Proposed Draft **Serial ATA International Organization**

Version 4

January 04, 2016

Serial ATA Revision 3.3 Technical Proposal #076
Title: Durable/Ordered Write Notification

Author Information

Author Name	Company	Email address
Nathan Obr	Western Digital	Nathan.Obr@WDC.com

Workgroup Chair Information

Workgroup (Phy, Digital, etc…)	Chairperson Name	Email address
Digital	James Hatfield	James.C.Hatfield@seagate.com

Document History

Version	Date	Comments
0	2015-10-26	Initial draft
1	2015-11-02	Reworked concurrent completions of same group IDs on multiple commands in the DURABLE/ORDERED WRITE NOTIFICATION subcommand overview section. Separated out the explicit high priority behavior. Addressed the Isochronous value. Added section 5.3.1 [13.6.2.4] Priority to list subcommand with other NCQ commands.
2	2015-11-16	Replaced the list of commands in the explicit high priority behavior paragraph with 'other NCQ commands'.
3	2015-12-07	Formatting modifications Added informative parallel to FLUSH. Clarified scope of isochronous abort to just the offending command Modeled GROUP ID MASK field after the ACT field in the SDB FIS. Added reference to 13.6.6.1 for all other fields defined in the D/OWN subcommand.
4	2016-1-4	Member review. Changed document number from D208 to TPR076.

1 Introduction

This proposal discusses the details necessary for defining & implementing a Durable/Ordered Write Notification mechanism on an ATA device. The advantage to such a notification is having a method to communicate to the host the order that writes were destaged from a device's internal cache to the device's permanent media. This new feature, when used with a host that monitors the order of writes for data consistency; will guarantee data integrity and prevent data corruption in the case of power loss during drive operation while not adversely affecting performance.

This document provides a common understanding of the new concepts that Durable/Ordered Write Notification introduces, provides a common language to discuss Durable/Ordered Write Notification functionality, and proposes new commands and modifications to existing commands, to take advantage of such a device.

2 Description of Consistency Problem on Storage

An application layer in a host that depends on the writes it creates to be committed to the non-volatile media of a device in a specific order cannot rely on the order that the writes were received by the device to convey write order information. Currently there are no requirements or methods in the ATA specifications for devices to provide an environment that destages writes from internal cache in the order they arrived to the device as the order that the writes will be committed to permanent media. The destaging issue seen in the devices internal cache can be seen anywhere in the system that queuing or caching occurs.

Multithreaded hosts that have multiple IO generators that are scheduled in time slices (1) and supported by storage subsystems that are also multithreaded and queued and provide no guarantee in the order those IO operations occur (2). With the addition of NCQ, SATA provides another opportunity of IO reordering within the device before writes make it to a device's non-volatile media. Finally, a device with a buffer or volatile cache may commit to permanent media writes that the device has received in an order that is optimized for performance rather than the order that the writes were received in (3). These three opportunities for reordering are shown in figure 1 below.

The consequences of this write reordering is that applications such as file systems and databases that rely on writes to be finalized to non-volatile media in the order the writes were issued by the application so as to maintain a record of operations completed when performing journaling or logging will become inconsistent after a loss of power during write operations when, in today's environment, previous operations which have not yet been finalized to non volatile media are lost. Figure 1 shows File System meta data writes as black number on IO blocks to illustrate this.

Today, to compensate for the out of order environment an application's only option to ensure that all of its previous writes have completed before the next write may complete is to perform an application file operation flush, which causes all data of all previously completed writes to be forced out of the system and committed to the device's permanent media. However, this operation is slow and hurts performance. Consequently application file operation flushes often are not translated into an ATA FLUSH command leaving the window of opportunity for data inconsistency open.

Alternately an application can serialize all of its ordered writes throughout the entire host and mark each one as a Forced Unit Access (FUA) write ensuring that each write is committed to permanent media before the command is completed. Although this doesn't cause all writes buffered in a device to be committed to disk, the serialization from the application layer all the way down to the device's permanent media is also very slow and FUA is also ignored.

