T10/99-166r2

To: T10 Membership
From: Lawrence J. Lamers, John Lohmeyer
Subject: Enabling Upper Byte Hamming
Date: Wednesday, March 31, 1999

This proposal presents a method for enabling the upper byte hamming code protection advocated by Compaq.

The upper byte hamming code protection is likened to parity in its purpose and function. Parity is currently not negotiated. Rather than negotiate the enabling it seems possible for the initiator and target to determine if the upper byte hamming code is being generated by a simple test during the first I/O process following a power on or reset condition.

This method is reliable because the occurrence of an error is extremely small; and the likelihood of an error occurring on several bytes is infinitesimal.

1) All devices supporting the upper byte hamming code protection shall generate the code on all COMMAND, MESSAGE, and STATUS phases.
2) If upper byte hamming code protection detection is enabled, then the device shall check the upper byte.
3) If the upper byte hamming code is detected during a message or command received by the target it shall use the upper byte hamming code protection detection on that I_T nexus for subsequent I/O processes until a power on or reset condition occurs.
4) If the upper byte hamming code protection is received by the initiator on the status and command completion message it shall use the upper byte hamming code protection detection on that I_T nexus for subsequent I/O processes until a power on or reset condition occurs.
5) If a device receives two bytes with good parity but an error on the upper byte hamming code the device should disable upper byte hamming code checking for that I_T nexus. (This is to handle hot-plug situations where a device supporting upper byte hamming code checking is replaced by one that does not.)

Note: The exception handling for hot plug events probably needs more enumeration and discussion. A good implementor’s note is needed. The most likely scenario is that detection of a hamming code error will result in the host aborting the outstanding I_O process for that nexus and re-iniitializing the connection. The considerations for rebuild operations on RAID implementations is beyond the scope of this standard.