

CONGRUENT SOFTWARE, INC.

514 48th AVENUE

SAN FRANCISCO, CALIFORNIA 94121

X3 T9.2/87-151

TO: John Lohmeyer
Chairman, X3T9.2

FROM: Peter Johansson

DATE: August 28, 1987

RE: Management of the "Grown" Defect List

=====
After reviewing the Direct-Access Device sections on the FORMAT UNIT, READ DEFECT DATA and REASSIGN BLOCKS commands, I believe the definitions need enhancement and clarification with respect to management of the "Grown" defect list.

First, the READ DEFECT DATA command. Just what does it mean to return a list of logical block addresses for defect locations if they are remapped by the target? The third sentence of the paragraph referenced below obliquely addresses the problem, but because defect locations that have been remapped have no logical block addresses, I believe it is not strong enough. I suggest that the fifth new paragraph on page 8-65, Section 8.1.11 of Revision 2 of the working draft be altered as follows:

285
The Defect List Format specifies the format of the defect list data being returned by the target. Legal values for this field are defined in the FORMAT UNIT command (see section 8.1.2, Defect List Format). The logical block and physical sector formats may not include defects in user inaccessible areas. If the logical block format is specified, only those defect locations that have not been remapped by the target shall be returned. If either the bytes from index or physical sector formats are specified, the complete list(s) of defects shall be returned, however, some of the defects may not lie in user accessible areas (depending on the disk geometry used by the most recent FORMAT UNIT command). If the value in this the Defect List Format field does not match the defect list format requested by the initiator in the CDB, the target shall transfer the defect data and return a CHECK CONDITION status with the sense key set to RECOVERED ERROR and an additional sense code of DEFECT LIST NOT AVAILABLE.

This ability to distinguish between defects that are recorded in a target's Primary or "Grown" lists and defects that are both recorded and remapped is useful in conjunction with REASSIGN BLOCKS as described below.

Secondly, the REASSIGN BLOCKS command description makes no mention of the fact that the "Grown" defect list is updated when the command is executed. This fact is buried back in the FORMAT UNIT command description. This command provides the only means of adding to the "Grown" defect list other than at the time the unit is formatted, and because of that should have the added capability to add defect(s) to the "Grown" list without remapping them. Three situations come to mind where this is valuable: 1) A volume in use for some time has exhausted its spare sectors and this fact is revealed by a CHECK CONDITION after a REASSIGN BLOCKS command. The initiator will need to reformat the volume with more spare capacity; rather than go through a tortuous logical block address calculation for the new "virtual" disk geometry and the "physical" logical block addresses (now rather confusedly described under the FORMAT UNIT command) it would be more direct for the initiator to issue a REASSIGN BLOCKS command for the affected blocks (specifying no remapping) followed by a FORMAT UNIT command that retains the "Grown" defect list. 2) The initiator does not remap any defects, performs its own defect management schemes but wishes to utilize the target's capability to record defect lists for future reference. 3) An initiator performs and incoming inspection test of the target with all defect management and remapping disabled and discovers "Grown" defects. These are to be added to the defect list when the target is reformatted with defect remapping enabled; once again, it would be more direct to use the REASSIGN BLOCKS command rather than a calculation of "physical" logical block addresses.

There are two ways to modify REASSIGN BLOCKS to add new functionality: a) add a bit in the CDB that specifies that the block(s) are to be recorded in the "Grown" list, only, and not reassigned or b) change the definition of the actions performed after the NO DEFECT SPARE LOCATION AVAILABLE error condition such that the defect location being processed is added to the "Grown" list. The first approach is cleaner and is reflected in the proposed wording for the REASSIGN BLOCKS command attached to this memorandum.

In informal discussions at the Colorado Springs meeting about the REASSIGN BLOCKS command, the objection was raised that all defects recorded in the "Grown" defect list at present are reassigned and that if REASSIGN BLOCKS were to be modified as described above the initiator would not be able to distinguish reassigned from unreassigned defects. The changes to READ DEFECT DATA (which I believe are needed on their own merits), along with the newly added address translation functions of SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS answer this objection.

Lastly, there is the management of the "Grown" defect list through the FORMAT UNIT command. The use of the logical block format when the "virtual" disk geometry is changing at the time of a FORMAT UNIT, because of changes to any or all of spare sector allocation, disk geometry, physical sector size of logical block size, is problematical. The second and third paragraphs on

page 8-5 in Section 8.1.2 attempt to address this problem, but the definition of "addresses that the logical blocks would occupy if there were no defects" is not clear, and, for some devices, might not even be ascertainable by the initiator. It is much better to let the target manage its internal defect lists properly. I propose that logical block address format always refer to the previous geometry, spare allocation, sector and logical block sizes of the target and the target perform the necessary computations to remap the defect information into the new geometry of the target. If there is no previous geometry, etc. the logical block format should be disallowed and a CHECK CONDITION of ILLEGAL REQUEST returned.

Thank you for your consideration of these interrelated problems. There may be simpler and more elegant solutions than the ones proposed; I would welcome any comments.

286

8.1.13. REASSIGN BLOCKS Command

Peripheral Device Type: Direct-Access and Write-Once
 Read Multiple
 Operation Code Type: Optional

Table 8-39: REASSIGN BLOCKS COMMAND

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code (07h)							
1	Logical Unit Number						NoMap	
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Control Byte							

The REASSIGN BLOCKS command (Table 8-32) requests the target to reassign the defective logical blocks to an area on the logical unit reserved for this purpose and to record the defective logical blocks to the "Grown" defect list. More than one physical or logical block may be relocated by each defect descriptor sent by the initiator. This command does not alter the contents or location of the Primary defect list (see section 8.1.2, FORMAT UNIT command).

If the suppress reassignment (NoMap) bit is one, the target shall enter the defect(s) in the "Grown" defect list but not attempt to reassign references to the location(s) specified in the defect list.

The initiator transfers... (the remainder of the REASSIGN BLOCKS command description is unchanged).