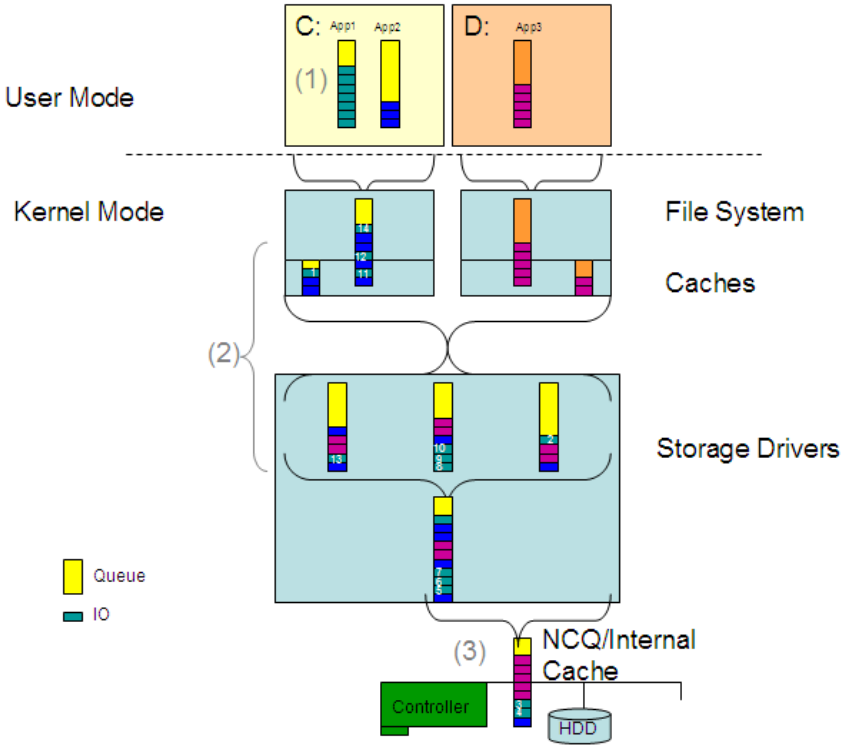

Figure 1: Contributors to Write Reordering

In order to avoid the performance penalties incurred by FLUSH and FUA, a solution is needed that allows a device to commit writes to non-volatile media with minimal impacts on performance. This proposal suggests the solution is a Write Barrier command implementation.

3 Description of Durable/Ordered Write Notification

The performance impact of the ATA FLUSH occurs for two reasons:

a) the ATA FLUSH disrupts the NCQ protocol as non-NCQ commands and NCQ commands cannot be mixed and ATA hardware FLUSH is a non-NCQ command. To send an ATA hardware FLUSH the NCQ work must become quiesced before the ATA hardware FLUSH may be sent. This action adds data not targeted by the application's file operation FLUSH to the write cache which further belabors the impending ATA hardware FLUSH; and

b) while the ATA FLUSH is outstanding no other commands can be sent or completed because the ATA hardware FLUSH is non-NCQ. The ATA hardware FLUSH cannot be completed until all of the data in the volatile write cache has been made durable.

Focusing on the needs of the application's file operation FLUSH, the simple requirements don't actually dictate an urgency as much as an order of operations. Consequently the application's file operation FLUSH can be performed without issuing an ATA hardware FLUSH at all as long as the data durability requirement is met before the notification requirement.

Consequently, all that is needed to perform a performant FLUSH is a mechanism by which:

a) the outstanding uncompleted application file operation WRITEs can be separated from the completed application file operation WRITEs at the time that the application file operation FLUSH is received; and

b) the device can notify when the set of data comprising the completed application file operation WRITEs have been made durable.

The intent of this proposal is to offer a definition for Write Cache Notification that ensures consistency of ordered writes while protecting the devices ability to make performance optimizations as a preferable alternative to the current behavior of relying on the durability of all order sensitive application data without using the ATA FLUSH and FUA commands.

