Date: 10/29/08

To: T10 Committee (SCSI)

From: George Penokie (LSI)

Subject: SBC-3: Adding a Protection Information Interval

1 Overview

This proposal adds the option to place protection information at regular intervals within a single logical block for types 2 and 3 protection.

This proposal also leaves room for, but does not define, the option of defining new protection types that would span multiple logical blocks.

1.0.1 protection information: Fields appended to each logical block or at specified intervals within a logical block or a group of logical blocks that contain a cyclic redundancy check (CRC), an application tag, and a reference tag. See 4.17.

1.1 Protection information model

1.1.1 Protection information overview

The protection information model provides for protection of user data while it is being transferred between a sender and a receiver. Protection information is generated at the application layer and may be checked by any object associated with the I_T_L nexus (see SAM-4). Once received, protection information is retained (e.g., written to medium, stored in non-volatile memory, or recalculated on read back) by the device server until overwritten. Power loss, hard reset, logical unit reset, and I_T nexus loss shall have no effect on the retention of protection information.

Support for protection information shall be indicated in the PROTECT bit in the standard INQUIRY data (see SPC-4).

If the logical unit is formatted with protection information and the EMDP bit is set to one in the Disconnect-Reconnect mode page (see SPC-4), then checking of the logical block reference tag within a service delivery subsystem without accounting for modified data pointers and data alignments may cause false errors when logical blocks are transmitted out of order.

Protection information is also referred to as the data integrity field (DIF).

1.1.2 Protection types

1.1.2.1 Protection types overview

The content of protection information is dependent on the type of protection to which a logical unit has been formatted.

The type of protection supported by the logical unit shall be indicated in the SPT field in the Extended INQUIRY Data VPD page (see SPC-4). The current protection type shall be indicated in the P_TYPE field in the READ CAPACITY(16) command (see 5.13).

An application client may format the logical unit to a specific type of protection using the FMTPINFO field and the PROTECTION FIELD USAGE field in the FORMAT UNIT command (see 5.2).

An application client may format the logical unit to place protection information at intervals other than on logical block boundaries using the PROTECTION INFORMATION DISTRIBUTION field in the FORMAT UNIT command (see 5.2).

The medium access commands are processed in a different manner by a device server depending on the type of protection in effect. When used in relation to types of protection, the term “medium access commands” is defined as the following commands:

a) ORWRITE;
b) READ (10);
c) READ (12);
d) READ (16);
e) READ (32);
f) VERIFY (10);
g) VERIFY (12);
h) VERIFY (16);
i) VERIFY (32);
j) WRITE (10);
k) WRITE (12);
l) WRITE (16);
m) WRITE (32);
n) WRITE AND VERIFY (10);
o) WRITE AND VERIFY (12);
p) WRITE AND VERIFY (16);
q) WRITE AND VERIFY (32);
r) WRITE SAME (10);
s) WRITE SAME (16);
t) WRITE SAME (32);
u) XDWRITE (10);
v) XDWRITE (32);
w) XDWRITEREAD (10);
x) XDWRITEREAD (32);
y) XPWRITE (10);
z) XPWRITE (32);
aa) XDREAD (10); and
ab) XDREAD (32).

The device server may allow the READ (6) command (see 5.7) and the WRITE (6) command (see 5.26) regardless of the type of protection to which the logical unit has been formatted.

1.1.2.2 Type 0 protection

Type 0 protection defines no protection over that which is defined within the transport protocol.

A logical unit that has been formatted with protection information disabled (see 5.2) or a logical unit that does not support protection information (i.e., the PROTECT bit set to zero in the Standard INQUIRY data (see SPC-4)) has type 0 protection.

If type 0 protection is enabled and the RDPROTECT field, WRPROTECT field, VRPROTECT field, or ORPROTECT field is set to a non-zero value, then media commands are invalid and may be terminated by the device server with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If type 0 protection is enabled and the RDPROTECT field, WRPROTECT field, VRPROTECT field, or ORPROTECT field is set to a zero value, then the following media commands are invalid and shall be terminated by the device server with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE:

a) READ (32);
b) VERIFY (32);
c) WRITE (32);
d) WRITE AND VERIFY (32); and
e) WRITE SAME (32).

1.1.2.3 Type 1 protection

Type 1 protection:

a) defines the content of the LOGICAL BLOCK GUARD field;
b) does not define the content of the LOGICAL BLOCK APPLICATION TAG field; and
c) defines the content the LOGICAL BLOCK REFERENCE TAG field.

