Proposed Draft

Serial ATA International Organization

Version 3
6 July 2015

Serial ATA Revision 3.2 ECN086
Title : Clarifying NCQ Commands

This is an internal working document of the Serial ATA International Organization. As such, this is not a completed standard and has not been approved. The Serial ATA International Organization may modify the contents at any time. This document is made available for review and comment only.

Permission is granted to the Promoters, Contributors and Adopters of the Serial ATA International Organization to reproduce this document for the purposes of evolving the technical content for internal use only without further permission provided this notice is included. All other rights are reserved and may be covered by one or more Non Disclosure Agreements including the Serial ATA International Organization participant agreements. Any commercial or for-profit replication or republication is prohibited. Copyright © 2000 to 2015 Serial ATA International Organization. All rights reserved.

This Draft Specification is NOT the final version of the Specification and is subject to change without notice. A modified, final version of this Specification ("Final Specification") when approved by the Promoters will be made available for download at this Web Site: http://www.sata-io.org.

THIS DRAFT SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE. Except for the right to download for internal review, no license, express or implied, by estopple or otherwise, to any intellectual property rights is granted or intended hereunder.

THE PROMOTERS DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OF INFORMATION IN THIS DRAFT SPECIFICATION. THE PROMOTERS DO NOT WARRANT OR REPRESENT THAT SUCH USE WILL NOT INFRINGE SUCH RIGHTS.

THIS DOCUMENT IS AN INTERMEDIATE DRAFT FOR COMMENT ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.

* Other brands and names are the property of their respective owners.

Copyright © 2000 to 2015 Serial ATA International Organization. All rights reserved.

Author Information

Author Name	Company	Email address
Ralph Weber	Western Digital	Ralph.Weber@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	27 June 2015	Initial release.
1	29 June 2015	Revised as per requests from 29 June Digital call
2	6 July 2015	Revised as per requests from 6 July Digital call This revision was forwarded to the SATA-IO board for member review.
3	6 July 2015	Member review, D197 changed to ECN086

1 Introduction

1.1 Problem Statement

As the usefulness of the NCQ mechanism has proven itself, new NCQ commands have been added to the SATA specification. Sometimes the list of NCQ commands has not kept pace with these advances.

1.2 Solution Summary

To resolve this problem permanently:

1) a new glossary entry 'NCQ command' is proposed; and
2) all lists of NCQ commands are replaced with 'an NCQ command' or equivalent text as appropriate for the context.

The proposed glossary entry takes advantage of a new subclause that appears in the March 2015 check print for what will eventually become SATA Revision 3.3.

The other citations for changes are based on SATA Revision 3.2 Gold, with global application of the changes to SATA Revision 3.3 being left as an exercise for the editor.

2 Editorial Specification Changes

[editor note: Existing text is black. New text is marked as underlined in blue color. Material to be deleted ~~is red with strikethrough markings.~~]

2.1 4.1.1 Definitions and abbreviations

4.1.1.89a NCQ command
One of the mandatory or optional commands in the NCQ feature set (see 13.6.1).

4.1.1.90 NCQ Non-Streaming command

An NCQ Non-Streaming command is a command using the FPDMA QUEUED protocol that the PRIO is not set to 01b, Isochronous - deadline dependent priority.

4.1.1.91 NCQ Streaming command

An NCQ Streaming command is a command using the FPDMA QUEUED protocol that the PRIO is set to 01b, Isochronous - deadline dependent priority.

2.2 Global

[editor note: Changes such as those shown in this section should be applied globally. Using SATA 3.2 Gold as a source, as many changes as possible have been cataloged in this document. Cases where different NCQ commands have different behaviors specified are ignored by this proposal.]

