Proposed Draft

Serial ATA International Organization

Version 3
June 23, 2014

Serial ATA Revision 3.2 ECN080
Title : NCQ Feature Set Clarification

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.

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.

Author Information

Author Name	Company	Email address
Jeff Wiles	HP	jeff.wiles@hp.com

Workgroup Chair Information

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

Document History

Version	Date	Comments
0	May 28, 2014	Initial release.
1	June 5, 2014	WG editorial inputs Move mandatory/optional command description to 13.6.1
2	June 9, 2014	Move IDENTIFY DEVICE and QEL references under 13.6.1 overview in an ordered list.
3	June 23, 2014	Change D185 to ECN080. Member review.

1 Introduction

1.1 Problem Statement

13.6.2 heading does not include the words “feature set” to align with 13.7.9.2.4 heading.

13.6.3.2 Overview does not specify mandatory vs. optional NCQ commands.

13.6.3.2 Overview content is misplaced; should be located in 13.6.1 instead of 13.6.3.2

Table 100, part 2 of 4, Identify Device Word 76 bit 8 does not reference the NCQ Feature Set.

Broken cross reference located in DMA Buffer Offset paragraph in section 13.6.2.2.

1.2 Solution Summary

Changed the heading in 13.6.2 to include the words “feature set”.

Changed the description of NCQ commands in 13.6.3.2 to show which are mandatory and which are optional.

Moved mandatory/optional description from 13.6.3.2 to 13.6.1 and removed section 13.6.3.1.

Revised the description of Identify Device Word 76 bit 8 to show NCQ feature set support.

Fixed broken cross reference in section 13.6.2.2 (correctly shows reference to section 13.3).

1.3 Background (optional)

2 Technical Specification Changes

2.1 <Title of section being changed>

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

13.6 Native Command Queuing (NCQ) feature set (optional)

13.6.1 Native Command Queuing feature set overview

This section defines a simple and streamlined command queuing model for Serial ATA.

Devices that support the NCQ feature set shall:

a) report support for the NCQ feature set (i.e., IDENTIFY DEVICE data Word 76 bit 8 is set to one);

b) report support for the general purpose logging feature set (i.e., IDENTIFY DEVICE data Word 84 bit 5 is set to one); and

c) implement the Queued Error Log (see 13.7.4).

The following commands are mandatory for devices that implement the NCQ feature set:

a) READ FPDMA QUEUED; and
b) WRITE FPDMA QUEUED.

The following commands are optional for devices that implement the NCQ feature set:

a) NCQ NON-DATA;
b) RECEIVE FPDMA QUEUED; and
c) SEND FPDMA QUEUED.

NCQ NON-DATA is the only NCQ command that is performed with no data transfer.

READ FPDMA QUEUED and WRITE FPDMA QUEUED commands have transfer sizes of logical sector size multiples.

RECEIVE FPDMA QUEUED and SEND FPDMA QUEUED commands have transfer sizes of 512 byte multiples.

READ FPDMA QUEUED and RECEIVE FPDMA QUEUED commands transfer data from the device to the host.

WRITE FPDMA QUEUED and SEND FPDMA QUEUED commands transfer data from the host to the device.

The native queuing definition utilizes the reserved 32 bit field in the Set Device Bits FIS to convey the pending status for each of up to 32 outstanding commands. The BSY bit in the Status register conveys only the device’s readiness to receive another command, and does not convey the completion status of queued commands. Upon receipt of a new command, the device clears the BSY bit to zero before proceeding to process received commands. The 32 protocol specific bits in the Set Device Bits FIS are handled as a 32-element array of active command bits (referred to as ACT bits), one for each possible outstanding command, and the array is bit significant such that bit “n” in the array corresponds to the pending status of the command with tag “n.”

Data returned by the device (or transferred to the device) for queued commands use the First Party DMA mechanism to cause the host controller to select the appropriate destination/source memory buffer for the transfer. The memory handle used for the buffer selection is the same as the tag associated with the command. For traditional desktop host controllers, the handle may be used to index into a vector of pointers to pre-constructed scatter/gather lists (often referred to as physical region descriptor tables or simply Physical Region Descriptor (PRD) tables) in order to establish the proper context in the host’s DMA engine. The First-party DMA Data Phase is defined as the period from reception of a DMA Setup FIS until either the associated transfer count is exhausted or the ERR bit in the shadow Status register is set. During this period the host may not issue new commands to the device nor may the device signal new command completions to the host.

Status is returned by updating the 32-element bit array in the Set Device Bits FIS for successful completions. For failed commands, the device halts processing commands allowing host software or controller firmware to intervene and resolve the source of the failure, by using the general purpose logging feature set, before processing is again explicitly restarted.