If type 1 protection is enabled, then the following media commands are invalid and shall be terminated by the device server with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE:

a) READ (32);
b) VERIFY (32);
c) WRITE (32);
d) WRITE AND VERIFY (32); and
e) WRITE SAME (32).

For valid medium access commands in which the RDPROTECT field, WRPROTECT field, VRPROTECT field, or ORPROTECT field is set to:

a) zero, the data-in buffer and/or data-out buffer associated with those commands shall consist of logical blocks with only user data; or
b) a non-zero value, the data-in buffer and/or data-out buffer shall consist of logical blocks with both user data and protection information.

1.1.2.4 Type 2 protection

Type 2 protection:

a) defines the content of the LOGICAL BLOCK GUARD field;
b) does not define the content of the LOGICAL BLOCK APPLICATION TAG field; and
c) defines, except for the first logical block addressed by the command, the content of the LOGICAL BLOCK REFERENCE TAG field.

If type 2 protection is enabled and the RDPROTECT field, WRPROTECT field, VRPROTECT field, or ORPROTECT field is set to a non-zero value, then the following media commands are invalid and shall be terminated by the device server with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE:

a) ORWRITE;
b) READ (10);
c) READ (12);
d) READ (16);
e) VERIFY (10);
f) VERIFY (12);
g) VERIFY (16);
h) WRITE (10);
i) WRITE (12);
j) WRITE (16);
k) WRITE AND VERIFY (10);
l) WRITE AND VERIFY (12);
m) WRITE AND VERIFY (16);
n) WRITE SAME (10);
o) WRITE SAME (16);
p) XDWRITE (10);
q) XDWRITE (32);
r) XDWRITEREAD (10);
s) XDWRITEREAD (32);
t) XPWRITE (10);
u) XPWRITE (32);

v) XDREAD (10); and
w) XDREAD (32).

For valid medium access commands in which the RDPROTECT field, WRPROTECT field, VRPROTECT field, or ORPROTECT field is set to:

a) zero, the data-in buffer and/or data-out buffer associated with those commands shall consist of logical blocks with only user data; or
b) a non-zero value, the data-in buffer and/or data-out buffer shall consist of logical blocks with both user data and protection information.

1.1.2.5 Type 3 protection

Type 3 protection:

a) defines the content of the LOGICAL BLOCK GUARD field within the logical blocks of the data-in buffer and/or data-out buffer;
b) does not define the content of the LOGICAL BLOCK APPLICATION TAG field; and
c) does not define the content of the LOGICAL BLOCK REFERENCE TAG field.

If type 3 protection is enabled, then the following media commands are invalid and shall be terminated by the device server with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE:

a) READ (32);
b) VERIFY (32);
c) WRITE (32);
d) WRITE AND VERIFY (32); and
e) WRITE SAME (32).

For valid medium access commands in which the RDPROTECT field, WRPROTECT field, VRPROTECT field, or ORPROTECT field is set to:

a) zero, the data-in buffer and/or data-out buffer associated with those commands shall consist of logical blocks with only user data; or
b) a non-zero value, the data-in buffer and/or data-out buffer shall consist of logical blocks with both user data and protection information.

1.1.3 Protection information format

Table 1 defines the placement of protection information in a logical block if there is no protection information interval.

Table 1 — User data and protection information format with no protection information interval

Bit Byte	7	6	5	4	3	2	1	0
0	USER DATA							
n - 1								
n	(MSB)	LOGICAL BLOCK GUARD						
n + 1								(LSB)
n + 2	(MSB)	LOGICAL BLOCK APPLICATION TAG						
n + 3								(LSB)
n + 4	(MSB)	LOGICAL BLOCK REFERENCE TAG						
n + 7								(LSB)

Table 2 shows an example of the placement of protection information in a logical block where the protection information interval is less than the length of the logical unit.

