To: T10 Membership
From: Paul Suhler, Seagate Technology
Date: 8 February 2000
Subject: Support for Large Block Addresses in SSC, T10/00-135r0

Seagate
THE DATA TECHNOLOGY COMPANY

1. Background

There now exist tape drives whose media are large enough that they could hold more blocks than could be addressed by existing SSC commands. For example, first-generation LTO drives have cartridge capacities of 200 GB, and the LTO roadmap has 1.6 TB media in generation four. Even with only 200 GB, a tape with small block sizes could hold more blocks than the existing LOCATE command can address with its four-byte BLOCK ADDRESS field. A 1.6 TB medium would max out the logical block address with block sizes of less than 400 bytes.

While workarounds will be possible without command set changes, we should use this opportunity to address the command set issues before we have to invent vendor-unique solutions which will become legacy support problems.

The January 2000 working group ranked general solutions by desirability:

1. New 16-byte commands
2. New service actions under the variable length CDB
3. New 12-byte commands
4. A new bit in the current commands

This proposal addresses the first solution, sixteen-byte commands, without considering the availability of op codes for the commands. This is contrary to the direction taken by the SCSI Protocol Working Group for SBC-2 commands, which supports large addresses via variable-length CDBs (T10/00-125r0).

2. Assumptions

The following assumptions were derived by looking at the command and data fields for SSC commands to see where fields may overflow. Assumptions 1 through 4 argue for changes in block address fields to eight bytes and transfer length fields to four bytes; the rest argue for no changes in other fields. These assumptions should be considered carefully and discussed.

1. Number of blocks on a medium may be more than 2^{32}.
2. Absolute block addressing may specify more than 2^{32} blocks.
3. Relative block addressing may specify more than 2^{32} blocks.
4. Read and write transfer lengths may be more than 2^{24} bytes, but will be less than 2^{32}.
5. Number of blocks in a device buffer will be less than 2^{24}.

6. Number of bytes in a device buffer will be less than 2^{24}.
7. Number of setmarks or filemarks in a partition will be less than 2^{32}.
8. Number of partitions on a medium will be less than 2^{8}.
9. Number of blocks in a file will be less than 2^{24}.

Assumption 3 justifies changing the SPACE command. Without this change, a workaround would be to issue multiple SPACE commands, since the block count is relative to the current position. However, on some media formats, such as serpentine, this could mean changing directions between commands, which would be more time-consuming. The larger block count would allow the device to move the medium directly to the desired location, without the intermediate stops.

Assumption 4 justifies changing the READ, READ REVERSE, and WRITE commands. A transfer of more than 16 MBytes would mean that a device would be on the transport medium for a long time, possibly leading to starvation problems. If the current 24-bit transfer length fields are deemed sufficient, then these commands need not change.

3. Command Changes

3.1 Summary

Command Name	Operation Code	Type	SSC Clause	Comment
ERASE	19h	M	5.3.1	Not affected
FORMAT MEDIUM	04h	O	5.3.2	Not affected
LOAD UNLOAD	1Bh	O	5.3.3	Not affected
LOCATE	2Bh	O	5.3.4	Use CDB format from 3.2 for Logical Block Address
READ	08h	M	5.3.5	Use CDB format from 3.3 for Count.
READ BLOCK LIMITS	05h	M	5.3.6	Not affected
READ POSITION	34h	M	5.3.7	If medium contains at least 2^{32} blocks and if LONG bit in CDB = 0, reject command with CHECK CONDITION status and sense key of ILLEGAL REQUEST.
READ REVERSE	0Fh	O	5.3.8	Use CDB format from 3.5 for Transfer Length.
RECOVER BUFFERED DATA	14h	O	5.3.9	Not affected
REPORT DENSITY SUPPORT	44h	M	5.3.10	CDB is not affected. The density support data block descriptor supports capacities up to 4 petabytes (4×10^{15}), and thus need not change.
REWIND	01h	M	5.3.11	Not affected
SPACE	11h	M	5.3.12	Use CDB format from 3.7 for Count
VERIFY	13h	O	5.3.13	Not affected
WRITE	0Ah	M	5.3.14	Use CDB from 3.7 for Transfer Length
WRITE FILEMARKS	10h	M	5.3.15	Not affected

3.2 LOCATE (16) Command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code (TBD)							
1	Reserved					BT	CP	IMMED
2	(MSB)							
3								
4								
5								
6				Logical Block Address				
7								
8								
9								(LSB)
10	Reserved							
11	Reserved							
12	Reserved							
13	Reserved							
14	Partition							
15	Control							

3.3 READ (16) Command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (TBD)							
1	Reserved						FIXED	SILI
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	(MSB)							
11				TRANSFER LENGTH				
12								
13								(LSB)
14	Reserved							
15	Control							

3.4 READ POSITION Command

The difficulty with the READ POSITION command is that the short form of the data uses four-byte fields for FIRST BLOCK LOCATION and LAST BLOCK LOCATION. (The long form of the data has an eight-byte field for BLOCK NUMBER, and thus requires no change.) There are a couple of options:

1. When the current medium contains 2^{32} or more blocks and the LONG bit is zero, reject the command with CHECK CONDITION status and sense key of ILLEGAL REQUEST and ASC/ASCQ of PARAMETER VALUE NOT SUPPORTED.
2. Reject the command as above, regardless the number of blocks on the medium.
3. Invent a new "short form" of data that eight-byte fields for FIRST and LAST BLOCK LOCATION. As this wouldn't be compatible with existing applications that expect the current short form, this would be at least as painful as allowing only the long form.

3.5 READ REVERSE (16) Command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (TBD)							
1	Reserved					BYTORD	FIXED	SILI
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	(MSB)			TRANSFER LENGTH				
11								
12								
13								(LSB)
14	Reserved							
15	Control							

3.6 SPACE (16) Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (TBD)							
1	Reserved						CODE	
2	(MSB)							
3								
4								
5								
6				COUNT				
7								
8								
9								(LSB)
10	Reserved							
11	Reserved							
12	Reserved							
13	Reserved							
14	Reserved							
15	Control							

3.7 WRITE (16) Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (TBD)							
1	Reserved							FIXED
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	(MSB)							
11				TRANSFER LENGTH				
12								
13								(LSB)
14	Reserved							
15	Control							