

To: X3 Secretariat
Attention: Lynn Barra
1250 Eye Street N.W., Suite 200
Washington, D.C. 20005-3922

cc: SCSI reflector

From: Charles Binford
Symbios Logic
3718 N. Rock Road
Wichita, KS 67226-1397

Charles.Binford@Symbios.com
(316) 636-8566

Subject: SCC Public review comments

All comments refer to Rev 4 of SCSI-3 Controller Commands (SCC), X3T10 Project 1047D.

Cmt: 1. 5.2.1 SCSI Storage array addressing, p 16

By specifying, in table 1, that a LUN_P follows the peripheral device addressing format, table 6, then all p_extents (and thus ps_extents) are limited to single LUN devices. For example the LUN_P field in table 13, Data format of P_EXTENT DESCRIPTOR must follow the described addressing format, table 6, thus it cannot specify a non-zero LUN.

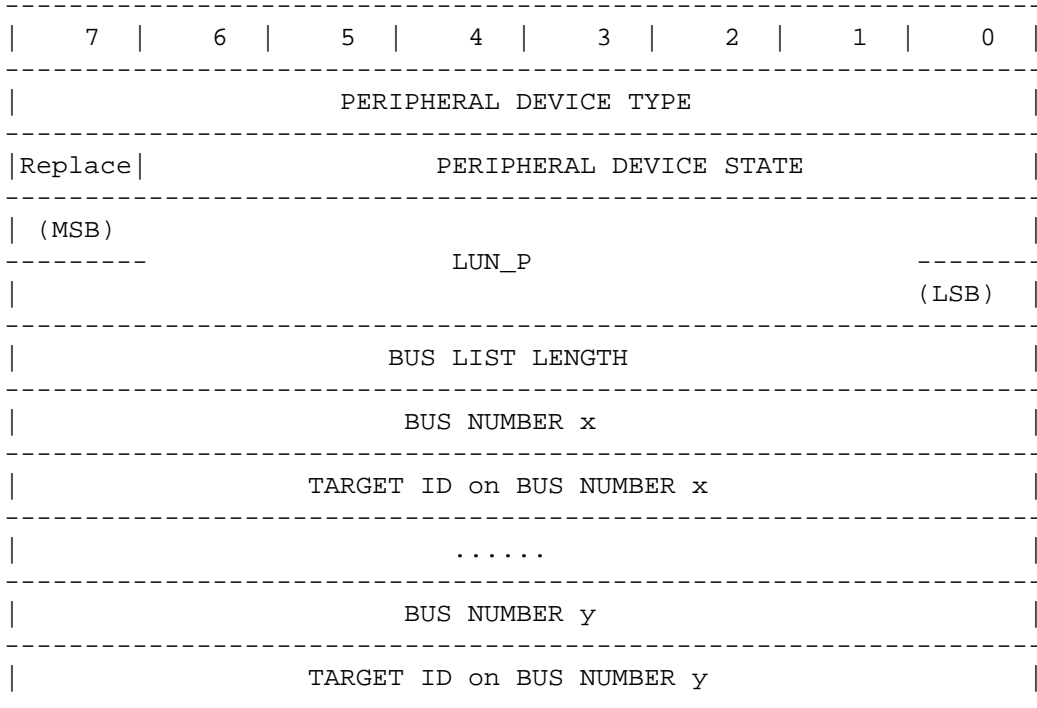
One potential fix would be to change the format of a LUN_P to a four byte LUN structure. Four bytes would accommodate two levels of the eight byte LUN structure of table 3. With this scheme a p_extent could be based upon a LUN_V of a second level SACL. If the LUN_P was always specified as the entire eight byte structure then p_extents could be based upon a device three or four levels deep, but I think restricting p_extents on the current level to be defined in terms of a "LUN" on the next level is sufficient.

Cmt: 2. 5.2.1.1 SCSI-3 storage array base address, p 16

This section should indicate the base address (LUN 0) shall report Array Controller Device for the peripheral device type in the standard inquiry data.

Cmt: 3. 5.2.1.4 Peripheral device address method, p 18,19

Table 6 describes the Peripheral device addressing and specifies BUS NUMBER and TARGET/LUN fields. In light of the dual port configurations of SSA and FC-AL devices along with the potential of dual ported SPI devices I believe BUS NUMBER as an integral part of the LUN_P is misplaced. I would propose that a LUN_P format use a 14 bit field similar to a LUN_V. The SACL would be responsible for tracking which bus a particular device was on. A variable number of BUS NUMBER/TARGET ID pairs could be reported as part of the REPORT PERIPHERAL DEVICE action. Specifically, I would change Table 25, Format of PERIPHERAL DEVICE DESCRIPTOR to the following:



Part of the rationale for this change is I believe BUS NUMBER and TARGET ID is important to the host for configuring purposes (to ensure maximum performance and redundancy by spreading a LUN_R across several buses). I want to avoid, however, the case where a SACL has been required to hide a path to a device (because of only one BUS NUMBER in the current LUN_P) but is actually routing all requests to the hidden bus and is implicitly lying to the host.

