Release Notes

Rev F1 • July 2017

Optimus.2 Eco/Ascend/Ultra/Extreme 2.5" SAS, SSD Product Line Generic

SanDisk® a Western Digital brand

Western Digital Technologies, Inc.
951 SanDisk Drive
Milpitas, CA 95035

www.SanDisk.com

Legal Disclaimer

The Western Digital Corporation or its affiliate's general policy does not recommend the use of its products in life support applications wherein a failure or malfunction of the product may directly threaten life or injury. Without limitation to the foregoing, SanDisk shall not be liable for any loss, injury, or damage caused by use of its products in any of the following applications:

- Special applications such as military related equipment, nuclear reactor control, and aerospace.
- Control devices for transportation equipment including automotive vehicles, trains, ships, and traffic equipment.
- Safety systems for disaster prevention and crime prevention.
- Medical-related equipment (including medical measurement devices).

Accordingly, in any use of SanDisk products in life support systems or other applications where failure could cause damage, injury, or loss of life, the products should only be incorporated in systems designed with appropriate redundancy, fault tolerant or back-up features. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support or other such applications assumes all risk of such use and agrees to indemnify, defend, and hold harmless Western Digital Corporation or its affiliates against all damages.

Security safeguards, by their nature, are capable of circumvention. SanDisk cannot, and does not, guarantee that data will not be accessed by unauthorized persons, and SanDisk disclaims any warranties to that effect to the fullest extent permitted by law.

This document and related material are for information use only and are subject to change without prior notice. Western Digital Corporation or its affiliates assumes no responsibility for any errors that may appear in this document or related material, nor for any damages or claims resulting from the furnishing, performance, or use of this document or related material. absent a written agreement signed by Western Digital Corporation or its affiliates or its authorized representative to the contrary, Western Digital Corporation or its affiliates explicitly disclaims any express and implied warranties and indemnities of any kind that may, or could, be associated with this document and related material, and any user of this document or related material agrees to such disclaimer as a precondition to receipt and usage hereof. Each user of this document expressly waives all guaranties and warranties of any kind associated with this document and/or related materials, whether expressed or implied, including without limitation, any implied warranty of merchantability or fitness for a particular purpose or infringement, together with any liability of Western Digital Corporation or its affiliates and its affiliates under any control, negligence, strict liability of Western Digital Corporation or its affiliates and its affiliates under any contract, profit or other incidental, punitive, indirect, special, or consequential damages, including without limitation physical injury or death, property damage, lost data, or costs of procurement of substitute goods, technology, or services.

This document and its contents, including diagrams, schematics, methodology, work product, and intellectual property rights described in, associated with, or implied by this document, are the sole and exclusive property of Western Digital Corporation or its affiliates and its applicable subsidiaries ("SanDisk"). No intellectual property license, express or implied, is granted by SanDisk associated with the document recipient's receipt, access and/or use of this document; SanDisk retains all rights hereto.

No work for hire, nor any form of joint ownership, is granted or implied by the document recipient's receipt, access and/ or use of this document.

Any work requested (or implied by the document recipient to be requested) to SanDisk associated with this document and/or its contents, shall be the sole and exclusive property of SanDisk, except to the extent, if any, expressly agreed otherwise by SanDisk in writing referencing this document.

This document and SanDisk's communications to the user associated therewith, shall be treated as SanDisk's proprietary and confidential information, protected by the recipient as such, and used by the recipient only for the purpose authorized in writing by SanDisk. This document shall be covered as SanDisk's confidential information under all applicable nondisclosure agreements between the recipient and SanDisk.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrievable manner, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written consent of an officer of Western Digital Corporation or its affiliates.

Revision History

Date	Revision	Description
July 2017	F1	Renamed: optimus2_eco_generic_F930_customer.dob To: optimus2_eco_generic_F930_fw_plus_boot.dob
April 2017	F	F930
June 2016	E	Legal Disclaimer updated and Western Digital branding
May 2016	D	F910
October 2015	C	F820
June 2015	B	F760 (Initial release for Eco E3.2)
August 2014	A	F6C2
April 2014	1	F670 Initial Release

Part Number - SKU	Product Description	Firmware Code Name
SDLKODDR-400G-5CA1	Optimus.2 Eco 2.5” 400GB (3 DWPD)	optimus2_eco_generic_F930_fw_plus_boot.dob
SDLKOCDR-800G-5CA1	Optimus.2 Eco 2.5” 800GB (3 DWPD)	
SDLLOCDR-016T-5CA1	Optimus.2 Eco 2.5” 1600GB (3 DWPD)	
SDLLOCDR-020T-5CA1	Optimus.2 Eco 2.5” 2000GB (1 DWPD)	
SDLKODDR-480G-5CA1	Optimus.2 Eco 2.5” 480GB (1 DWPD)	
SDLKOCDR-920G-5CA1	Optimus.2 Eco 2.5” 920GB (1 DWPD)	
SDLKOEDM-200G-5CA1	Optimus.2 Ascend 2.5” 200GB (10 DWPD)	optimus2_generic_F930_fw_plus_boot.dob
SDLKODDM-400G-5CA1	Optimus.2 Ascend 2.5” 400GB (10 DWPD)	
SDLKOCDM-800G-5CA1	Optimus.2 Ascend 2.5” 800GB (10 DWPD)	
SDLLOCDM-016T-5CA1	Optimus.2 Ascend 2.5” 1.6TB (10 DWPD)	
SDLKOEGW-150G-5CA1	Optimus.2 Ultra 2.5” 150GB (30 DWPD)	
SDLKODGW-300G-5CA1	Optimus.2 Ultra 2.5” 300GB (30 DWPD)	
SDLKOCGW-600G-5CA1	Optimus.2 Ultra 2.5” 600GB (30 DWPD)	
SDLLOCGW-012T-5CA1	Optimus.2 Ultra 2.5” 1.2TB (30 DWPD)	
SDLKOE9W-100G-5CA1	Optimus.2 Extreme 2.5” 100GB (45 DWPD)	
SDLKOD9W-200G-5CA1	Optimus.2 Extreme 2.5” 200GB (45 DWPD)	
SDLKOC9W-400G-5CA1	Optimus.2 Extreme 2.5” 400GB (45 DWPD)	
SDLLOC9W-800G-5CA1	Optimus.2 Extreme 2.5” 800GB (45 DWPD)	

ESD Caution – Handling

Static electricity may be discharged through this disk subsystem. In extreme cases, this may temporarily interrupt the operation or damage components. To prevent this, make sure you are working in an ESD- safe environment. For example, before handling the disk subsystem, touch a grounded device, such as a computer case.

