To: T10 Technical Committee
From: Jim Hatfield, Seagate (James.C.Hatfield@seagate.com),
Rob Elliott, HP (elliott@hp.com)
Date: June 12, 2007
Subject: SAT-2 Translation of Large Physical Blocks

Revision History

Revision 0 (May 3, 2007)	First revision
Revision 1 (June 12, 2007)	Comments made during May SAT meeting
Revision 2 (July 10, 2007)	Comments made during July SAT meeting: approved for SAT-2

Related Documents

sat-r09 - SCSI/ATA Translation (SAT) revision 9
SAT-2r01 - SCSI/ATA Translation (SAT-2) revision 1
sbc3r09 - SCSI Block Commands - 3 (SBC-3) revision 9
ata8-acs-r4a - AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS) revision 4a

Overview

ATA8-ACS defines how an ATA device reports its geometry of logical and physical sectors. SBC-3 defines how a SCSI device server reports similar information. This proposal describes how to map these geometry parameters between ATA and SCSI.

Some additional text in SAT is cleaned-up.

Suggested Changes

3.1 Definitions

[Editor's note: add these new definitions]

3.1.XX ATA logical sector alignment: The logical sector offset within the first physical sector where the first logical sector is placed in an ATA device. If ATA IDENTIFY DEVICE data word 209 ('alignment of logical blocks within physical blocks'), bit 15) is set to zero and bit 14 is set to one, this is returned in ATA IDENTIFY DEVICE data word 209, bits 13:0; otherwise, it is zero. See 9.9.

3.1.YY ATA logical sectors per physical sector exponent: The power of two exponent of the number of logical sectors per physical sector in an ATA device. If ATA IDENTIFY DEVICE word 106 ('physical size/logical sector size') bit 15 is set to zero, bit 14 is set to one, and bit 13 is set to one, this is returned in ATA IDENTIFY DEVICE data word (106) bits 3:0; otherwise, it is zero. See 9.9.

3.1.ZZ ATA logical sectors per physical sector: The number of logical sectors per physical sector in an ATA device (i.e., $2^{(\text{ATA logical sectors per physical sector exponent})}$). See 9.9.

9.8 READ CAPACITY (10) command

9.8.1 READ CAPACITY (10) command overview

The READ CAPACITY (10) command (see SBC-3) requests that the device server transfer eight bytes of parameter data describing the capacity and medium format of the direct-access block device to the application client. Table 34 shows the translation for fields specified in the READ CAPACITY (10) CDB.

Table 34 — READ CAPACITY (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 25h. The SATL shall use ATA IDENTIFY DEVICE data to compute the ATA device's maximum user addressable medium capacity of the ATA device.
LOGICAL BLOCK ADDRESS	If the LOGICAL BLOCK ADDRESS field is not set to zero, the SATL 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.
PMI	If the PMI bit is not set to zero, the SATL 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.
CONTROL	6.4

9.8.2 READ CAPACITY (10) parameter data

The SATL shall return READ CAPACITY (10) parameter data as defined by SBC-3. Table 35 describes the translation of fields in the READ CAPACITY (10) parameter data.

Table 35 - READ CAPACITY(10) parameter data

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS [a]	If the SATL implements direct logical block mapping (see 3.1.33), this field shall contain the lower of: a) the ATA maximum LBA (see 3.1.16); and b) FFFF_FFFFh. If the value is FFFF_FFFFh, the ~~If the number of logical blocks exceeds the maximum value that is able to be specified in the RETURNED LOGICAL BLOCK ADDRESS field, the device server shall set the RETURNED LOGICAL BLOCK ADDRESS field to FFFFFFFFh. The~~ application client should ~~then~~ issue a READ CAPACITY (16) command (see 9.9) to retrieve the READ CAPACITY (16) parameter data. If the SATL implements indirect logical block mapping, this field is ~~This field is otherwise~~ unspecified (see 3.4.2).
LOGICAL BLOCK LENGTH IN BYTES [a]	If the SATL implements direct logical block mapping (see 3.1.33) then ~~the BLOCK LENGTH IN BYTES~~ this field shall contain the ATA logical sector size (see 3.1.15). Otherwise, ~~the BLOCK LENGTH IN BYTES~~ this field is unspecified (see 3.4.2).
a The values reported in the RETURNED LOGICAL BLOCK ADDRESS field and the LOGICAL BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.49) is less than or equal to the ATA device capacity (see 3.1.9).	

9.9 READ CAPACITY(16) command

9.9.1 READ CAPACITY (16) command overview

The READ CAPACITY (16) command (see SBC-3) requests that the device server transfer ~~eight bytes of~~ parameter data describing the capacity and medium format of the direct-access block device to the application client. Table 36 shows the translation for fields specified in the READ CAPACITY (16) CDB.