4 Durable/Ordered Write Notification Feature Proposal

The primary focus of the proposed Durable/Ordered Write Notification feature is to enable the device to share write order information with the host, which is a new concept to the SATA standard. Consequently many of the concepts that will be used in discussion and in the remainder of this document are also new. This section establishes the goals and requirements for the Durable/Ordered Write Notification proposal, defines terminology for new Durable/Ordered Write Notification concepts, and then illustrates the terminology using scenarios that typify Durable/Ordered Write Notification usage.

4.1 Proposed Solution Requirements

4.1.1 Durable/Ordered Write Notification Feature Requirements

The purpose of Durable/Ordered Write Notification feature is to create a mechanism by which the host may discover write order information within the device. Requirements for the Durable/Ordered Write Notification proposal break down into the mechanism's ability to discover that the device supports the new functionality, the ability to deliver metadata mapped to targeted writes, the ability to discover when data has left the write cache, and the expected devices behavior when it receives the request for notification.

The Durable/Ordered Write Notification mechanism shall:

- Be discoverable and negotiable (e.g., enable/disable).
- Have both an NCQ command set component so it is not necessary to drain the queue to send a Durable/Ordered Write Notification command.
- Not modify the device's ability to re-order NCQ commands
- Continue to allow write re-ordering by the device

The best solution to improve the performance of the application's file operation FLUSH request has its dependency on the ATA hardware FLUSH removed in such a way that:

- ensures that all completed application's file operation WRITEs have been made durable without unnecessarily also making durable additional application's file operation WRITEs that were yet uncompleted at the time the application's file operation FLUSH was requested. This means that while NCQ is occurring and multiple ATA hardware WRITEs may be in process, these in-process ATA hardware WRITEs can be identified and excluded from set of data that must be made durable at the time of the application's file operation FLUSH request.
- is performed without disrupting any ATA hardware WRITEs including the quiescing of NCQ processing.
- doesn't monopolize the storage bus or the processing capability of the device while the set of data that is being made durable.

The device must:

- Be able to implement Durable/Ordered Write Notification without a significant negative performance impact when compared to similar models of SATA drive that does not implement Durable/Ordered Write Notification.
- Be able to implement Durable/Ordered Write Notification without a significant increase in the complexity of the firmware of the device or its testability.

5 Proposed Changes to SATA 3.3

The following SATA 3.3 sections require modification:

a) the WRITE FPDMA QUEUED command is modified to contain the Write Barrier fields necessary for conveying Write Barrier information; and
b) a new NCQ NON-DATA subcommand is defined and the NCQ NON-DATA log has been modified to be able to identify device support.

Proposed additions to Serial ATA Revision 3.3_CheckPrintOctober16_2015 text are marked in blue underline. Proposed deletions to Serial ATA Revision 3.3_CheckPrintOctober16_2015 text are marked in ~~red strikethrough~~. Black text is the original Serial ATA Revision 3.3_CheckPrintOctober16_2015 text. Section headers correspond to the section in Serial ATA Revision 3.3_CheckPrintOctober16_2015 into which the proposed text is to be inserted.

5.1 Modifications to WRITE FPDMA QUEUED

5.1.1 [13.6.5.1] WRITE FPDMA QUEUED

Register	7	6	5	4	3	2	1	0
FEATURES(7:0)	SECTOR COUNT(7:0)							
FEATURES(15:8)	SECTOR COUNT(15:8)							
COUNT(7:0)	TAG(4:0)					Reserved		
COUNT(15:8)	PRIO(1:0)		GROUP ID(5:0)~~Reserved~~					
LBA(7:0)	LBA(7:0)							
LBA(15:8)	LBA(15:8)							
LBA(23:16)	LBA(23:16)							
LBA(31:24)	LBA(31:24)							
LBA(39:32)	LBA(39:32)							
LBA(47:40)	LBA(47:40)							
ICC(7:0)	ICC(7:0)							
AUXILIARY(7:0)	Reserved							
AUXILIARY(15:8)	Reserved							
AUXILIARY(23:16)	HYBRID INFORMATION(7:0)							
AUXILIARY(31:24)	Reserved							
DEVICE(7:0)	FUA	1	0	0	Reserved			
COMMAND(7:0)	**61h**							

Figure 342 – WRITE FPDMA QUEUED command definition

GROUP ID — If the SUPPORTS DURABLE/ORDERED WRITE NOTIFICATION bit is set to one, the data transferred by this WRITE FPDMA QUEUED command is associated with the Group ID, while the data is in the device's write cache.