Table 2 — Example of user data and protection information format with a protection information interval

Bit Byte	7	6	5	4	3	2	1	0
0	USER DATA (first)							
n - 1								
n	(MSB)	LOGICAL BLOCK GUARD (first)						
n + 1								(LSB)
n + 2	(MSB)	LOGICAL BLOCK APPLICATION TAG (first)						
n + 3								(LSB)
n + 4	(MSB)	LOGICAL BLOCK REFERENCE TAG (first)						
n + 7								(LSB)
n + 8	USER DATA (second)							
m - 1								
m	(MSB)	LOGICAL BLOCK GUARD (second)						
m + 1								(LSB)
m + 2	(MSB)	LOGICAL BLOCK APPLICATION TAG (second)						
m + 3								(LSB)
m + 4	(MSB)	LOGICAL BLOCK REFERENCE TAG (second)						
m + 7								(LSB)
	. . .							
	USER DATA (last for this logical block)							
z - 1								
z	(MSB)	LOGICAL BLOCK GUARD (last for this logical block)						
z + 1								(LSB)
z + 2	(MSB)	LOGICAL BLOCK APPLICATION TAG (last for this logical block)						
z + 3								(LSB)
z + 4	(MSB)	LOGICAL BLOCK REFERENCE TAG (last for this logical block)						
z + 7								(LSB)

The USER DATA field shall contain user data. ~~The~~ Only the contents of the USER DATA field immediately preceding THE LOGICAL BLOCK GUARD field (e.g., the user data between the preceding logical block reference tag, if any, and the current logical block guard) shall be used to generate and check the CRC contained in the LOGICAL BLOCK GUARD field.

The LOGICAL BLOCK GUARD field contains the CRC (see 1.1.4) of the contents of only the USER DATA field immediately preceding THE LOGICAL BLOCK GUARD field.~~.~~

The LOGICAL BLOCK APPLICATION TAG field is set by the application client. If the device server detects a:

a) LOGICAL BLOCK APPLICATION TAG field set to FFFFh and type 1 protection (see 1.1.2.3) or type 2 protection (see 1.1.2.4) is enabled; or
b) LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFF_FFFFh, and type 3 protection (see 1.1.2.5) is enabled,

then the device server disables checking of all protection information for the logical block when reading from the medium. Otherwise, the contents of the logical block application tag are not defined by this standard.

The LOGICAL BLOCK APPLICATION TAG field may be modified by a device server if the ATO bit is set to zero in the Control mode page (see SPC-4). If the ATO bit is set to one in the Control mode page, then the device server shall not modify the LOGICAL BLOCK APPLICATION TAG field.

The contents of the LOGICAL BLOCK APPLICATION TAG field shall not be used to generate or check the CRC contained in the LOGICAL BLOCK GUARD field.

The LOGICAL BLOCK REFERENCE TAG field of the first logical block reference tag in the data-in buffer and/or data-out buffer shall contain the value specified in table 3.

Table 3 — Contents of the first LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer

Protection Type	**Content of the first LOGICAL BLOCK REFERENCE TAG field of the first logical block in the data-in buffer and/or data-out buffer**
Type 1 protection [a] (see 1.1.2.3)	The least significant four bytes of the LBA contained in the LOGICAL BLOCK ADDRESS field of the command.
Type 2 protection (see 1.1.2.4)	The value in the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field of the command.
Type 3 protection (see 1.1.2.5)	Not defined in this standard. However, this field may be modified by the device server if the ATO bit is set to zero in the Control mode page (see SPC-4). If the ATO bit is set to one in the Control mode page, then the device server shall not modify this field.

[a] The PROTECTION INFORMATION INTERVAL field (see 1.1.7.1) shall be set to zero (i.e., the length of the user data between protection information is equal to the logical block length).

Subsequent ~~The~~ LOGICAL BLOCK REFERENCE TAG fields ~~subsequent logical blocks~~ in the data-in buffer and/or data-out buffer shall be set as specified in table 4.

Table 4 — Setting ~~the~~subsequent LOGICAL BLOCK REFERENCE TAG fields ~~of the subsequent logical blocks~~ in the data-in buffer and/or data-out buffer

Protection Type	The content of ~~the~~subsequent LOGICAL BLOCK REFERENCE TAG fields ~~of each subsequent logical block~~ in the data-in buffer and/or data-out buffer
Type 1 protection (see 1.1.2.3) and Type 2 protection (see 1.1.2.4)	The previous logical block reference tag ~~of the previous logical block~~ plus one.
Type 3 protection (see 1.1.2.5)	Not defined in this standard. However, this field may be modified by the device server if the ATO bit is set to zero in the Control mode page (see SPC-4). If the ATO bit is set to one in the Control mode page, then the device server shall not modify this field.

The contents of the LOGICAL BLOCK REFERENCE TAG field shall not be used to generate or check the CRC contained in the LOGICAL BLOCK GUARD field.