Cmt: 4. 5.2.1.7 Volume set address method, p 19

This section should state (possibly in a note) that if a volume set supports the Inquiry command it shall indicate a valid SCSI-3 peripheral device type, e.g. direct access or streaming device type.

Cmt: 5. 5.2.2.4 Covering of objects, p 21

Section 5.2.2.13 says that spares automatically exchange. Since "covering" is part of the definition of spares I conclude that covering involves the ability to perform an automatic exchange. If this interpretation is correct, then 5.2.2.4, paragraph 2, sentence 1 should be reworded to specifically reference an automatic exchange.

Cmt: 6. 5.2.2.5 Exchanging objects, p 21

It is clearly stated that the new object takes on all characteristics of the old object. A statement needs to be added indicating the characteristics of the old object after the exchange. Does the old object obtain the new objects previous characteristics, stay the same, or is it explicitly undefined?

Cmt: 7. 5.2.2.13 Spares, p 27

The first full paragraph on page 27 starting with "After an automatic...." states that the spare takes on the essential characteristics. In sections 5.2.2.4 and 5.2.2.5 it is stated that all characteristics are taken. These sections need to be consistent.

Cmt: 8. 5.2.2.13 Spares, p 27

In the last sentence of the first full paragraph on page 27 starting with "After an automatic...." reads "The failed p_extent, peripheral device, or component device shall be marked as failed." As there is no state of "failed", the sentence should be changed to read "... marked as broken."

Cmt: 9. 5.2.2.13, note 8 and 9, p 27

The wording of these notes is unclear. Note 8 claims the spare moves to a new physical position. It seems to me that the spare stays in the same position. Note 9 claims the spare stays in the same physical position. It seems to me that the spare moves.

Please add labels to the objects in the description (e.g. p_extent X, spare Y) so the reader can follow the flow. Also, more background information is needed to set up the example. I assume a user has just physically replaced a failed part with a new one, and the examples start with the controller taking the appropriate action - automatic exchange, etc., but I am not sure of this interpretation.

Cmt: 10. 6.1.1.4 REPORT PERIPHERAL DEVICE service action, p 46

Add an option bit to the REPORT PERIPHERAL DEVICE command to specify that the report shall only (also) include devices with a state of Not Available. The intent of this request is to give the host access to the list of unpopulated drive bays. This also allows a host to determine the number of channel/busses on a controller.

Cmt: 11. 6.1.1.4 REPORT PERIPHERAL DEVICE service action, p 46

The description for the REPORT PERIPHERAL DEVICE service action should indicate that only one PERIPHERAL DEVICE DESCRIPTOR, Table 25, shall be reported for each physical device, even if the device contains multiple LUNs.

Cmt: 12. 6.1.1.5 REPORT PERIPHERAL DEVICE ASSOCIATIONS service action, p 48

The description for the REPORT PERIPHERAL DEVICE ASSOCIATIONS service action should indicate that one PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR, Table 28, shall be reported for each LUN of each physical device with an association.

Cmt: 13. 6.1.1.7 REPORT STATES service action, p 58

Table 37, Redundancy group states. Add a state of Spare in Use.

Cmt: 14. 6.1.1.7 REPORT STATES service action, p 59

Table 38, Peripheral device and p_extent states. Add a state of Rebuild in Progress.

6.2.1.4 EXCHANGE P_EXTENT service action, p 65, 66

on the top of page 66.

Cmt: 16. 6.3.1.2 REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action, p 78
Table 65 Data format of PS_EXTENT DESCRIPTOR, page 78, needs to include a LUN_R field. This

from the same peripheral device, then two ps_extents may have exactly the same descriptor (table 65).
(This assumes the start LBA_PS of a ps_extent is 0 relative to the redundancy group, not 0 relative to

Cmt: 17.

Add a statement (possibly a note) to this section indicating that creating a redundancy group may not
result in unassigned ps_extents. The SACL may automatically create default volume sets as a result of

Cmt: 18.

In the third full paragraph on page 83 starting "It is not required..." the last sentence states: "All units
between the beginning of the first block address of the p_extent and the START CHECK

Cmt: 19.

The last paragraph on page 91 says "For any uncorrectable verification failures...". The word
uncorrectable implies the VERIFY command would automatically perform a RECALCULATE

bit, but I would not want it be the normal action for VERIFY.

Cmt: 20. 6.5.1.1 REPORT VOLUME SETS service action, p 96

Table 84, page 96 refers to Table 65 for the description of the PS_EXTENT DESCRIPTOR field. The

The START LBA_PS is no longer "unassigned" if the context is report volume sets.

Cmt: 21. 6.6.1.6 VERIFY VOLUME SET CHECK DATA service action, p 105

In Table 94, the bits in byte 10 are not in consistent positions compared to other commands. VERIFY

fields (see tables 31 and 72 for examples). CONTVER should use bit 2 to be consistent with table 79.

Cmt: 22. 6.7.1 SPARE (IN) command service actions, p 108

REPORT P_EXTENT SPARE service action and REPORT PERIPHERAL DEVICE/COMPONENT

being covered is the state of the spare is Spare In Use.