

Date: April 18, 2004
To: T10 Technical Committee
From: Keith Holt (LSI Logic)
Subject: SBC-2 Data Protection Option to Check Only the Logical Block Guard

1. Introduction

The RDPROTECT and WRPROTECT data protection fields were introduced in Revision 2 of the Simplified End-to-End Data Protection Proposal (03-176r2, July 31, 2003). The field sizes were two bits each, which allowed for four different protection options. In the August 2003 Data Integrity Study Group meeting (see 03-281r0 for the meeting minutes), there was consensus that three of the options had to be as follows:

1. "No protection information included."
2. "Protection information included, check all protection fields."
(The refinement for APPLICATION TAG ownership came later.)
3. "Protection information included, don't check any of the protection fields."

The pros and cons of various usages for the fourth option were discussed. In the end, the group recommended that the fourth option be defined as "check the REFERENCE TAG, but don't check BLOCK GUARD" in order to satisfy the usage model that called for the guard to be a checksum, rather than CRC. This left out the option of "check the BLOCK GUARD, but don't check any other fields." The thought was that this case could always be covered by the "don't check anything" option. Two recent developments have led me to believe that we need to reconsider this decision.

The first is that in the November CAP meeting, it was suggested that the field sizes be changed to 3 bits each for future expansion. This change was made in the final draft of the proposal, 03-365r1. No additional protection options were added at this time. With the RDPROTECT and WRPROTECT field sizes in SBC-2 at 3 bits each, only 4 of the 8 possible combinations are defined, so the infrastructure is already in place to easily add an additional option.

The second development is that there has been considerable discussion in the last few CAP meetings regarding REFERENCE TAG usage models, especially in the context of layered virtualization devices between the host and media. SBC-2 Revision 13 requires that the REFERENCE TAG be equal to the lower 4 bytes of the LBA. There is a proposal being developed to allow the REFERENCE TAG to differ from the LBA, 03-307r6, 32 Byte Commands for End-to-End Data Protection.

Given the possibility that we may ultimately have some devices that support the "non LBA locked" model and some that don't, it would seem to be prudent to provide a means for the application client to instruct the device server to treat the REFERENCE TAG as opaque data. Currently, the only option available to accomplish this is the "don't check any of the protection fields" option. This means that the device server would not be able to check the BLOCK GUARD, either, resulting in decreased data protection and a loss of fault isolation. The purpose of this proposal is to define an option that will allow the application client to specify that the device server can check the BLOCK GUARD, but no other data protection fields.

2. Changes to SBC-2 Revision 13

5.9 READ (10) command

Editor's note: Modify Table 32 – RDPROTECT field as shown below. The requirements for LOGICAL BLOCK GUARD checking for value 100b are the same as those specified for value 001b, while the requirements for LOGICAL BLOCK APPLICATION TAG and LOGICAL BLOCK REFERENCE TAG checking are the same as those specified for value 011b.

Table 32 – RDPROTECT field (part 4 of 4)

Value	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information ^h	Protection Information VPD page bit value ^g	If check fails ^{d f} , additional sense code
100b ^b	Yes	Yes ^e	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
				GRD_CHK = 0	No check performed
			LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
			LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No ^a	No protection information available to transmit to the application client or for checking			
1001b-111b	Reserved				
<p>^a A read operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.</p> <p>^b If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.</p> <p>^c The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. The method for acquiring this knowledge is not defined by this standard.</p> <p>^d If an error is reported the sense key shall be set to ABORTED COMMAND.</p> <p>^e Transmit protection information to the application client.</p> <p>^f If multiple errors occur, the selection of which error to report is not defined by this standard.</p> <p>^g See the Protection Information VPD page (see SPC-3) for a description of the GRD_CHK, APP_CHK, and REF_CHK bits.</p> <p>^h If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, the checking of all protection information in the associated logical block shall be disabled.</p>					

5.22 VERIFY (10) command

Editor's note: Modify Tables 54, 55, 56 and 57 – VRPROTECT field as shown below. The requirements for LOGICAL BLOCK GUARD checking for value 100b are the same as those specified for value 001b, while the requirements for LOGICAL BLOCK APPLICATION TAG and LOGICAL BLOCK REFERENCE TAG checking are the same as those specified for value 011b.

Table 54 — VRPROTECT field with BYTCHK set to zero - checking protection information read from the medium (part x of x)

Value	Logical unit formatted with protection information	Field in protection information ^g	Protection Information VPD page bit value ^f	If check fails ^{d e} , additional sense code
100b ^b	Yes	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition ^a		
1001b-111b	Reserved			

Table 55 — VRPROTECT field with BYTCHK set to one - checking protection information read from the medium (part x of x)

Value	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d e} , additional sense code
100b ^b	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition ^a		
1001b-111b	Reserved			

Table 56 — VRPROTECT field with BYTCHK set to one - checking protection information from the application client (part x of x)

Value	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d e} , additional sense code
100b ^b	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition ^a		
1001b-111b	Reserved			

Editor's note: As mentioned earlier, the requirements specified in Table 57 for LOGICAL BLOCK GUARD checking for value 100b are the same as those specified for value 001b, while the requirements for LOGICAL BLOCK APPLICATION TAG and LOGICAL BLOCK REFERENCE TAG checking are the same as those specified for value 011b. In SBC-2 revision 13, the comparison requirements for values 001b and 011b are identical for all three fields.

Table 57 — VRPROTECT field with BYTCHK set to one – byte-by-byte comparison requirements (part x of x)

Value	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails ^{c d} , additional sense code
100b ^b	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 1) ^e	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (APP_TAG_OWN = 0) ^f	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition ^a		
1001b-111b	Reserved			

5.26 WRITE (10) command

Editor's note: Modify Table 62 – WRPROTECT field as shown below. The requirements for LOGICAL BLOCK GUARD checking for value 100b are the same as those specified for value 001b, while the requirements for LOGICAL BLOCK APPLICATION TAG and LOGICAL BLOCK REFERENCE TAG checking are the same as those specified for value 011b.

Table 62 – WRPROTECT field (part 3 of 3)

Value	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails ^{d g} , additional sense code
100b ^b	Yes ^e	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No ^a	No protection information available to check		
1001b-111b	Reserved			

^a A write operation to a logical unit that supports protection information (see 4.15) and has not been formatted with protection information shall fail with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

^b If the logical unit does not support protection information the requested command should fail with CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

^c The device server may check the logical block application tag if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. The method for acquiring this knowledge is not defined by this standard.

^d If an error is reported the sense key shall be set to ABORTED COMMAND.

^e Device server shall preserve the contents of protection information (e.g., write to medium, store in non-volatile memory).

^f The device server shall write a properly generated CRC (see 4.15.3.2) into the LOGICAL BLOCK GUARD field, a properly calculated logical block reference tag into the LOGICAL BLOCK REFERENCE TAG field (see 4.15.2), and if the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, FFFFh into the LOGICAL BLOCK APPLICATION TAG field as it writes the logical block to the medium. If the APP_TAG_OWN bit is set to zero, the device server may set the LOGICAL BLOCK APPLICATION TAG field to any value.

^g If multiple errors occur, the selection of which error to report is not defined by this standard.