1.1.4 Logical block guard

1.1.4.1 Logical block guard overview

The LOGICAL BLOCK GUARD field shall contain a CRC that is generated from the contents of only the USER DATA field immediately preceding THE LOGICAL BLOCK GUARD field.

Table 5 defines the CRC polynomials used to generate the logical block guard from the contents of the USER DATA field.

Table 5 — CRC polynomials

Function	Definition
F(x)	A polynomial representing the transmitted USER DATA field, which is covered by the CRC. For the purposes of the CRC, the coefficient of the highest order term shall be byte zero bit seven of the USER DATA field and the coefficient of the lowest order term shall be bit zero of the last byte of the USER DATA field.
F'(x)	A polynomial representing the received USER DATA field.
G(x)	The generator polynomial: $G(x) = x^{16} + x^{15} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (i.e., G(x) = 18BB7h)
R(x)	The remainder polynomial calculated during CRC generation by the transmitter, representing the transmitted LOGICAL BLOCK GUARD field.
R'(x)	A polynomial representing the received LOGICAL BLOCK GUARD field.
RB(x)	The remainder polynomial calculated during CRC checking by the receiver. RB(x) = 0 indicates no error was detected.
RC(x)	The remainder polynomial calculated during CRC checking by the receiver. RC(x) = 0 indicates no error was detected.
QA(x)	The quotient polynomial calculated during CRC generation by the transmitter. The value of QA(x) is not used.
QB(x)	The quotient polynomial calculated during CRC checking by the receiver. The value of QB(x) is not used.
QC(x)	The quotient polynomial calculated during CRC checking by the receiver. The value of QC(x) is not used.
M(x)	A polynomial representing the transmitted USER DATA field followed by the transmitted LOGICAL BLOCK GUARD field.
M'(x)	A polynomial representing the received USER DATA field followed by the received LOGICAL BLOCK GUARD field.

1.1.4.2 CRC generation

The equations that are used to generate the CRC from F(x) are as follows. All arithmetic is modulo 2.

The transmitter shall calculate the CRC by appending 16 zeros to F(x) and dividing by G(x) to obtain the remainder R(x):

$$\frac{(x^{16} \times F(x))}{G(x)} = QA(x) + \frac{R(x)}{G(x)}$$

R(x) is the CRC value, and is transmitted in the LOGICAL BLOCK GUARD field.

M(x) is the polynomial representing the USER DATA field followed by the LOGICAL BLOCK GUARD field (i.e., F(x) followed by R(x)):

$$M(x) = (x^{16} \times F(x)) + R(x)$$

1.1.4.3 CRC checking

M’(x) (i.e., the polynomial representing the received USER DATA field followed by the received LOGICAL BLOCK GUARD field) may differ from M(x) (i.e., the polynomial representing the transmitted USER DATA field followed by the transmitted LOGICAL BLOCK GUARD field) if there are transmission errors.

The receiver may check M’(x) validity by appending 16 zeros to F’(x) and dividing by G(x) and comparing the calculated remainder RB(x) to the received CRC value R’(x):

$$\frac{(x^{16} \times F'(x))}{G(x)} = QB(x) + \frac{RB(x)}{G(x)}$$

In the absence of errors in F’(x) and R’(x), the remainder RB(x) is equal to R’(x).

The receiver may check M’(x) validity by dividing M’(x) by G(x) and comparing the calculated remainder RC(x) to zero:

$$\frac{M'(x)}{G(x)} = QC(x) + \frac{RC(x)}{G(x)}$$

In the absence of errors in F’(x) and R’(x), the remainder RC(x) is equal to zero.

Both methods of checking M’(x) validity are mathematically equivalent.

1.1.4.4 CRC test cases

Several CRC test cases are shown in table 6.

Table 6 — CRC test cases

Pattern	CRC
32 bytes each set to 00h	0000h
32 bytes each set to FFh	A293h
32 bytes of an incrementing pattern from 00h to 1Fh	0224h
2 bytes each set to FFh followed by 30 bytes set to 00h	21B8h
32 bytes of a decrementing pattern from FFh to E0h	A0B7h

1.1.5 Application of protection information

Before an application client transmits or receives logical blocks with protection information it shall:

1) determine if a logical unit supports protection information using the INQUIRY command (see the PROTECT bit in the standard INQUIRY data in SPC-4);
2) if protection information is supported, then determine if the logical unit is formatted to accept protection information using the READ CAPACITY (16) command (see the PROT_EN bit in 5.13); and
3) if the logical unit supports protection information and is not formatted to accept protection information, then format the logical unit with protection information enabled.