Overview

This document describes the fixes and known issues for the Optimus.2 Eco/Ascend/Ultra firmware.

F930 Firmware

The F930 firmware contains the following fixes:

Title: Format corrupt occurs following drive becoming unresponsive and requiring power-cycle
Failure Scenario: During processing of a mixed Read and Write workload from the host to the drive's primary SAS port (Port A), SAS connection errors are observed by the drive. The drive subsequently becomes unresponsive on both SAS ports, and requires a power-cycle in order to restore communication. Following power-cycle, the drive reports 03/31 sense data (format corrupt) due to a failure to read SCRAM data.
Root Cause: During error processing associated with traffic resulting from the SAS connection errors (NAKs and Task Management Function Abort Tasks), the Port A and Port B micro-controllers (MCs) are temporarily halted. If a given MC fails to suspend itself within the allowed time limit (655 ms), then a function is called to log the internal state of both MCs for debug purposes. In this case the MC for Port A failed to halt due to known SAS port abort handling problems. The debug logging function erroneously modified a global micro-controller register file pointer when capturing the state of both ports' micro-controllers, always leaving the global pointer set to the micro-controller belonging to Port B. The calling routine of the debug logging function then used the same global pointer when attempting to recover the micro-controller for Port A, which actually resulted in modifications to Port B's micro-controller. The MC halt routine exited with the MC for Port A still running and in an unknown state. A HARD RESET was then issued by the host, which resulted in a module reset for Port A's micro-controller as part of re-initialization of the Host Port Controller Complex. Because Port A's micro-controller was not properly halted, it was possible for a register access to the Port A host port controller (HPC) by one of the drive's Cortex CPUs to hang because the Port A MC was in the middle of a register access when reset was applied. The register access hang on the Cortex CPU caused the drive to become unresponsive and require a power-cycle. When power-loss occurred, the drive was not able to properly save SCRAM data due to the host port controller hang. The drive was format corrupt on the subsequent power-on due to SCRAM information not having been saved.
Change Description: Modify the debug logging routine to save the register file pointers prior to use, and restore these pointers to the initial value when logging is completed. This ensures that the register file pointers are not changed by the debug logging routine, and avoids the case of the incorrect micro-controller getting programmed.
Likelihood: medium
Severity: data risk

Title: Improve DRAM timing margin
Failure Scenario: ECC failures on marginal DRAM/CPU combinations during normal I/O
Root Cause: DRAM timing settings were not properly configured to provide the largest margin of error.
Change Description: Provide more timing margin on DRAM accesses, by centering the data strobe to be in the middle of the data access.
Likelihood: low
Severity: major

Title: I_T NEXUS LOSS reported following aborted command due to NAK
Failure Scenario: During a workload involving write commands, the drive observed errors on the SAS link, which resulted in the drive NAKing write data frames from the host. An abort was sent to the drive as part of the host's processing of the NAK response from the drive. The drive went unresponsive for a period of 2 seconds, and then reported a Unit Attention for I_T NEXUS LOSS to the host.
Root Cause: The SAS Port micro-controller code had problems dealing with aborts from the host, which led to the drive becoming unresponsive on the SAS port. The 2 second I_T NEXUS LOSS timer was expiring due to these issues, and caused all outstanding commands in the drive to be aborted.
Change Description: Multiple SAS port layer changes were made to address various deficiencies in abort handling to resolve the I_T NEXUS LOSS errors. 1) During Task Management Function Abort processing, SAS micro-controller logic firmware control flags may become out of sync with the port hardware state. 2) Addressed an issue where during NAK error processing, the incorrect context for a read transfer could be loaded, resulting in a CRC error being detected on

the read data, and the command being aborted with 0B/47 sense data. 3) Addressed an issue where SAS micro-controller processing of a BREAK primitive could lead to the link being left in a suspended state, resulting in command timeouts or I_T NEXUS LOSS conditions. 4) Addressed an issue where abort processing may modify register controls associated with the incorrect port, leading to unexpected behavior on the alternate port. 5) Address cases where the SAS micro-controller was leaving the drive's port hardware in a suspended state, resulting in command timeouts. 6) Address an issue with an internal timer resulting in the drive to incorrectly send OPEN REJECT(RETRY) in response to OPEN ADDRESS requests. 7) Address an issue where a Check Condition Response for an aborted transfer was being incorrectly discarded by the SAS micro-controller, rather than being serviced in order to complete the command. This resulted in I_T NEXUS LOSS to be declared by the drive in error.
Likelihood: medium
Severity: major

Title: Drive Reports Incorrect Response to QUERY TASK
Failure Scenario: Task Management Request for QUERY TASK to the drive for a tag which there is no command present in the task set from the specified I_T nexus. The drive reports FUNCTION SUCCEEDED rather than FUNCTION COMPLETE.
Root Cause: A logic error in the process which checks for the specific tag in the drive's internal command queue resulted in the firmware always reporting that the tag had been found, and led to a FUNCTION SUCCEEDED response.
Change Description: Address the logic error in QUERY TASK processing to allow correctly return FUNCTION COMPLETE if the tag is not found in the drive's internal in the task set from the specified I_T nexus.
Likelihood: high
Severity: major

Title: Insufficient Registration Resources seen on PERSISTENT RESERVATION OUT
Failure Scenario: Issuing a PERSISTENT RESERVATION OUT registration with the ALL TARGET PORTS parameter set to 1 would result in an 05/55/04 - Insufficient Registration Resources.
Root Cause: A counter of registered initiators was incrementing based on the size of the World Wide Name (WWN) table and not the actual number of WWNs currently in the table, causing the counter to overflow.
Change Description: Base the registration counter on the number of initiators listed in the WWN table.
Likelihood: high
Severity: major

Title: Code download fails with ABORTED_INSUFFICIENT_RESOURCES sense data
Failure Scenario: WRITE BUFFER command issued to the drive to update microcode returns check condition status with 0B/55/03 sense data indicating ABORTED_INSUFFICIENT_RESOURCES.
Root Cause: If a data transfer for a WRITE BUFFER command to update microcode fails due to a SAS frame error (e.g. CRC or disparity error), the drive will return check condition for the WRITE BUFFER command and return Sense Data with a key of 0x0B, indicating that the command has been aborted. In this case, the drive does not clean up internal resources allocated in order to process the microcode update. Any subsequent WRITE BUFFER commands to update microcode return check condition with 0B/55/03 sense data because download resources are already allocated.
Change Description: Allow the drive to properly clean up internal resources allocated for the download process in the case that the download has been aborted due to SAS frame errors.
Likelihood: low
Severity: major