5.2 Modifications to NCQ NON-DATA subcommands

5.2.1 [13.6.6.2] NCQ NON-DATA subcommands

Table 107 – Subcommands for NCQ NON-DATA

SUBCOMMAND field	Description	Reference
0h	ABORT NCQ QUEUE subcommand	13.6.6.3
1h	DEADLINE HANDLING subcommand	13.6.6.4
2h	HYBRID DEMOTE BY SIZE subcommand	13.6.6.5
3h	HYBRID CHANGE BY LBA RANGE subcommand	13.6.6.6
4h	HYBRID CONTROL subcommand	13.6.6.7
5h	SET FEATURES subcommand	13.6.6.8
6h	ZERO EXT subcommand	13.6.6.9
7h	ZAC MANAGEMENT OUT subcommand	13.6.6.10
8h	DURABLE/ORDERED WRITE NOTIFICATION subcommand	13.6.6.11
9h~~8h~~..Fh	Reserved	

5.2.2 [13.7.5] NCQ NON-DATA Log (12h)

Dword	Bits	Description	Reference
8	Subcommand 8h		
	31:2	Reserved	
	1	SUPPORTS D/OW bit	13.7.5.16
	0	SUPPORTS DURABLE/ORDERED WRITE NOTIFICATION bit	13.7.5.17

Figure 388 – NCQ NON-DATA Log (12h) data structure definition

[13.7.5.16] SUPPORTS DURABLE/ORDERED WRITE NOTIFICATION bit

If the SUPPORTS DURABLE/ORDERED WRITE NOTIFICATION bit is set to one, then the device supports the DURABLE/ORDERED WRITE NOTIFICATION subcommand (see 13.6.6.11). If the Supports DURABLE/ORDERED WRITE NOTIFICATION bit is cleared to zero, then the device does not support the DURABLE/ORDERED WRITE NOTIFICATION subcommand.

[13.7.5.17] SUPPORTS D/OW bit

If the SUPPORTS D/OW bit is set to one, then the device supports the D/OW bit of the DURABLE/ORDERED WRITE NOTIFICATION subcommand. If the SUPPORTS D/OW bit is cleared to zero, then the device does not support the D/OW field of the DURABLE/ORDERED WRITE NOTIFICATION subcommand.

5.3 Additions to Priority

5.3.1 [13.6.2.4] Priority

The priority class is specified in the Priority (PRIO) field for READ FPDMA QUEUED commands, WRITE FPDMA QUEUED commands, RECEIVE FPDMA QUEUED commands, ~~and~~ SEND FPDMA QUEUED commands, and DURABLE/ORDERED WRITE NOTIFICATION subcommands. This bit may specify either the normal priority or high priority value. If a command is marked by the host as high priority, the device should attempt to provide better quality of service for the command. It is not required that devices process all high priority requests before satisfying normal priority requests.

5.4 New DURABLE/ORDERED WRITE NOTIFICATION subcommand

Everything in this section is new. It is in black text to improve readability.

5.4.1 [13.6.6.11] DURABLE/ORDERED WRITE NOTIFICATION subcommand (8h)

5.4.1.1 [13.6.6.11.1] DURABLE/ORDERED WRITE NOTIFICATION subcommand overview

The DURABLE/ORDERED WRITE NOTIFICATION subcommand provides all Group IDs to be used when processing the DURABLE/ORDERED WRITE NOTIFICATION subcommand. Support for this subcommand is indicated in the NCQ NON-DATA log (see 13.7.5.16).