If the logical unit supports protection information and is formatted to accept protection information, then the application client may use commands performing read operations that support protection information and should use commands performing write and verify operations that support protection information.

1.1.6 FORMAT UNIT command overview

...

1.1.7 FORMAT UNIT parameter list

1.1.7.1 FORMAT UNIT parameter list overview

;;;

1.1.7.2 Parameter list header

The parameter list headers (see table 7 and table 8) provide several optional format control parameters. Device servers that implement these headers provide the application client additional control over the use of the four defect sources, and the format operation. If the application client attempts to select any function or value not implemented by the device server, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

If the LONGLIST bit is set to zero in the FORMAT UNIT CDB, then the short parameter list header (see table 7) is used.

Table 7 — Short parameter list header

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved					PROTECTION FIELD USAGE		
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor-specific
2	(MSB)	DEFECT LIST LENGTH						
3								(LSB)

If the LONGLIST bit is set to one in the FORMAT UNIT CDB, then the long parameter list header (see table 8) is used.

Table 8 — Long parameter list header

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved					PROTECTION FIELD USAGE		
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor-specific
2	Reserved							
3	~~Reserved~~PROTECTION INFORMATION DISTRIBUTION							
4	(MSB)	DEFECT LIST LENGTH						
7								(LSB)

The PROTECTION FIELD USAGE field in combination with the FMTPINFO field (see table 9) specifies the requested protection type (see 4.17.2).

Table 9 — FMTPINFO field and PROTECTION FIELD USAGE field (part 1 of 2)

Device server indication		Application client specification		Description
SPT [a]	PROTECT [b]	FMTPINFO	PROTECTION FIELD USAGE	
xxxb	0	00b	000b	The logical unit shall be formatted to type 0 protection [c] (see 4.17.2.2) resulting in the P_TYPE field [d] being set to 000b.
xxxb	0	00b	>000b	Illegal [e]
xxxb	0	01b	xxxb	Illegal [f]
xxxb	0	1xb	xxxb	Illegal [f]
xxxb	1	00b	000b	The logical unit shall be formatted to type 0 protection [c] (see 4.17.2.2) resulting in the P_TYPE field [d] being set to 000b.
xxxb	1	00b	>000b	Illegal [e]
xxxb	1	01b	xxxb	Illegal [f]
000b 001b 011b	1	10b	000b	The logical unit shall be formatted to type 1 protection [g] (see 4.17.2.3) resulting in the P_TYPE field [d] being set to 000b.
000b 001b 011b	1	10b	>000b	Illegal [e]
000b	1	11b	xxxb	Illegal [f]
001b	1	11b	000b	The logical unit shall be formatted to type 2 protection [g] (see 4.17.2.4) resulting in the P_TYPE field [d] being set to 001b.
001b	1	11b	>000b	Illegal [e]
011b	1	11b	000b	Illegal [e]

[a] See the Extended INQUIRY Data VPD page (see SPC-4) for the definition of the SPT field.
[b] See the standard INQUIRY data (see SPC-4) for the definition of the PROTECT bit.
[c] The device server shall format the medium to the logical block length specified in the mode parameter block descriptor of the mode parameter header (see SPC-4).
[d] See the READ CAPACITY command (see 1.2.1) for the definition of the P_TYPE field.
[e] The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
[f] The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
[g] The device server shall format the medium to the logical block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the logical block length is 512, then the formatted logical block length is 520). Following a successful format, the PROT_EN bit in the READ CAPACITY (16) parameter data (see 1.2.1) indicates whether protection information (see 4.17) is enabled.

Table 9 — FMTPINFO field and PROTECTION FIELD USAGE field (part 2 of 2)

Device server indication		Application client specification		Description
SPT [a]	PROTECT [b]	FMTPINFO	PROTECTION FIELD USAGE	
011b	1	1	001b	The logical unit shall be formatted to type 3 protection. [g] (see 4.17.2.5) resulting in the P_TYPE field [d] being set to 010b.
011b	1	11b	>001b	Illegal [e]
010b	1	1xb	xxxb	Reserved
1xxb	1	1xb	xxxb	Reserved

[a] See the Extended INQUIRY Data VPD page (see SPC-4) for the definition of the SPT field.