Title: QUERY TASK incorrectly returns Task Not Found
Failure Scenario: Issuing a QUERY TASK on the final WRITE BUFFER command of a segmented code download returns FUNCTION COMPLETE (Task Not Found) instead of the expected FUNCTION SUCCEEDED.
Root Cause: At the time that the QUERY TASK is issued, the final WRITE BUFFER command has already been removed from the I_T Nexus task list, even though status for that command is held off until the newly-downloaded code has finished updating. As QUERY TASK uses this list to locate the command, the command is not found and the wrong status is returned.
Change Description: Add a condition to QUERY TASK to find and report on a completed WRITE BUFFER command for which status has not yet been returned.
Likelihood: low
Severity: minor

Title: Unexpected Check Condition on Write Buffer command to update firmware
Failure Scenario: A Write Buffer command is issued to the drive with a mode field specifying download microcode. The command is aborted via task management or hard reset following completion of write buffer data being transferred. A subsequent Write Buffer command is issued to the drive to download microcode, which encounters a check condition with sense data of 05/2C/00 (ILLEGAL SEQUENCE ERROR).
Root Cause: The abort for the write buffer command was processed by the drive after the download image was validated, and the request to program flash was initiated. The drive did not allow cleanup of code download resources for the abort in this case, since they were in use for flash programming. Following completion of flash programming, there was no path for this download abort cleanup to take place. A subsequent request to download microcode encountered a 05/2C/00 (ILLEGAL SEQUENCE ERROR) due to a global variable "expected offset" code download variable not getting reset. This global variable is normally initialized during download abort cleanup processing.
Change Description: If abort processing could not take place due to flash programming in progress, then a flag will be set, indicating that cleanup needs to take place once flash processing has completed.
Likelihood: low
Severity: minor

Title: False I_T NEXUS LOSS OCCURRED following TASK MANAGEMENT FUNCTION requests
Failure Scenario: Following processing of TASK MANAGEMENT FUNCTION requests, the drive reported I_T NEXUS LOSS OCCURRED in error.
Root Cause: Improper processing of TASK MANAGEMENT FUNCTION requests in the SAS port layer code resulted in a communication stall between the drive's SAS port logic and the drive's device server logic. The communication stall resulted in I_T NEXUS LOSS being detected at the device server logic level.
Change Description: SAS Port layer code was updated with multiple changes in the TASK MANAGEMENT FUNCTION processing area to address the following symptoms: Code deadlock resulting in I_T NEXUS LOSS. I_T NEXUS LOSS following TASK MANAGEMENT FUNCTION - ABORT TASK. Drive attempts to open the wrong WWN. False tag overlap detection, which could lead to command timeout in multi-initiator environments.
Likelihood: medium
Severity: minor

Title: Enforce WRITE BUFFER MODE field consistency when using multiple WRITE BUFFER commands to download microcode
Failure Scenario: During compliance testing, drive microcode was being updated by a series of WRITE BUFFER commands which did not use the same value for the WRITE BUFFER MODE field. The drive was expected to reject a follow-on WRITE BUFFER command which did not contain the same value in the WRITE BUFFER MODE field as the prior WRITE BUFFER command.
Root Cause: The drive's WRITE BUFFER command processing was not checking for consistent values in the WRITE BUFFER MODE field.
Change Description: WRITE BUFFER MODE field consistency enforcement was added to WRITE BUFFER command processing. The drive will return check condition to the host with 05/2C/00 Sense Data for Illegal Sequence Error in the case that a WRITE BUFFER command to update microcode is received with a different WRITE BUFFER MODE field value than the initial WRITE BUFFER command.
Likelihood: low
Severity: minor

Title: GET LBA STATUS does not handle an ALLOCATION LENGTH < 24 correctly
Failure Scenario: The host issues a GET LBA STATUS command with an ALLOCATION LENGTH less than 24.
Root Cause: When the ALLOCATION LENGTH field has a value less than 24, the drive erroneously returns a value of 4 in the parameter data's PARAMETER DATA LENGTH field instead of an allowed value.
Change Description: The value returned in the PARAMETER DATA LENGTH field has been corrected. The drive ensures that at least one LBA status descriptor is returned.
Likelihood: medium
Severity: minor

Title: Link Counters reported incorrectly
Failure Scenario: SAS Link counter registers become saturated in a multi port drive environment. The counters may be reported by the drive incorrectly. It is also possible that the drive performance becomes degraded.
Root Cause: The drive controller receives an interrupt when a Link counter reaches saturation. The drive logic to process the saturation event used a stale pointer to access the PHY control registers. If the stale pointer happened to be set to the wrong port's PHY control registers, then the link counter saturation event would not be handled properly, resulting in incorrect Link event counters. Also, the interrupt for Link Counter saturation would be repeated for each counter increment, which could affect drive performance due to excessive interrupt handling overhead.
Change Description: Ensure that the PHY control register pointer is updated properly before being used in Link counter saturation processing.
Likelihood: medium
Severity: major

Title: Port Layer Improvements for Multi-Initiator System Setups
Failure Scenario: Running in a multi-initiator queued-workload environment, the drive will occasionally time out while sending status for a command.
Root Cause: Context switching and routing between multiple initiators' outstanding workloads could occasionally cause SSP Response Frames to be incorrectly routed, or dropped before delivery.
Change Description: Multi initiator improvements:
1. Increased the priority of WWN interrupts in the Port layer interrupt service routine.
2. Fixed a possible code hang in the "wait for receive path idle" code. Allow for instances where this path never goes idle.
3. Made Port layer improvements to address issues seen running multi-initiator firmware update tests.
4. Added logic to detect and record instances where the incoming SAS frame WWN is not what the drive was expecting. Logging this event would indicate an error in the port initiator context control logic.
5. Added logic to detect pending WWN interrupts when interrupts are disabled. The new logic temporarily re-enables interrupts to service the WWN.