If a DURABLE/ORDERED WRITE NOTIFICATION subcommand is marked by the host as high priority, the device should attempt to provide better quality of service for the command than for other NCQ commands that are not marked by the host as high priority.

WRITE FPDMA QUEUED commands outstanding at the same time as a DURABLE/ORDERED WRITE NOTIFICATION subcommand may have data associated with any of the Group IDs provided by this command and may add data associated with those Group IDs to the volatile cache and may delay the completion of this command. If the device indicates command acceptance of multiple DURABLE/ORDERED WRITE NOTIFICATION subcommands with the same values in the GROUP ID MASK field, then the device shall complete all DURABLE/ORDERED WRITE NOTIFICATION subcommands with the same values in the GROUP ID MASK field concurrently.

NOTE n- This command may take more than 30 seconds to complete.

Register	7	6	5	4	3	2	1	0
FEATURES(7:0)	D/OW	PRIO		R	8h			
FEATURES(15:8)	GROUP ID MASK (55:48)							
COUNT(7:0)	TAG (4:0)					RESERVED		
COUNT(15:8)	GROUP ID MASK (63:56)							
LBA(7:0)	GROUP ID MASK (7:0)							
LBA(15:8)	GROUP ID MASK (15:8)							
LBA(23:16)	GROUP ID MASK (23:16)							
LBA(31:24)	GROUP ID MASK (31:24)							
LBA(39:32)	GROUP ID MASK (39:32)							
LBA(47:40)	GROUP ID MASK (47:40)							
ICC(7:0)	RESERVED							
AUXILIARY(7:0)	RESERVED							
AUXILIARY(15:8)	RESERVED							
AUXILIARY(23:16)	RESERVED							
AUXILIARY(31:24)	RESERVED							

Figure TBD1 – DURABLE/ORDERED WRITE NOTIFICATION subcommand

Field Definitions

D/OW

If Supports D/OW is cleared to zero or the D/OW bit is cleared to zero, the device shall not indicate successful completion until all data received and stored in the device's cache associated with any of the Group IDs provided by this command has been flushed to the non-volatile media. If the volatile cache is disabled or no volatile cache is present, the device shall indicate command completion without error.

If Supports D/OW is set one and the D/OW bit is set to one, successful completion indicates that all data received and stored in the device's write cache associated with any of the Group IDs provided by this command shall be flushed to the non-volatile media before all data not yet received and stored in the device's write cache associated with any of the Group IDs provided by this command. If the volatile write cache is disabled or no volatile write cache is present, the device shall indicate command completion without error.

PRIO

The PRIO field value is assigned by the host based on the priority of the command issued. The device should complete high priority requests in a more timely fashion than normal priority and isochronous requests. If the PRIO field value is isochronous, then the device shall complete this command with command aborted (see Table 106).

R

Reserved.

GROUP ID MASK

The GROUP ID MASK field of the DURABLE/ORDERED WRITE NOTIFICATION subcommand communicates the data targeted in the device's write cache. Data associated with a GROUP ID was specified during the WRITE FPDMA QUEUED

command. The GROUP ID MASK field is bit-significant with each bit corresponding to a GROUP ID, where bit 0 corresponds to GROUP ID 0 and bit 63 corresponds to GROUP ID 63. More than one bit may be set to one.

All other fields as defined in 13.6.6.1.

5.4.1.2 [13.6.6.11.2] Success Outputs

If a Durable/Ordered Write Notification subcommand completes without error (see Figure TBD2), a Set Device Bits FIS shall be sent to the host. This Set Device Bits FIS may also indicate other completed commands.