~~Devices supporting Native Command Queuing shall implement, and report support for, the general purpose logging feature set as defined in ACS-3. In addition, the device shall implement the Queued Error Log.~~

13.6.2 Native Command Queuing (NCQ) feature set ~~D~~definition

13.6.2.1 Command issue mechanism

The Serial ATA transmission protocol is sensitive to the state of the BSY bit in the Shadow Status register that provides write protection to the shared Shadow Command Block registers. Since the Shadow Command Block registers may be safely written only if the BSY bit is cleared to zero, the BSY bit conventions defined in the Transport layer shall be adhered to, and issuing a new command shall only be attempted if the BSY bit is cleared to zero. If the BSY bit in the Shadow Status register is cleared to zero, another command may be issued to the device.

The state of the BSY bit in the Shadow Status register shall be checked prior to attempting to issue a new queued command. If the BSY bit is set to one, issuing the next command shall be deferred until the BSY bit is cleared to zero. It is desirable to minimize such command issue deferrals, so devices should clear the BSY bit to zero in a timely manner. Host controllers may have internal designs that mitigate the need for host software to block on the state of the BSY bit.

The native queuing commands include a tag value that identifies the command. The tag value is in the range 0 ~~through~~to 31 inclusive, and is conveyed in the Register Host to Device FIS if the command is issued. For devices that report a value less than 31 in their IDENTIFY DEVICE data Word 75, the host shall issue only unique tag values that are less than or equal to the value reported.

Upon issuing a new native queued command, the bit in the SActive register corresponding to the tag value of the command being issued shall be set to one by the HBA prior to the command being transmitted to the device. As described in 14.2.5 the SActive register and the access conventions for it.

Upon accepting the command, the device shall clear the BSY bit to zero if it is prepared to receive another command by transmitting a Register Device to Host FIS with the BSY bit cleared to zero in the Status field of the FIS, and the Interrupt bit cleared to zero.

13.6.2.2 Data delivery mechanism

The First-party DMA mechanism is used by the device to transmit (or receive) data for an arbitrary queued command. The command's tag value shall also be the DMA Buffer Identifier used to uniquely identify the source/destination memory buffer for the transfer.

The DMA Setup FIS is used by the device to select the proper transfer buffer prior to each data transfer. Only a single DMA Setup FIS is required at the beginning of each transfer and if the transfer spans multiple Data FISes a new DMA Setup FIS is not required before each Data FIS. Serial ATA host controller hardware shall account for the DMA Setup FIS buffer identifier being a value between 0 and 31 and the host controller shall select the proper transfer buffer based on such an index.

For data transfers from the host to the device, an optimization to the First-party DMA mechanism is included to eliminate one transaction by allowing the requested data to immediately be transmitted to the device following such a request without the need for a subsequent DMA Activate FIS for starting the flow of data. This optimization to the First-party DMA mechanism as defined in 10.5.9.4.2.

If non-zero buffer offsets in the DMA Setup FIS are not enabled (see 13.3.2) or not supported (see 13.2.2), the data transfer for a command shall be satisfied to completion following a DMA Setup FIS before data transfer for a different command may be started. Host controllers are not required to preserve DMA engine context upon receipt of a new DMA Setup FIS, and if non-zero buffer offsets are not enabled or not supported, a device is unable to resume data transfer for a previously abandoned context at the point where it left off.

If the host controller hardware supports non-zero buffer offsets in the DMA Setup FIS and use of non-zero offsets is enabled, and if guaranteed in-order data delivery is either not supported by the device (see 13.2.2) or is disabled (see 13.3.5), the device may return (or receive) data for a given command out of order (i.e., returning data for the last half of the command first). In this case the device may also interleave partial data delivery for multiple commands provided the device keeps track of the appropriate buffer offsets.

> NOTE 60 - An example of interleaving partial data delivery for multiple command is data for the first half of command 0 may be delivered followed by data for the first half of command 1 followed by the remaining data for command 0.

By default use of non-zero buffer offsets is disabled. See 13.3.2 for information on enabling non-zero buffer offsets for the DMA Setup FIS.

If the host controller hardware supports non-zero buffer offsets in the DMA Setup FIS and use of non-zero offsets is enabled, and if the device supports guaranteed in-order data delivery and guaranteed in-order data delivery is enabled, then the device may use multiple DMA Setup FISes to satisfy a particular I/O process. If multiple DMA Setup FISes are used, then the data shall be delivered in-order, starting at the first LBA. In this case the device may not interleave partial data delivery for either individual or multiple commands.

> NOTE 61 - Data for the first half of a command may be delivered using one DMA Setup FIS and one or more subsequent Data FISes, followed by the remaining data for that command, delivered using a second DMA Setup FIS and one or more subsequent Data FISes.

Non-zero buffer offsets are used as in the more general out-of-order data delivery case described above. By default use of guaranteed in-order data delivery is disabled.