10.5.7 Set Device Bits - Device to Host FIS

…

I – Interrupt Bit. This bit signals …
If the host is processing NCQ commands ~~native queued commands:~~
~~a) READ FPDMA QUEUED;~~
~~b) WRITE FPDMA QUEUED;~~
~~c) NCQ NON-DATA;~~
~~d) RECEIVE FPDMA QUEUED; or~~
~~e) SEND FPDMA QUEUED;~~
with the device, the interrupt pending state is entered regardless …

11.15 FPDMA QUEUED command protocol

The device command layer FPDMA Queued state machine is defined in Figure 332.

This class includes NCQ commands.~~:~~
~~a) READ FPDMA QUEUED;~~
~~b) WRITE FPDMA QUEUED;~~
~~c) NCQ NON-DATA;~~
~~d) RECEIVE FPDMA QUEUED; and~~
~~e) SEND FPDMA QUEUED.~~

…

12.1 FPDMA QUEUED command protocol overview

This high-level state machine describes the behavior of the host for the Native Command Queuing command protocol. The host behavior described by the state machine may be provided by host software or host hardware and the intent of the state machines is not to indicate any particular implementation.

This class includes NCQ commands.~~:~~
~~a) READ FPDMA QUEUED;~~
~~b) WRITE FPDMA QUEUED;~~
~~c) NCQ NON-DATA;~~
~~d) RECEIVE FPDMA QUEUED; and~~
~~e) SEND FPDMA QUEUED.~~

…

HFPI0: Idle, if in this state, if queuing is supported and enabled, the Command layer is Awaiting an NCQ command ~~a:~~
~~a) READ FPDMA QUEUED;~~
~~b) WRITE FPDMA QUEUED;~~
~~c) NCQ NON-DATA;~~
~~d) RECEIVE FPDMA QUEUED; or~~
~~e) SEND FPDMA QUEUED~~:
a) ~~A)~~ command from the higher level protocol;
b) ~~B)~~ awaiting an interrupt from the Device indicating completion of previously queued commands; or
c) ~~C)~~ waiting for a TAG location to become available for a command waiting in the command queue.

Transition HFPI0:1, if an NCQ command ~~a READ FPDMA QUEUED, WRITE FPDMA QUEUED, NCQ NON-DATA, RECEIVE FPDMA QUEUED, or SEND FPDMA QUEUED command~~ is pending that has not had a TAG value assigned to it and there is a free TAG location available for assignment then a transition shall be made to the HFPDMAQ2: PresetACTBit state.

Transition HFPI0:2, if ~~a command~~ an NCQ command with assigned TAG value is awaiting issue to the device and BSY bit cleared to zero and the interface is not in the First-party DMA Data Phase, then a transition shall be made to the HFPDMAQ3: IssueCommand state.

Transition HFPI0:3, if an interrupt is received from the device, indicating status is available for a previously queued ~~command~~ NCQ command, it shall transition to the HFPDMAQ4: DeviceINT state.

Transition HFPI0:4, if the queuing is supported and enabled, and the Command layer is awaiting a free TAG, a new ~~command~~ NCQ command, or an interrupt for a previously queued command, it shall transition to the HFPI0: Idle state.

HFPDMAQ1: AddCommandToQueue, the Command layer enters this state if it has received an NCQ command ~~a READ FPDMA QUEUED, WRITE FPDMA QUEUED, NCQ NON-DATA, RECEIVE FPDMA QUEUED, or SEND FPDMA QUEUED command~~ from the higher level protocol, and adds it to the internal host command queue.

Transition HFPDMAQ1:1: After the Command Layer has added the ~~command~~ NCQ command to the internal host command queue, it shall transition to the HFPI0: Idle state.

HFPDMAQ2: PresetACTBit, if in this state, the Command layer assigns a free TAG value to the previously queued ~~command~~ NCQ command and writes the SActive register with the bit position corresponding to the assigned TAG value.

Transition HFPDMAQ2:1: After the Command layer has assigned a TAG value and written the corresponding bit to the SActive register, it shall transition to the HFPI0: Idle state.