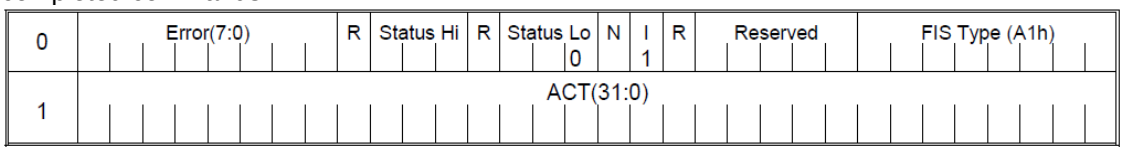

Figure TBD2 – DURABLE/ORDERED WRITE NOTIFICATION - successful completion

Field Definitions

Error The Error register shall contain 00h.

R Reserved, shall be cleared to zero.

Status As defined in 10.5.7. The ERR bit shall be cleared to zero indicating successful command completion. Bit 4 may be set to one.

I Interrupt bit. The interrupt bit shall be set to one.

ACT The ACT field of the Set Device Bits FIS communicates completion notification for each of up to 32 commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns. The device shall set to one the bit associated with the TAG value for the Deadline Handling command.

All other fields as defined in 10.5.7.

5.4.1.3 [13.6.6.11.3] Error Outputs

5.4.1.3.1 [13.6.6.11.3.1] Upon receipt of a command

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register Device to Host FIS (see Figure TBD3).

Register	7	6	5	4	3	2	1	0
ERROR(7:0)	ERROR(7:0)							
COUNT(7:0)	na							
COUNT(15:8)	na							
LBA(7:0)	na							
LBA(15:8)	na							
LBA(23:16)	na							
LBA(32:24)	na							
LBA(40:33)	na							
LBA(48:41)	na							
DEVICE(7:0)	na							
STATUS(7:0)	BSY	DRDY	DF	na	DRQ	na	na	ERR

Figure TBD3 – DURABLE/ORDERED WRITE NOTIFICATION - error on command receipt

Field Definitions

ERROR ATA error code for the failure condition of the failed command

BSY 0

DRDY 1

DF 0

DRQ 0

ERR 1

Status bit 4 may be set to one.

Following transmission of the Register Device to Host FIS, the device shall stop processing any outstanding or new commands until the Queued Error log (see 13.7.4) has been read before continuing to abort all outstanding commands. See 13.6.4.4 for more details.

5.4.1.3.2 [13.6.6.11.3.2] During execution of a command

If all commands have been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Set Device Bits FIS (see figure TBD4) to the host. All outstanding commands at the time of an error are aborted as part of the error response and may be re-issued as appropriate by the host. For any commands that have not completed or have completed with error, the device shall clear the corresponding ACT bits to zero in the Set Device Bits FIS.

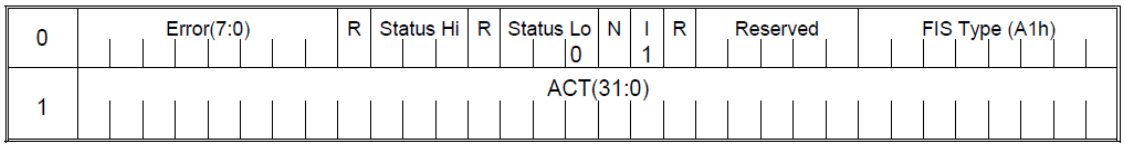

Figure TBD4 – DURABLE/ORDERED WRITE NOTIFICATION - error during processing

Field Definitions

Error The Error register shall contain the ATA error code.

R Reserved, shall be cleared to zero.

Status As defined in 10.5.7**.** The ERR bit shall be set to one indicating an NCQ error has occurred. Status bit 4 may be set to one.

I Interrupt bit. The interrupt bit shall be set to one.

ACT The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.

All other fields as defined in 10.5.7.

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the shadow Status register is cleared to zero, leaving the other Shadow Register Block Registers unchanged. If the device signals an error condition when the BSY bit in the shadow Status register is set to one, the device clears the BSY bit to zero with a Register Device to Host FIS which updates all registers in the Shadow Register Block.

Following transmission of the Set Device Bits FIS, the device shall stop processing any outstanding or new commands until the Queued Error log (see 13.7.4) has been read before continuing to abort all outstanding commands. See 13.6.4.4 for more details.