For selecting the memory buffer for data transfers, the DMA Setup FIS is issued by the device. The DMA Setup FIS fields are defined in Figure 335 (see 10.5.9).

0	**Reserved (0)**	**Reserved (0)**	**A**	**I**	**D**	**Reserved (0)**	**FIS Type (41h)**
1	**0h**						**TAG**
2	**0h**						
3	**Reserved (0)**						
4	**DMA Buffer Offset**						
5	**DMA Transfer Count**						
6	**Reserved (0)**						

Figure 335 – DMA Setup FIS definition for memory buffer selection

Field Definitions

FIS Type As defined in 10.5.9.

D As defined in 10.5.9. Since the DMA Setup FIS is only issued by the device for the queuing model defined here, the value in the field is defined as 1 = device to host transfer (write to host memory), 0 = host to device transfer (read from host memory).

A As defined in 10.5.9, including additional details according to 10.5.9.4.2. For DMA Setup with transfer direction from device to host, this bit shall be zero.

TAG This field is used to identify the DMA buffer region in host memory to select for the data transfer. The low order 5 bits of the DMA Buffer Identifier Low field shall be set to the TAG value corresponding to the command TAG that data is being transferred. The remaining bits of the DMA Buffer Identifier Low/High shall be cleared to zero. The 64 bit Buffer Identifier field defined in the DMA Setup FIS according to 10.5.9 is used to convey a TAG value that occupies the five least-significant bits of the field.

DMA Buffer Offset

As defined in section 10.5.9. The device may specify/indicate a non-zero value in this field only if the host indicates support for it through the SET FEATURES mechanism as defined in section 13.3~~1~~. Data is transferred to/from sequentially increasing logical addresses starting at the specified offset in the specified buffer.

DMA Transfer Count

As defined in 10.5.9. The value shall accurately reflect the length of the data transfer to follow. According to 10.5.12.3 for special considerations if the transfer count is for an odd number of Words. Devices shall not set this field to 0h; a value of 0h for this field is illegal and results in indeterminate behavior.

I Interrupt Native Command Queuing does not make use of an interrupt following the data transfer phase (after the transfer count is exhausted). The Interrupt bit shall be cleared to zero.

R/Reserved All reserved fields shall be cleared to zero.

13.6.2.3 Status return mechanism

For maximum efficiency, the status return mechanism is not interlocked (does not include a handshake) while at the same time ensuring no status notifications are lost or overwritten (i.e., status notifications are race-free). The status return mechanism relies on an array of ACT bits – one ACT bit to convey the active status for each of the 32 possible outstanding commands, resulting in a 32 bit ACT status field. The 32 bit reserved field in the Set Device Bits FIS as defined in 10.5.7 is defined as the SActive field and is used to convey command completion information for updating the ACT bit array. The zero bit position in the 32 bit field corresponds to the ACT bit for the command with tag value of zero. Host software shall check the SActive register (containing the ACT bit array) if checking status in order to determine that command(s) have completed since the last time the host processed a command completion. It is possible for multiple commands to indicate completion by the time the host checks the status due to the software latencies in the host (i.e., by the time the host responds to one completion notification, another command may also have completed). Only successfully completed commands indicate their status using this mechanism – failed commands use an additional mechanism described in 13.6.4.3.1 and 13.6.5.3.2 to convey error information as well as the affected command tag. The Queued Error Log is used to convey additional queued command error information as outlined in 13.7.4 and 13.7.

13.6.2.4 Priority

Host knowledge of I/O priority may be transmitted to the device as part of the command. There are two priority values for native command queuing (NCQ) commands, normal and high. If the host marks an NCQ command as high priority, the host is requesting a better quality of service for that command than commands issued with normal priority.

The classes are forms of soft priority. The device may choose to complete a normal priority command before an outstanding high priority command, although preference should be given to the high priority commands.

EXAMPLE - One example where a normal priority command may be completed before a high priority command is if the normal priority command is a cache hit, whereas the high priority command requires access of the device media.

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

The device should complete high priority requests in a more timely fashion than normal and isochronous requests. The device should complete isochronous request prior to its associated deadline.

The device should complete isochronous request prior to its associated deadline (see Table 105).

Table 105 – Priority

Prio(1:0)	Description
00b	Normal Priority
01b	Isochronous – deadline dependent priority
10b	High priority
11b	Reserved

13.6.2.5 Unload

If using Native Command Queuing in a laptop environment, the host needs to be able to park the head of a device with rotating media due to excessive movement (e.g., the laptop being dropped). This section defines a mechanism that the host may use to park the heads if NCQ commands are outstanding in the device. The typical time for completion of the unload operation is defined in ATA/ATAPI-7 clause 6.20.10.

If NCQ commands are outstanding, the device is able to accept the IDLE IMMEDIATE command with the Unload Feature as defined in ACS-3.