HFPDMAQ3: IssueCommand, if in this state, the Command layer attempts to issue ~~a command~~ an NCQ command with preassigned TAG to the device by transmitting a Register Host to Device FIS with the new ~~command~~ NCQ command and assigned TAG value if the interface state permits it.

Transition HFPDMAQ3:1: After the Command layer has transmitted the Register Host to Device FIS, it shall mark the corresponding ~~command~~ NCQ command as issued and transition to the HFPI0: Idle state.

Transition HFPDMAQ3:2: After the Command layer has deferred transmission of the Register Host to Device FIS due to the interface state not permitting it to be delivered, it shall transition to the HFPI0: Idle state. The corresponding ~~command~~ NCQ command is still considered as not having been issued.

HFPDMAQ4: DeviceINT, if in this state, the Command layer reads the Device Status register to reset the pending interrupt flag and save the value as SavedStatus.

Transition HFPDMAQ4:1: After the Command layer has read the Device Status register, it shall transition to the HFPDMAQ5: CompleteRequests1 state.

HFPDMAQ5: CompleteRequests1, if in this state, the Command layer compares the SActive register with the SavedStatus SActive register value that resulted from the last interrupt to identify completed ~~commands~~ NCQ commands.

Transition HFPDMAQ5:1, if the SActive comparison indicates one or more ~~commands~~ NCQ commands have completed, it shall transition to the HFPDMAQ6: CompleteRequests2 state.

Transition HFPDMAQ5:2, if the SActive comparison indicates no ~~commands~~ NCQ commands have completed, it shall transition to the HFPDMAQ7: CompleteRequests3 state.

HFPDMAQ6: CompleteRequests2, if in this state, the Command layer retires commands in its internal host command queue that are associated with TAG values corresponding to newly cleared bits in the SActive register and updates the stored SActive register with the new value.

Transition HFPDMAQ6:1: After updating the stored SActive value, it shall transition to the HFPDMAQ7: CompleteRequests3 state.

HFPDMAQ7: CompleteRequests3, if in this state, the Application layer tests the ERR bit in the SavedStatus value to determine whether the queue should be maintained or reset.

Transition HFPDMAQ7:1, if the SavedStatus value ERR bit cleared to zero, the queue is maintained and it shall transition to the HFPI0: Idle state.

Transition HFPDMAQ7:2, if the SavedStatus value ERR bit set to one, an error has been reported by the Device and the Command layer shall transition to the HFPDMAQ8: ResetQueue state.

HFPDMAQ8: ResetQueue, if in this state, the Application layer issues a command to read the Queued Error Log (see 13.7.4 and 13.7) to the device.

Transition HFPDMAQ8:1: After a READ LOG EXT command has been accepted, it shall transition to the HFPDMAQ9: CleanupACK state.

Transition HFPDMAQ8:2: After a READ LOG DMA EXT command has been accepted, it shall transition to the HFPDMAQ12: RetrieveRequest_SenseDMA .

Transition HFPDMAQ8:3, if the command was not accepted, the host shall transition to the HPFDMAQ8: ResetQueue state.

HFPDMAQ9: CleanupACK, if in this state, the Command layer tests the Device for DRQ bit set to one and BSY bit cleared to zero in preparation for a PIO data FIS transfer.

Transition HFPDMAQ9:1, if DRQ bit is set to one and BSY bit is cleared to zero, it shall transition to the HFPDMAQ10: ReceiveRequestSense state.

Transition HFPDMAQ9:2, if DRQ bit is cleared to zero or BSY bit is set to one, it shall transition to the HFPDMAQ9: CleanupACK state.

HFPDMAQ10: RetrieveRequest_Sense, if in this state, the Command layer completes the PIO Data FIS that retrieves the Queued Error Log contents.

Transition HFPDMAQ10:1: After the completion of the PIO Data FIS, it shall transition to the HFPDMAQ11: ErrorFlush state.