Table 36 — READ CAPACITY (16) CDB field translations

Field	Description or reference
OPERATION CODE/SERVICE ACTION	Set to 9Eh/10h. ~~The SATL shall use ATA IDENTIFY DEVICE data to compute the ATA device's maximum user addressable medium capacity of the ATA device.~~
LOGICAL BLOCK ADDRESS	~~If the LOGICAL BLOCK ADDRESS field is not set to zero the SATL 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.~~ As defined in READ CAPACITY (10) (see 9.8)
PMI	~~If the PMI bit is not set to zero the SATL 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.~~ As defined in READ CAPACITY (10) (see 9.8)
ALLOCATION LENGTH	Unspecified (see 3.4.2)
CONTROL	6.4

9.9.2 READ CAPACITY(16) parameter data

The SATL shall return READ CAPACITY(16) parameter data as defined by SBC-3. describes the translation of fields in the READ CAPACITY (16) parameter data.

Table 37 - READ CAPACITY(16) parameter data

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS [A]	If the SATL implements direct logical block mapping (see 3.1.33), this field shall contain the ATA maximum LBA (see 3.1.16). If the SATL implements indirect logical block mapping, this field is unspecified (see 3.4.2). ~~The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFFFFFF_FFFFFFFEh. This field is otherwise unspecified (see 3.4.2).~~
LOGICAL BLOCK LENGTH IN BYTES [A]	~~If the SATL implements direct logical block mapping (see 3.1.33) then the LOGICAL BLOCK LENGTH IN BYTES field shall contain the ATA logical sector size (see 3.1.15). Otherwise, the BLOCK LENGTH IN BYTES this field is unspecified (see 3.4.2)~~. As defined in READ CAPACITY (10) (see 9.8).
~~RTO_EN~~	~~SBC-2: Unspecified (see 3.4.2)~~
PROT_EN	Unspecified (see 3.4.2)
P_TYPE	Unspecified (see 3.4.2)
LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT	If the SATL implements direct logical block mapping (see 3.1.33) then this field shall contain the ATA logical sectors per physical sector exponent (see 3.1.YY). If the SATL implements indirect logical block mapping, this field is unspecified (see 3.4.2).
LOWEST ALIGNED LOGICAL BLOCK ADDRESS	If the SATL implements direct logical block mapping (see 3.1.33) then a) if the ATA logical sector alignment is zero, then this field shall contain zero; b) if the ATA logical sector alignment is not zero, then this field shall contain (ATA logical sectors per physical sector) - (ATA logical sector alignment) If the SATL implements indirect logical block mapping, this field is unspecified (see 3.4.2).

[a] The values reported in the RETURNED LOGICAL BLOCK ADDRESS field and the LOGICAL BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.49) is less than or equal to the ATA device capacity (see 3.1.9).

Figure 1 shows examples of LOWEST ALIGNED LOGICAL BLOCK ADDRESS field values for various values of ATA logical sector alignment and ATA logical sectors per physical sector.

Figure 1- LOWEST ALIGNED LOGICAL BLOCK ADDRESS field examples

ATA logical sectors per physical sector=1
ATA logical sector alignment=0
LOWEST ALIGNED LOGICAL BLOCK ADDRESS field=0

LBA 0	LBA 1	LBA 2	LBA 3	LBA 4	LBA 5	LBA 6	LBA 7	LBA 8	…
PB	PB	PB	PB	PB	PB	PB	PB	PB	…

ATA logical sectors per physical sector=2
ATA logical sector alignment=1
LOWEST ALIGNED LOGICAL BLOCK ADDRESS field=1

	LBA 0	LBA 1	LBA 2	LBA 3	LBA 4	LBA 5	LBA 6	LBA 7	…
PB		PB		PB		PB		…	

ATA logical sectors per physical sector=8
ATA logical sector alignment=1
LOWEST ALIGNED LOGICAL BLOCK ADDRESS field =7

	LBA 0	LBA 1	LBA 2	LBA 3	LBA 4	LBA 5	LBA 6	LBA 7	LBA 8	LBA 9	LBA 10	LBA 11	LBA 12	LBA 13	LBA 14	LBA 15	...
PB								PB								…	

ATA logical sectors per physical sector=8
ATA logical sector alignment=7
LOWEST ALIGNED LOGICAL BLOCK ADDRESS field =1

							LBA 0	LBA 1	LBA 2	LBA 3	LBA 4	LBA 5	LBA 6	LBA 7	LBA 8	LBA 9	…
PB								PB								…	

Key:
LBA = logical block address
PB = physical block

Editor's note: Figure must be converted into Visio. Get a reformatted version of this figure from Ralph Weber]