General port layer updates:
1. Made improvements to the Port layer handling of BREAK primitives.
2. Removed obsolete recovery logic from the data frame handler.
3. Made improvements to the command abort path that eliminates a race condition detected during aggressive hard reset testing.
Likelihood: medium
Severity: major

Title: Significant performance degradation when data retention recycling begins
Failure Scenario: Reduced performance and long latency commands detected when data retention recycling starts during host commands.
Root Cause:
(1) A multiplier was added to accelerate reclamation recycling when necessary. This multiplier was incorrectly applied to data retention recycling as well.
(2) Unnecessary reclamation recycling operations can start on blocks that have just completed data integrity recycling.
(3) At power-on initialization, multiple superblocks can be queued for data retention recycling, even though data integrity recycling does not begin until initialization completes.
(4) Superblocks with host data and empty superblocks can both hit the data retention limit and require recycling. The old but empty superblock can be selected as the destination superblock, and may immediately require retention recycling after filling.
Change Description:
(1) Reduce the data retention minimum ratio to account for the multiplier used for reclamation recycling.
(2) Prevent initiation of unnecessary reclamation recycling while waiting for completion of the data integrity recycling of the same superblock.
(3) Do not queue data retention recycling superblocks during initialization.
(4) If available, choose an unused superblock from the data retention recycle list to process before selecting superblocks containing host data.
Likelihood: low
Severity: minor

Title: Performance degradation when processing UNMAP commands
Failure Scenario: Processing a large UNMAP command degrades host workload performance by an unacceptable amount.
Root Cause: Within the FTL, unmap commands are broken up into smaller subtasks and scheduled in a loop that includes other tasks, including host commands. When successive unmap subtasks are being sent to the scheduling loop, host commands do not get enough time to run.
Change Description: Change the scheduler to skip the unmap task a fixed number of times when it appears in the scheduler loop, to allow enough time to process host commands. This change only affects the scheduling loop when an unmap task is present.
Likelihood: medium
Severity: minor

Title: Add Forced Write Feature to the Memory Dump Command
Change Description: Implement a new option for the Memory Dump command (0xF7) that will force an internal memory dump, even if there is already a previously written memory dump.

Title: Add periodic logging of additional FTL information to the event log
Change Description: Periodically record FTL information in the internal event log in order to facilitate debug of performance issues. Information is recorded to the internal event log every 20 minutes if there has been any related activity.

Title: Multi initiator frame misdirection
Failure Scenario: In a multi initiator system the drive could direct an XFER_RDY to the wrong initiator leading to the command being aborted for an Initiator response time out.
Root Cause: A corner case was found in the connection context switching code when sending XFER_RDYs for write data. If the lookup of the WWN for the initiator was delayed for more than 810uSec the drive would use the wrong value. This caused the drive to send out an XFER_RDY to the WWN saved at index 0 of the WWN table.
Change Description: Eliminated multiple sources of delay that could cause the 810uSec internal timer to expire in error during context switching.
Likelihood: low
Severity: major

Title: Invalid tag overlap reported
Failure Scenario: In a multi initiator environment where all the active Initiators are using a small number of similar command tag values the drive could reject an incoming command as a tag overlap when no actual overlap existed.
Root Cause: The firmware responsible for removing tags from the active tag table was using an index value that was not always valid in a multi initiator system.
Change Description: Fixed the removal of tags from the active tag table to ensure the use of the valid index.
Likelihood: low
Severity: major

Title: Allow SAS negotiation to continue if reserved fields in the IDENTIFY Frame are set.
Failure Scenario: Following the IDENTIFY sequence between host and drive, SAS negotiation failed to progress further (with the drive resetting the link).
Root Cause: A previously-reserved field in the IDENTIFY Address Frame was being verified to contain all 0x00 values. With the usage of the POWER CAPABLE bit field in one of those reserved bytes in newly developed HBAs, the reserved-field check was failing and thus the drive would not acknowledge a valid IDENTIFY frame from the host.
Change Description: Suppress reserved-field checking in the IDENTIFY Address frame to allow handshaking to continue.
Likelihood: low
Severity: major

F910 Firmware

The F910 firmware contains the following fixes:

Title: SANITIZE with OVERWRITE service action and IMMEDIATE=1
Failure Scenario: SANITIZE with OVERWRITE service action and IMMEDIATE=1 results in inconsistent LBA data.
Root Cause: In the specific case of SANITIZE with OVERWRITE and IMMEDIATE=1, the initialization pattern is read from the incorrect place, and read cache is not invalidated, causing inconsistent results.
Change Description: Use the correct reference for the initialization pattern buffer, and invalidate the read cache.
Likelihood: low
Severity: critical

Title: SANITIZE command hangs with a Sanitize operation in progress
Failure Scenario: SANITIZE issued with a Sanitize operation in progress results in a drive hang.
Root Cause: Status is not being set correctly for a SANITIZE command that is received while a Sanitize operation is in progress.
Change Description: When a Sanitize operation is in progress, respond to a new SANITIZE command with LUN NOT READY, SANITIZE IN PROGRESS (02/04/1B) status.
Likelihood: low
Severity: major

Title: WRITE SAME command timeout
Failure Scenario: A WRITE SAME command did not return status during high queue depth traffic.
Root Cause: A WRITE SAME command did not complete during high queue depth traffic because of resource starvation.
Change Description: Properly reschedule a WRITE SAME command's execution when resource becomes available.
Likelihood: low
Severity: major

Title: Power failure may result in format corrupt
Failure Scenario: Format corrupt after power failure.
Root Cause: A loop within a specific function can take 200 msec to complete and power failure was not checked during the loop. Depending upon timing, there may not be time to SCRAM data.
Change Description: Check for pfail within the function loop and terminate when pfail activated.
Likelihood: low
Severity: major

Title: Write performance could suffer after several HARD RESET events
Failure Scenario: Following several HARD RESET events, the drive's write performance was degraded.
Root Cause: HARD RESETs occurring while small write buffers are owned by the SAS port could result in the buffers being lost until the drive is power-cycled. The drive's HARD RESET processing was not properly bookkeeping all cases of small transfer buffers primed into the ports through a HARD RESET.
Change Description: Re-sequenced the FIFO resets so that buffers being released by the port were not lost during the HARD RESET processing.
Likelihood: medium
Severity: major

Title: Drive became unresponsive following firmware download
Failure Scenario: Multi-Initiator code download testing (with high IO volume). The drive stopped responding to SAS commands, resulting in command timeouts.
Root Cause: The code download processing was waiting for the SAS Link to be IDLE before proceeding to boot-up with new the code-image. In the failure scenario, the SAS Link was in a suspended state, and the IDLE condition was never met, which prevented the drive from loading the new code-image, and caused pending commands to time-out.
Change Description: Updated code download processing to suspend the SAS port's Link state and wait an additional 500us for all active connections to close. This eliminates possible 'race' conditions and properly handles the transition to the new code.
Likelihood: low
Severity: major