[b] See the standard INQUIRY data (see SPC-4) for the definition of the PROTECT bit.

[c] The device server shall format the medium to the logical block length specified in the mode parameter block descriptor of the mode parameter header (see SPC-4).

[d] See the READ CAPACITY command (see 1.2.1) for the definition of the P_TYPE field.

[e] The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

[f] The device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

[g] The device server shall format the medium to the logical block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the logical block length is 512, then the formatted logical block length is 520). Following a successful format, the PROT_EN bit in the READ CAPACITY (16) parameter data (see 1.2.1) indicates whether protection information (see 4.17) is enabled.

....

The PROTECTION INFORMATION DISTRIBUTION field specifics the distribution of protection information based on the requested protection type (see table 9) as follows:

a) for a type 2 protection format request this field shall be as specified in table 10; or
b) for a type 3 protected format request this field shall be as specified in table 10.

If type 0 protection format or type 1 protection format is requested and the protection information distribution field is not set to zero, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 10 — PROTECTION INFORMATION DISTRIBUTION field

Byte \ Bit	7	6	5	4	3	2	1	0
3	0	0	0	0	PROTECTION INFORMATION INTERVAL			

The PROTECTION INFORMATION INTERVAL field specifies the number of protection information intervals to be placed within each logical block.

The length of user data between protection information is calculated as follows:

$$\text{length of user data} = \text{logical block length} / 2^{(\text{protection information interval})}$$

where:

logical block length   is the length in bytes of a logical block protection information interval   is the contents of the PROTECTION INFORMATION INTERVAL field

If the length of user data calculates to a value that is not even (e.g., 520/2^{3} = 65) or not a whole number (e.g., 520/2^{4} = 32.5 and 520/2^{10} = 0.508), then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The protection interval length shall not include the 8 byte protection information (e.g., a logical unit being formatted with 2048 byte logical blocks with the PROTECTION INFORMATION INTERVAL field set to 2 contains 8 bytes of protection information after each 512 bytes (i.e., 2048 / 2^{2}) of user data for a total length of 2080 bytes).

1.2 READ CAPACITY (16) command

1.2.1 READ CAPACITY (16) command overview

...

1.2.2 READ CAPACITY (16) parameter data

The READ CAPACITY (16) parameter data is defined in table 11. Any time the READ CAPACITY (16) parameter data changes, the device server should establish a unit attention condition as described in 4.7.

Table 11 — READ CAPACITY (16) parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	RETURNED LOGICAL BLOCK ADDRESS						
7								(LSB)
8	(MSB)	LOGICAL BLOCK LENGTH IN BYTES						
11								(LSB)
12	Reserved				P_TYPE			PROT_EN
13	~~Reserved~~P_I_INTERVAL				LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT			
14	Reserved		(MSB)	LOWEST ALIGNED LOGICAL BLOCK ADDRESS				
15								(LSB)
16	Reserved							
31								

The RETURNED LOGICAL BLOCK ADDRESS field and LOGICAL BLOCK LENGTH IN BYTES field of the READ CAPACITY (16) parameter data are the same as the in the READ CAPACITY (10) parameter data (see 5.12). The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFF_FFFF_FFFF_FFFEh.

The protection type (P_TYPE) field and the protection enable (PROT_EN) bit (see table 12) indicate the logical unit's current type of protection.

Table 12 — P_TYPE field and PROT_EN bit

PROT_EN	P_TYPE	Description
0	xxxb	The logical unit is formatted to type 0 protection (see 4.17.2.2).
1	000b	The logical unit is formatted to type 1 protection (see 4.17.2.3).
1	001b	The logical unit is formatted to type 2 protection (see 4.17.2.4).
1	010b	The logical unit is formatted to type 3 protection (see 4.17.2.5).
1	011b to 111b	Reserved

The LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field is defined in table 13.

Table 13 — LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field

Code	Description
0	One or more physical blocks per logical block [a]
n > 0	2^n logical blocks per physical block

[a] The number of physical blocks per logical block is not reported.

The P_I_INTERVAL field indicates the number of protection information intervals placed within each logical block (see 1.1.7).

The LOWEST ALIGNED LOGICAL BLOCK ADDRESS field indicates the LBA of the first logical block that is located at the beginning of a physical block (see 4.5).

NOTE 1 - The highest LBA that the lowest aligned logical block address field supports is 3FFFh (i.e., 16 383).