Upon reception of this command with the Unload Feature specified, the device shall:

1) unload/park the heads, if any,
2) immediately; and
3) respond to the host with a Register Device to Host FIS with the ERR bit set to one in the Status register since this is a non-queued command.

If the host receives the error indication, it should proceed to read the Queued Error Log (see 13.7.4 and 13.7). In the Queued Error Log, the device shall indicate whether the error was due to receiving an UNLOAD and whether the UNLOAD was processed. The device shall not load the heads to the media if reading the Queued Error Log.

The Queued Error Log indicates whether the device has accepted the Unload and is in the process of processing the command. To get a definitive indication of Unload completion (and success), the IDLE IMMEDIATE command with the Unload Feature needs to be issued again after the Queued Error Log has been read. After the Queued Error Log has been read, there are no NCQ commands outstanding and the NCQ error is cleared to zero. A sequent IDLE IMMEDIATE command with the Unload Feature shall be processed normally and a successful status shall be returned if the unload process completes successfully.

There may be a delay in issuing the IDLE IMMEDIATE command with the Unload feature to the device if the device is currently performing a data transfer for a previously issued NCQ command. If the device happens to be processing extensive data error recovery procedures, this delay may be longer than acceptable. However, this same issue may occur if a non-queued data command is outstanding and the device is performing error recovery procedures.

13.6.3 Intermixing Non-Native Queued Commands and Native Queued Commands

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

The host shall not issue a non-native queued command while a native queued command is outstanding. Upon receiving a non-native queued command while a native queued 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 command to read the Queued Error Log (see 13.7) after an error has occurred shall cause any outstanding Serial ATA native queued 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 commands regardless of any previous errors on a queued command.

In the case that a command to read the Queued Error Log is issued while a native queued 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 command if a native queued command is outstanding. In this case, a subsequent 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~~

~~NCQ commands consist of the following:~~

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

~~NCQ NON-DATA is the only NCQ command that is performed with no data transfer.~~

~~READ FPDMA QUEUED and WRITE FPDMA QUEUED commands have transfer sizes of logical sector size multiples.~~

~~RECEIVE FPDMA QUEUED and SEND FPDMA QUEUED commands have transfer sizes of 512 byte multiples.~~

~~READ FPDMA QUEUED and RECEIVE FPDMA QUEUED commands transfer data from the device to the host.~~

~~WRITE FPDMA QUEUED and SEND FPDMA QUEUED commands transfer data from the host to the device.~~

~~NCQ NON-DATA, RECEIVE FPDMA QUEUED and SEND FPDMA QUEUED contain subcommands.~~

Table 100 – IDENTIFY DEVICE information (part 2 of 4)

Word	O/M	F/V		
68	M		Minimum PIO transfer cycle time with IORDY flow control	
		F	15..0	Cycle time in nanoseconds
69..74			Set as indicated in ACS-3	
75	O		Queue depth	
		R	15..5	Reserved
		F	4..0	Maximum queue depth – 1
76	O		Serial ATA capabilities	
		F	15	Supports READ LOG DMA EXT as equivalent to READ LOG EXT
		F	14	Supports Device Automatic Partial to Slumber transitions
		F	13	Supports Host Automatic Partial to Slumber transitions
		F	12	Supports Native Command Queuing priority information
		F	11	Supports Unload while NCQ commands outstanding
		F	10	Supports Phy event counters
		F	9	Supports receipt of host-initiated interface power management requests
		F	8	~~Supports Native Command Queuing~~ Supports NCQ feature set
		R	7..4	Reserved for future Serial ATA signaling speed grades
		F	3	Supports Serial ATA Gen3 signaling speed (6.0 Gbps)
		F	2	Supports Serial ATA Gen2 signaling speed (3.0 Gbps)
		F	1	Supports Serial ATA Gen1 signaling speed (1.5 Gbps)
		F	0	Shall be cleared to zero
77	O			Serial ATA Additional capabilities
		R	15..~~8~~9	Reserved
		F	8	Power Disable feature always enabled
		F	7	DevSleep_to_ReducedPwrState
		F	6	Supports RECEIVE FPDMA QUEUED and SEND FPDMA QUEUED commands
		F	5	Supports NCQ NON-DATA Command
		F	4	Supports NCQ Streaming
		V	3..1	Coded value indicating current negotiated Serial ATA signal speed
		F	0	Shall be cleared to zero

Key:
M = Support of the Word is mandatory.
O = Support of the Word is optional.
F = the content of the bit, field, or Word is fixed and does not change. For removable media devices, these values may change if media is removed or changed.
V = the contents of the bit, field, or Word is variable and may change depending on the state of the device or the commands processed by the device.
R = the content of the bit, field, or Word is reserved and shall be zero.

Editor's note: TPR056 Enable new Power Disable feature on standard SATA connector P3 added Word 77 bit 8.