Title: Unaligned 4K Block Write Transfer Failure Fix
Failure Scenario: After a hotswap an unaligned 4K block write failed to transfer data properly.
Root Cause: Incorrect LBA bitmask management.
Change Description: Corrected the LBA bitmask for unaligned 4K block writes.
Likelihood: medium
Severity: major

Title: Unmap interrupted by power fail can result in read-only drive
Failure Scenario: Power failure during unmap resulted in read-only drive.
Root Cause: The unmap batches up the valid page count (VPC) update to the super block at the end of function. If there is a pFAIL before this happens, then there is a mismatch in VPC from the persistent info (the wrong value) on the next power up as init does not replay the VPC updates of the super blocks.
Change Description: Replay the VPC updates if unmap was interrupted by a pFAIL after the L2P was updated but before the VPC updates.
Likelihood: low
Severity: major

Title: Power failure handling of partition load
Failure Scenario: Format corrupt occurred after power failure.
Root Cause: FTL code was searching for properly loaded L2P partitions when power failure was detected which delayed the start of SCRAM.
Change Description: Terminate the routine if pfail detected so SCRAM operations can begin.
Likelihood: low
Severity: major

Title: SAS link reset after firmware download
Failure Scenario: During a firmware download, a link reset occurred.
Root Cause: The phy registers were re-programmed during the start-up with the newly downloaded firmware. The hardware register was cleared before writing the desired value. The brief incorrect register values caused some host adapters to bring the link down.
Change Description: Changed the post-firmware-download start-up behavior to not re-program the phy registers. The registers are programmed after the next power cycle.
Likelihood: low
Severity: major

Title: Drive returns incorrect WWN hash address in Read data
Failure Scenario: Drive returns incorrect WWN hash address in Read data in a high-traffic, high queue depth, multi-initiator workload environment.
Root Cause: 1. When two incoming SAS frames from two different initiators are received back-to-back, the WWN hash from the second frame's hardware register may overwrite the value from the first frame. 2. The WWN index table was not cleared during LUN Reset, resulting in a stale entry being used.
Change Description: 1. Detect the race condition and use the correct WWN value from the index table. 2. When a LUN Reset is received, clear the hash hardware register and clear the WWN index table to force the drive to re-populate the WWN table when an initiator logs in.
Likelihood: low
Severity: major

Title: Command timeout occurs in workloads with overlapped tags
Failure Scenario: In queued workloads with multiple initiators, where overlapped tags are issued for outstanding commands, the drive may stop responding to I/O commands.
Root Cause: In the failure scenario, the SAS Port layer logic identified a tag overlap condition, but the device server logic did not find the associated command to abort. A problem with the abort processing in this case caused commands from the wrong nexus ID to be aborted, resulting in command timeouts to be seen by the host for those commands.
Change Description: Addressed the flaw in overlapped tag abort handling to correctly identify the Nexus ID of commands to abort.
Likelihood: low
Severity: major

Title: Data Abort exception occurs during code download
Failure Scenario: In a multi-initiator environment, while the drive was processing high volume I/O, a code download occurred, and the drive encountered a Data Abort exception and stopped responding to commands, resulting in command timeouts, and code download failure.
Root Cause: Due to a command exception, the firmware had performed code download cleanup processing when it was in a state where it had already committed to flash programming. When flash programming processing started, the Data Abort exception occurred.
Change Description: Download processing was modified to ensure download cleanup does not occur if flash programming has not yet occurred, and the drive is committed to flash programming.
Likelihood: low
Severity: major

Title: Firmware Download Error
Failure Scenario: Firmware download may not complete.
Root Cause: During firmware download the hardware auto-transfer-ready buffer management detects buffer loss due to a firmware error.
Change Description: Firmware has been fixed to properly manage auto-transfer-ready buffers during firmware download.
Likelihood: low
Severity: major

Title: Drive does not reject SANITIZE command with write protected medium
Failure Scenario: A SANITIZE command issued when the medium is write protected does not fail with data protect (07/xx/xx) status.
Root Cause: Drive SANITIZE processing was missing a check for the write protection case.
Change Description: Fail a SANITIZE command with the appropriate data protect status (e.g. 07/27/00) if the medium is write protected.
Likelihood: low
Severity: major

Title: Unmap during initialization causes assert and hang
Failure Scenario: Unmap command issued immediately after power cycle resulted in hang.
Root Cause: During initialization, unmap commands that were being processed prior to the L2P partition file completing can cause an assert and drive hang.
Change Description: Defer the processing of unmap command till the partition table is completely built.
Likelihood: low
Severity: major

Title: Drive not ready after interrupted SANITIZE OVERWRITE
Failure Scenario: A SANITIZE OVERWRITE operation that is interrupted by a power fail cannot be restarted.
Root Cause: In the case of a SANITIZE OVERWRITE that is interrupted by a power cycle, the state of the medium will be set to Not Ready, which prevents writes from completing, and effectively prevents SANITIZE OVERWRITE from being restarted.
Change Description: Set the medium state correctly for interrupted SANITIZE OVERWRITE operation after power on.
Likelihood: low
Severity: major

Title: Format corrupt after power cycle
Failure Scenario: Power cycle during Data Retention Recycling can lose parity and go format corrupt.
Root Cause: The failure is caused because we don't have parity data in order to recover a page in a partially written stripe during init reads of open super blocks. We don't need the parity data if we can recover the page with just doing read retry.
Change Description: Even though we don't have parity data for the stripe, attempt a read retry to recover the page.
Likelihood: low
Severity: major

Title: WWN mismatch when running Multi-Initiator I/O
Failure Scenario: When running multi-initiator I/O at high queue depth, Initiator Response timeouts occurred.
Root Cause: Back-to-back frames in the hardware from different initiators caused the hardware to mistakenly change the hash address of the previous frame to that of the incoming frame. This was a hardware bug.
Change Description: A hardware feature was invoked to detect the hash mismatch and have the firmware correct the hash address.
Likelihood: medium
Severity: major

Title: FORMAT command doesn't complete
Failure Scenario: Repeated FORMAT commands can consume resources and result in a FORMAT command not completing.
Root Cause: When the format is performed, buffers from host die operation history are not released. So after many formats, write buffers are exhausted and a hang results due to insufficient resources.
Change Description: Properly release history buffers when formatting the drive.
Likelihood: low
Severity: major

Title: Drives not admitted into Storage Spaces in a Microsoft Windows environment
Failure Scenario: Drives were not being admitted into Storage Spaces in a Microsoft Windows environment.
Root Cause: Microsoft Windows was not recognizing drives which reported a value of greater than 3 in the LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field for READ CAPACITY.
Change Description: Limit LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT reporting to 3 for READ CAPACITY.
Likelihood: medium
Severity: major