HFPDMAQ11: ErrorFlush, if in this state, the Command layer retires the failed queued command with the error status set to the error condition reported by the device. It flushes all allocated ~~native queued command~~ NCQ command tags, and flushes pending ~~native~~ ~~commands~~ NCQ commands from the host command queue with system-specific error condition or re-issue pending NCQ commands. ~~queued commands~~

Transition HFPDMAQ11:1: After the error flush actions have been completed, it shall transition to the HFPI0: Idle state.

HFPDMAQ12: RetrieveRequest_SenseDMA, this state is entered if the device has the data ready to transfer a data FIS to the host containing the Queued Error Log contents.

If in this state, the device shall request that the Transport layer transmit a data FIS containing the data. The device command layer shall request a Data FIS size of no more than 2 048 Dwords.

Transition HFPDMAQ12:1, if the FIS has been transmitted, the device shall transition to the HFPDMAQ13: SendStatus state.

HFPDMAQ13: SendStatus, this state is entered if the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

If in this state, the device shall request that the Transport layer transmit a Register Device to Host FIS with register content as described in the command description in the ACS-3 standard and the Interrupt bit set to one.

Transition HFPDMAQ13:1, if the FIS has been transmitted, the device shall transition to the HFPDMAQ11: ErrorFlush state.

…

13.6.2.4 Priority

…

The priority class is specified in the PRIO bit for ~~NCQ commands (i.e.,~~ READ FPDMA QUEUED commands, WRITE FPDMA QUEUED commands, RECEIVE FPDMA QUEUED commands, and SEND FPDMA QUEUED commands~~)~~. 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.

…

13.6.3 Intermixing ~~Non-Native Queued Commands and Native Queued~~ Non-NCQ commands and NCQ c~~C~~ommands

13.6.3.1 ~~Intermixing Non-Native Queued Commands and Native Queued Commands scope~~

Editor's note: ECN080 deleted 13.6.3.1 section heading.

Non-NCQ commands are all commands other than NCQ commands.

The host shall not issue a ~~non-native queued~~ non-NCQ command while ~~a native queued~~ an NCQ command is outstanding. Upon receiving a ~~non-native queued~~ non-NCQ command while ~~a native queued~~ an NCQ command is outstanding, the device shall signal the error condition to the host by transmitting a Register Device to Host FIS with the ERR and ABRT bits set to one and the BSY bit cleared to zero in the Status field of the FIS and halt command processing as defined in 13.6.4.4 except as noted below.

~~Non-native queued commands include all commands other than:~~

~~a) READ FPDMA QUEUED (see 13.6.4);~~
~~b) WRITE FPDMA QUEUED (see 13.6.5);~~
~~c) NCQ NON-DATA (see 13.6.6);~~
~~d) RECEIVE FPDMA QUEUED (see 13.6.7); and~~
~~e) SEND FPDMA QUEUED (see 13.6.8).~~

Reception of a non-NCQ command to read the Queued Error Log (see 13.7) after an error has occurred shall cause any outstanding ~~Serial ATA native queued~~ NCQ commands to be aborted, and the device shall perform necessary state cleanup to return to a state with no commands pending. The device shall clear all bits in the SActive register by transmitting a Set Device Bits FIS to the host with all the bits in the SActive field set to one (i.e., FFFF FFFFh). After reading the Queued Error Log, the device shall be prepared to process subsequently issued ~~queued~~ NCQ commands regardless of any previous errors on an ~~queued~~ NCQ command.

In the case that a non-NCQ command to read the Queued Error Log is issued while ~~a native queued~~ an NCQ command is outstanding and no error was previously reported by the device, then the device shall signal an error condition. The receipt of this command if no error is outstanding shall be handled as any other ~~non-native queued~~ non-NCQ command if ~~a native queued~~ an NCQ command is outstanding. In this case, a subsequent non-NCQ command to read the Queued Error Log is required to recover from the error.

13.6.3.2 ~~Intermixing Non-Native Queued Commands and Native Queued Commands overview~~

Editor's note: ECN080 deleted 13.6.3.2 section.