Date: July 9, 1996
To: X3T10 Committee
From: Gerry Houlder, Seagate Technology
Subj: Add Immediate bit to XPWRITE and XDWRITE Extended commands

The need for this feature was identified by Chris Burns (Maximum Strategy) in reflector messages and follow up phone calls with me. He is concerned about poor performance in situations where several (and particularly if all) data drives that have the same parity drive need to be updated at the same time. The parity drive will be a bottleneck in this situation.

As an example, consider a 4 drives plus parity situation. If all 4 drives need to be written using XDWRITE command the command sequence would be as follows:

(1) An XDWRITE command is issued to a data drive.
(2) An XDREAD command is issued to return the xor data to the initiator.
(3) An XPWRITE command is issued to write the xor data to the parity drive.
(4) Steps 1 through 3 are repeated for each of the data drives.

This sequence results in 2 commands (one XDWRITE and one XDREAD) being issued to each data drive and 4 XPWRITE commands to the parity drive. If each XPWRITE command has to write the xor result to the drive before doing the next XPWRITE command, at least one extra disk revolution will be lost on each XPWRITE command. Performance would be better if the first 3 XPWRITE commands left the resulting parity data in cache and returned GOOD status without writing the data to disk. The last XPWRITE would write the data to disk when it is completed.

Chris suggests using a bit in the command block to indicate that the data shouldn’t be written to disk yet. If the bit is one, the XPWRITE command returns GOOD status as soon as the xor operation is complete and doesn’t attempt to write the data to disk. If the bit is zero, then GOOD status cannot be returned until the data has been written to disk. This conforms to the existing requirements on this command.

This feature must also work with the XDWRITE Extended command. A system that uses XDWRITE Extended would use the following command sequence:

(1) An XDWRITE extended command is issued to a data drive. The data drive sends XPWRITE command to parity drive. When XPWRITE returns status, data drive returns status to the controller.
(2) Controller repeats step 1 for each of the data drives.

In order to make use of the “XPWRITE immediate” feature, a bit must also be added to the XDWRITE Extended command. That way the bit can be carried over to the XPWRITE command that is issued by the data drive. This would be used in the same way: the first 3 XDWRITE Extended commands would set the “immediate” bit and the last command would have the bit cleared so it would cause the data to be written to disk.

There is an alternative that must also be discussed. We could generalize the use of the WCE bit in Mode Page 8 so that it applies to XPWRITE commands as well as regular write commands. This would be a reasonable extension because the xor data that is left in cache after completion of the xor operation is the same as the data written to disk. Therefore it is safe to let it be used to satisfy a read request for that LBA (as a cache hit) and otherwise has the same retention and safety requirements as regular write data.

An advantage for the WCE bit alternative is that the controller wouldn’t have to be concerned with which of the xor commands executes first or last. The disadvantage is that there is no assurance that the parity drive won’t try to write the data to disk before the last command is received. There is also no assurance the data will be written to disk as soon as the last command has completed. RAID controllers like to know (and exert control over) exactly when data is written to disk. That is why Chris Burns prefers adding a bit to the xor commands -- it provides explicit control over the write operation.

The WCE bit option should only be persued if the RAID controller companies feel comfortable with it. Even if we decide that the XPWRITE command should make use of the WCE bit, some changes to the standard will probably be needed. The model for xor commands will need to be updated to describe how write caching can be used to help certain xor operations and cannot be used for others.