Title: Logical Block Application Tag mismatches are not detected
Failure Scenario: Logical Block Application Tag mismatches are not detected by 32-byte SCSI commands.
Root Cause: The firmware inadvertently fails to propagate its calculated Application Tag check flag to the hardware.
Change Description: The firmware programs its calculated Application Tag check flag into the hardware.
Likelihood: low
Severity: major

Title: Incorrect sense data in check condition for Write with zero byte payload
Failure Scenario: Write with injected zero-length data payload. Drive was incorrectly reporting 0B/4B/07/01 (BREAK ENCOUNTERED).
Root Cause: Incorrect sense data was being returned by drive
Change Description: Changed sense data to be 0B/0E/00/00 (INVALID INFORMATION UNIT).
Likelihood: low
Severity: minor

Title: Fix and enable SCRAM monitoring during code load
Failure Scenario: SCRAM monitoring for code load was previously removed due to an issue causing drive hangs.
Root Cause: Attempts to access shared memory by both CPUs resulted in an AXI bus hang.
Change Description: Coordinate the timing of CPU restarts such that each processor can receive data from shared memory when requested and restart when safe. Enabled SCRAM monitoring for code load with resolved CPU coordination.
Likelihood: low
Severity: minor

Title: Program and erase failures may cause critical event
Failure Scenario: Injected random write and erase failures (400 in a row) cause critical event to be logged.
Root Cause: Log writes and host writes save sequence numbers when programmed and a program failure can alter what is recorded.
Change Description: Make sequence numbers be consistent between memory versions and cached versions for failure cases.
Likelihood: low
Severity: minor

Title: Poor performance seen in sequential workloads
Failure Scenario: Performance issues seen in 10/90 R/W sequential workload.
Root Cause: Recycle logic chose poor candidates to recycle when sequential workloads were invalidating 90% of the superblocks. This caused higher write amplification and low performance.
Change Description: Changed the sparse pool delineations to better identify low valid page count superblocks, and allowed logic to choose alternative subdrives and superdevices when much better choices existed.
Likelihood: medium
Severity: minor

Title: Average number of block erases reported incorrectly
Failure Scenario: Average number of block erases reported incorrectly in log page 31h.
Root Cause: Calculation did not account for the number of blocks per die.
Change Description: Fixed the average block erase and average page program calculations.
Likelihood: high
Severity: minor

Title: Reduced performance after quick repeated power cycles
Failure Scenario: Hot swap during write workload can produce drive logs with many critical events logged, which could lead to reduced performance.
Root Cause: When drive is power cycled frequently before background initialization completes, incoming host writes could consume free blocks and cause drive to enter critical reclamation mode.
Change Description: Change the recycle-during-background-initialization trigger threshold and allow recycle writes during background initialization.
Likelihood: low
Severity: minor

Title: Excessive data retention recycling reduces performance
Failure Scenario: After an extended idle period, lower performance is observed when the workload resumes.
Root Cause: Data retention recycling was performed on superblocks that did not require immediate recycling.
Change Description: Select superblocks that will not require data retention recycling immediately.
Likelihood: low
Severity: minor

Title: Reduced performance after repeatedly reading the same block
Failure Scenario: Repeatedly reading the same block may lead to reduced performance.
Root Cause: Repeatedly reading the same block forces read disturb recycling, and will eventually move the superblock to a separate recycle list. The search for a recycle candidate can miss this entry.
Change Description: Consider alternate super devices when recycling; select a recycle candidate from the read disturb recycle list when available.
Likelihood: low
Severity: minor

Title: Command timeout may occur during heavy recycling
Failure Scenario: Write command timeout detected after idle period.
Root Cause: Performing different types of recycling concurrently can consume available resources and prevent host writes from completing.
Change Description: Reduce number of concurrent recycle operations.
Likelihood: low
Severity: minor

Title: Persistent reservation key lost
Failure Scenario: - Initiator 0 attempts and successfully makes a persistent reservation. - Initiator 0 successfully reads back the persistent reservation key. - Initiator 1 attempts to make a persistent reservation and is rejected. - Initiator 0 unsuccessfully reads back the persistent reservation key a second time. The persistent reservation key is reported as lost.
Root Cause: A code path was incorrectly decrementing a reservation count when a persistent reservation was rejected.
Change Description: Correctly decrement the reservation count when an initiator unregisters it.
Likelihood: low

Severity: minor

Title: Sequential Read Prefetch Improvements
Failure Scenario: Queued multiple overlapping sequential read streams (e.g. multiple host threads all issuing a sequential read stream from the same start LBA). Performance across the range of queue depths may be lower than expected.
Root Cause: In queued multiple overlapping read stream workloads, the earlier-issued streams can get ahead of the later-issued streams in LBA space. In such a workload, the prefetch algorithm would sometimes re-trigger at a lower LBA to service one of the trailing streams instead of maintaining prefetch ahead of the forward-most stream. This resulted in the same data being prefetched multiple times and a degradation in performance.
Change Description: A change to the prefetch algorithm was made to not trigger a new prefetch if there has been a recent hit on the current prefetch data stream. In the high queue depth, multiple overlapping read stream case, prefetch is now maintained ahead of the forward-most stream.
Likelihood: medium
Severity: minor

Title: Incorrect error code may be returned for uncorrectable read
Failure Scenario: After WRITE LONG is used to create an unrecoverable block, a subsequent READ may result in a DATA BCRC ERROR DETECTED (0B/47/82) error instead of a MARKED BAD BY APPLICATION CLIENT (03/11/14) error if the bad block is not at the beginning of the requested read range.
Root Cause: The unrecoverable code path was different depending upon whether the host data transfer had started or not. In the case of when the transfer has started, there was debug code that incorrectly caused the wrong error code to be returned.
Change Description: Removed the debug code.
Likelihood: medium
Severity: minor

Title: Long initialization time
Failure Scenario: Sparse writes followed by power cycling can result in long initialization time.
Root Cause: Sparsely written data generates metadata that requires a lot of re-arrangement and copying at init time.
Change Description: Improve meta-data processing during data block recycling by utilizing hardware-assisted DMA copy to speed-up the copying process time.
Likelihood: low
Severity: minor

Title: Hang after power cycle during I/O
Failure Scenario: Hang could occur after power cycle while running mixed I/O workloads with UNMAP commands.
Root Cause: The log recycle metadata buffer is incorrectly addressed and could result in lost buffers when aborting and restarting a metadata read operation.
Change Description: When restarting log recycling, operate on the correct logging superblock reference.
Likelihood: low
Severity: minor

Title: Implement GET LBA STATUS command
Change Description: Implemented GET LBA STATUS command.

