ALIASES command.

### 3.6.3 Additional changes to Reservation clause of SPC-3.

In addition to the above clauses, two new lines should be added to Table 8 (of SPC-2 rev 18) concerning the commands allowed in the presence of reservations. One line should be added for REPORT ALIASES and this line should match that for REPORT DEVICE IDENTIFIER (that is, "Allowed" in all cases). A second line should be added for CHANGE ALIASES; this line should match that for SET DEVICE IDENTIFIER (that is, "Conflict" in all cases except "From registered initiator (RO all types)" where it should be "Allowed").

[EDITOR'S NOTE: with all the "special" commands which are just service actions for MAINTENANCE IN/OUT defined in SPC-3, perhaps we can look for a cleaner approach to all these, with respect to proliferating clauses for each service action and with respect to reservation clause. I don't have a good suggestion, however.]

## 3.7 Addition to SPC-3 of a new EXTENDED COPY target descriptor

Add the following clause to the EXTENDED COPY clause on target descriptors (7.5.6).

[EDITOR'S NOTE: it has been suggested that ALL the target descriptors be moved to the model clause of EXTENDED COPY for cleaner reference by the XOR/SBC-x commands. I agree with this sentiment, but have no specific proposal to that effect.]

#### 7.5.6.zz Alias target descriptor format

Targets may be identified indirectly using the alias target descriptor format shown in Table zz1.

Table zz1. Alias target descriptor format.

| Byte | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | DESCRIPTOR TYPE CODE (E6h) | | | | | | | |
| 1 | RESERVED | | NUL | PERIPHERAL DEVICE TYPE | | | | |
| 2 | RESERVED | | | | | | | |
| 3 | RESERVED | | | | | | LU ID TYPE | |
| 4 | MSB | | | | | | | |
| 11 | LU IDENTIFIER | | | | | | | LSB |
| 12 | MSB | | | | | | | |
| 19 | TARGET DEVICE ALIAS | | | | | | | LSB |
| 20 | | | | | | | | |
| 27 | RESERVED | | | | | | | |
| 28 | | | | | | | | |
| 31 | Device type specific parameters | | | | | | | |

The DESCRIPTOR TYPE CODE, PERIPHERAL DEVICE TYPE and NUL fields and the device type specific parameters are described in 7.5.6.1.

The LU ID TYPE field and LU IDENTIFIER field are described in 7.5.6.1.

The TARGET DEVICE ALIAS field indicates an alias value in the requesting initiator's applicable alias list as managed by the CHANGE ALIASES command (see 7.yy) and maintained by the device server. If the value of the TARGET DEVICE ALIAS field is not in the requesting initiator's applicable alias list, the device server shall return CHECK CONDITION status, with sense key set to COPY ABORTED and additional sense code set to COPY TARGET DEVICE NOT REACHABLE.

The copy manager shall resolve the TARGET DEVICE ALIAS values in all alias target descriptors of a single EXTENDED COPY command as an atomic operation with respect to the applicable alias list.