Title: GET LBA STATUS can report the wrong provisioning status soon after startup
Failure Scenario: GET LBA STATUS commands issued immediately after firmware startup can report a provisioning status of Mapped for LBAs that should be reported as Anchored.
Root Cause: The wrong criteria is used to decide whether a logical page's context has been loaded during flash layer initialization, such that pages with incomplete context are reported as Mapped.
Change Description: Use the proper criteria to decide whether the provisioning status for a logical page can be reported.
Likelihood: low
Severity: minor

Title: HLBA error on READ command
Failure Scenario: HLBA mismatch error on read error recovery during mixed workloads.
Root Cause: During completion of host reads using media queues, both CPUs can simultaneously access a shared memory structure. The Flash layer CPU doesn't post a completion message to the Front End CPU in case of media queue reads, assuming that the Front End CPU is notified of the completion by the port layer. This leads to a window in which both CPUs can access the structure simultaneously.
Change Description: The Flash layer will make a copy of the required information in shared memory when the command is initially received from the Front End CPU, so that the access collision will not occur at completion time.
Likelihood: low
Severity: minor

Title: Incorrect drive handling of NAK error on XFER_RDY
Failure Scenario: When a NAK was received in response to an XFR RDY from the drive, the drive responded with 0B/47/01/01 (Data Phase CRC Error Detected - NAK Sent). This is incorrect. It should be 0B/4B/04 (NAK Received).
Root Cause: The drive was waiting for the host to send all the data and then responded with Data Phase CRC Error Detected - NAK Sent.
Change Description: Fixed code to respond with proper sense data when a NAK is received for an XFER_RDY.
Likelihood: low
Severity: minor

F820 Firmware

The F820 firmware contains the following fixes:

- Implemented Read/Write (transitional) performance improvement.
- Fixed an issue where a TRIM at PFAIL along with Host Writes to the same LBA's can cause Read Only drives.
- Fixed an issue where the drive ignores initiator Open Request and becomes non-responsive during link up/down testing. Enabled normal link operations by setting the 'kReset_IDX_Register' bit in the Link Control Register after an Abondon Class Primitive is received.
- Fixed an issue where Read Capacity 10, byte 1 and byte 8, bit 0x0 obsolete field must return a Good Status.
- Fixed an error in MC_Release_Rx_Resources that mis-handled the abort of a non-block data command with an RX AES context slot assigned to it.
- Fixed an issue found during internal stress testing where a Critical Event was noticed during Notify Power Loss Expected Primitive insertion between Write Data.
- Fixed an issue when a synchronize cache(16) returned GOOD status when the LBA specified exceeded the Maximum LBA of the drive.
- Fixed an issue that caused a Request Sense command to increment the Non-Medium Error Log Page counter (06h).
- Added OP (Over Provision percentage), DWPD (Drive Write Per Day) and EOL PE (End-of-Life Program Erase) count to Inquiry page E9h.
- Fixed an issue where the internal value used to calculate percent_life_left was incorrectly using 10 DWPD rather than 1 DWPD value. (Eco only)
    - Note: Although the value will decrease to the correct value, the drive will continue to function as advertised from a Total Bytes Written/DWPD perspective.
- Fixed an issue found during internal stress testing where Good Status was being returned even if the OPEN for Status frame was being denied longer than the IT Nexus Loss timeout.
- Fixed an issue where percent of life used was not being reported correctly in Log Page 11h immediately following a power cycle. (Eco only)
- Fixed Mode Page E8h.
- Fixed an issue that caused false negatives for bad PI Ref Tags.

Known issues with F820

Optimus.2 Eco/Ascend/Ultra/Extreme drives have the following known issues:

- SanDisk FWQA testing indicates a potential of seeing a port lockup condition in multi-initiator environments with high stress levels. These issues are still under investigation at SanDisk and will be driven to root cause and a solution. SanDisk currently believes these lockups would not be seen within reasonable operating conditions and are a result of the extreme conditions presented by the FWQA regression tests:

  Test Environment: Multiple Initiators, through an expander to the drives while subjecting the drives under test to:

  - Varied I/O workloads submitted to the drive while injecting the following:
    - Varied queue depths 1, 8, 16, 32, 64, 126 (depth on each port). Task Set Full conditions at QD 126.
    - Extra traffic – random non-block commands (non-block commands including illegal requests).
    - Hard Resets injected at fixed intervals / Cable Pulls via a hardware signal controller.
    - Abort task set injected.
    - Firmware update testing

F760 Firmware

The F760 firmware contains the following fixes:

- Implemented a fix for an issue that was causing a command timeout event.
- Fixed an issue where low OP drives have incorrect database maximums.
- Added a safeguard against a memory allocation failure to prevent drives from hanging.
- Changed the default global SCRAM timeout from 5 milliseconds to 6 milliseconds.
- Added a persistent non-volatile counter to track single-bit ECC corrections.
- Enabled write limiter by default to reduce write fragmentation that improves write performance at high QD and xfer-size.
- Added changes to disable rate matching support and increase queue depth support to 252.
- Fixed Logical Block Reference Tag error being falsely reported on 32 byte read command.
- Modified port behavior so that after the new firmware is written to the SPI and before the new firmware gets activated, the port sends out BUSY to all requests from the initiator. We can now safely come back up with the new firmware and continue as normal.
- Fixed an issue with sense data errors in Send diagnostic command.
- Fixed an issue to check the ParameterListLength field in the Unmap CDB.
- Changed Persistent Reservation Out command to correctly ignore non 0 Scope values for Service Actions, Register, Register and Ignore, and Clear.
- Fixed an issue where a response to a read command issued to flash was not being handled correctly while Servicing a verify operation.
- Changed Persistent Reservation Out command to check Allocation Length and return kPARAMETER_LIST_LENGTH_ERROR if it is !=24 bytes.
- Ensured that overlap_chain is always flagged as invalid on all commands, even immediately after reboot when a slot is being used by a read.
- Avoided faulty ACK/NAK Time Out (ANTO) on certain retries exhausted cases.
- Avoided early ReadQueue transmits from accessing unexpected DDR location causing single and double bit DDR errors.
- Resolved an issue where by recovering from a failed read, we read the data from DDR but don't take into account the allocation unit of the recovery.
- Fixed rate matching logic so that it will restore the Link state for normal operations when a RETRY EXHAUSTED (or CONNECTION DENIED with no extended status) event happens. This will keep the drive from going unresponsive.
- Moved process_unmap_region back into fast RAM to restore its performance.
- Added safeguard against memory allocation failure to prevent drives from hanging.
- Fixed an issue where not removing tags after errors caused overlapped commands.
- Fixed and issue for SCRAM margining on software reset and code downloads.
- Fixed an issue for Persistent Reservation Out issues seen in Microsoft "SCSI-3 Persistent Reservation requirements test" and "Validate Storage Spaces Persistent Reservations test". Added code additions to the handling of "Register and Ignore" and "Preempt".
- Fixed an issue that caused HEAD OF QUEUE commands to block other SIMPLE commands.

- Added a check to ensure a correct response when a parameter list length resulted in truncation.
- Implemented a WriteLimiter functionality which prevents Write commands from starting if there is too much write data already in flight.
- Fixed an issue where an Ack/Nak timeout on a read could cause IT Nexus loss instead of reporting Aborted Command.
- Shortened the host data transfer lengths for 4Kb+ block size in socal_regression_pi2.py that can cause Windows to blue-screen.
- Repaired an inadvertent change to the TOKENIZED_PRINT that generates the FLASH_EXERCISER_ROW entry in the common diagnostic output.
- Added a compile time assert to check whether MAX_DEFECTS_IN_DB can actually guarantee a certain percent of erase blocks.
- Increased the threshold for doing 100% recycles.
- Increased the CUTOFF_AGE_OF_PENDING_LOGICAL_PAGE to take into account the size of the logical pages.
- Increased the precondition of power cycle tests to prevent event injection failure.
- Added supported builds to Makefile and improved layout of help.
- Enabled support for Rate matching. Fixed ITNL and IRTOs in multi-initiator environment with mixed read and write traffic.
- Fixed an issue found in multi-initiator TMF testing.
- Added a feature which will allow events in the event log to consume less space, which will allow for more events to be stored in the log.
- Added support for temperature throttling on PCB-based NAND sensors.
- Set the subpage write caching feature to enable by default.
- Resolved an issue related to stranded bytes in the hardware FIFO that could result in an 03/0C/00/06 sense key.
- Disabled the scram margining feature which caused the erasing of the area after each code load.
- Reduced code load time.
- Fixed an issue with the SMART min/max data being unnecessarily saved after every power cycle. It should be saved only when one of the min or max values change.
- Changed the SDF ID of the saved write amplification min/max values so that any previous (and false) values will be forgotten.
- Fixed an issue where Read Buffer mode 0 was returning an incorrect status.
- Resolved an issue with the SPI flash where a sector was stuck in a state that appeared to be erased, but writing to that sector caused program errors. This would happen if the drive lost power while that flash section was being erased.
- Fixed an issue where the Read and Verify counters were getting modified due to false unrecovered errors introduced by Write Long command.
- Made modifications that provide steadier performance at low queue depths.
- Made changes so that the drive checks for data alignment for the Write Buffer command.
- Added support to notify CPU0 of recycle failure.
- Added support for a SMART trip in the event of a SPI flash write failure.
- Fixed an issue to protect the SCSI format against power failures.
- Fixed an issue of CPU0 data aborts in the event log handler when 15 parameters are used.
- Fixed an issue to give INITIATOR RESPONSE TIMEOUT priority over IT NEXUS LUN TIMEOUT when they are set the same.
- Added updates for high queue depth and PI Stripping.
- Avoided false PI Stripping detection and could lead to false ITNL and bad link behavior.

Known issues with F760

- Drive failed during SRT Power-cycle.
- Drive hit a mismatched Valid Page Count for Closed SB failure during FlashManagement-FI_WriteInterrupt Test.
- Drive failed with ERROR_DATA_OVER_UNDER_FLOW during TMF multi initiator tests using Oak Gate.
- Drive failed with a IO Hung Error during IO Multi-Initiator tests using Oak Gate.
- Drive became inactive or offline during Cable Pull signal only.
- When formatting with Type 2 PI, under rare conditions the drives may cause a command timeout on reads.

F6C2 Firmware

The F6C2 firmware contains the following fixes:

- Fixed an issue with Send Diag Self-test that caused various data errors on verify, read, and write verify commands while running command loop tests.
- Improved performance boost when reading and verifying unmapped LBAs.
- Fixed an issue in the format path that was causing LBA counts to be truncated.
- Fixed an issue in the detection of reserve byte/bit violations in performance commands.
- Fixed an issue so that the drive does not return a Reservation conflict if the Standard Reservation holder issues a second Reserve command.
- Fixed an issue where the drive would return an unexpected 0B/44/00/03 DELIVERY_TIME_LIMIT_EXCEEDED after having run large IOs on commands that may take longer than 100ms.
- Fixed an issue that caused the drive to return invalid 05/24/00 errors under mixed workloads using 10 and 16 byte commands.
- Removed un-needed 5 sec delay from Send Diag Short self-test.
- Fixed an issue with the handling of skipwrites and overlap detection.
- Added 48-bit LBA support to overwrite.
- Fixed an issue with Write Buffer command for a transferlength of 0.
- Improved write bandwidth at 16K to 32K transfer sizes by increasing AXR allocation from 8K to 16K and by disabling AXR on xfers larger than 16K.
- Added support for an RSA signature on firmware images.
- Fixed an issue that was causing Write Same to fail.
- Added support for 4096 + PI2 sector size.
- Made performance enhancements to WriteSame in uninitialized/unmapped pages.
- Modified the code to reject a Write Buffer command if the allocation length is larger than the Write Buffer Max Download Size.
- Changed percent drive life used SMART trip threshold from 0% to 2%.
- Added support for parameter 5 in the Verify Error Counter log page (05h).
- Fixed an issue when a drive temperature goes over the threshold it will report the warning or trip immediately instead of delaying it by up to 15 minutes.
- Fixed an issue where Linux mounted Optimus ECO resulted in an aborted command during read/write.
- Added support for the Exit Failure Mode and Overwrite service actions in the Sanitize command.
- Added an event log for changing the SMART sense code.
- Added support for the PBLOCK bit in the Read Long and Write Long commands.
- Disabled SCSI Protection Information so that formatting the drive with PI enabled is illegal. When the drive is formatted with PI disabled the customer will not see the problem that can occur when PI is being stripped during a Read command. If the drive remains formatted with PI enabled after loading this code, its behavior will not be changed except to make formatting with PI enabled illegal.
- Fixed an issue where the Interrupted SANITIZE operations will start again after a power cycle.