

**Proposed  
Draft**

**Serial ATA  
International Organization**

**Version 7  
30 March 2015**

---

**Serial ATA Revision 3.2 Technical Proposal 066  
Title : Reporting Current Write Pointer for NCQ  
Errors in Write Pointer Zones**

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 estoppel 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
Neil Wanamaker	PMC-Sierra	<a href="mailto:Neil.wanamaker@pmcs.com">Neil.wanamaker@pmcs.com</a>

## Workgroup Chair Information

Workgroup (Phy, Digital, etc...)	Chairperson Name	Email address
Digital	James C. Hatfield	<a href="mailto:James.c.hatfield@seagate.com">James.c.hatfield@seagate.com</a>

## Document History

Version	Date	Comments
0	11/24/2014	Initial release.
1	11/24/2014	Change formatting to match template
2	12/15/2014	Addition of array of validity indicators and 32 write pointers
3	1/26/2015	Response to comments from 12/15 and 1/5 Digital WG
4	1/26/2015	Approved version with editorial changes per 1/26 Digital WG.
5	1/26/2015	Member review, changed TPR D191 to TPR066.
6	2/2/2015	Fixed write pointer checksum.
7	3/30/2015	Inclusion of changes from 3/16 Digital WG, specifically clarification of write pointer checksum definition, and when write pointer valid is to be set.

## 1 Introduction

Historically, in the Queued error log (10h), the LBA[47:0] contains the LBA in error, rather than a write pointer. As the Queued error log currently has sufficient unallocated space, we are not restricted to returning 48 bits of LBA information. It is proposed that we continue to store the error LBA in the current location in the Queued error log, and store the current write pointer in a currently-unused location.

This proposal adds a second page to the log at log address 10h for a ZAC device.

[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 Technical Specification Changes

### 13.7.3 General purpose log directory (00h)

Devices supporting the Queued Error Log (see 13.7) reflect this support in the General Purpose Log Directory (see Table 113) log (00h) by having the value 1 at offset 020h and the value 0 at offset 021h of that log to indicate existence of a log at address 10h of at least 1 page in length. For a ZAC device (see TPR057), this log is 2 pages in length.

### 13.7.4 Queued error log (10h)

Byte	7	6	5	4	3	2	1	0
0	NQ	UNL	R	TAG				

1	Reserved
2	Status(7:0)
3	Error(7:0)
4	LBA(7:0)
5	LBA(15:8)
6	LBA(23:16)
7	Device(7:0)
8	LBA(31:24)
9	LBA(39:32)
10	LBA(47:40)
11	Reserved
12	Count(7:0)
13	Count(15:8)
14	Sense Key
15	Additional Sense Code
16	Additional Sense Code Qualifier
17	Final LBA in Error(7:0)
18	Final LBA in Error(15:8)
19	Final LBA in Error(23:16)
20	Final LBA in Error(31:24)
21	Final LBA in Error(39:32)
22	Final LBA in Error(47:40)
23	Reserved
...	
255	
256	Vendor Specific
...	
510	
511	Data Structure Checksum
<a href="#">512</a>	<a href="#">Write Pointer Valid(7:0)</a>
<a href="#">513</a>	<a href="#">Write Pointer Valid(15:8)</a>
<a href="#">514</a>	<a href="#">Write Pointer Valid(23:16)</a>
<a href="#">515</a>	<a href="#">Write Pointer Valid(31:24)</a>
<a href="#">516</a>	<a href="#">Write Pointer[0](7:0)</a>
<a href="#">517</a>	<a href="#">Write Pointer[0](15:8)</a>
<a href="#">518</a>	<a href="#">Write Pointer[0](23:16)</a>
<a href="#">519</a>	<a href="#">Write Pointer[0](31:24)</a>
<a href="#">520</a>	<a href="#">Write Pointer[0](39:32)</a>
<a href="#">521</a>	<a href="#">Write Pointer[0](47:40)</a>
...	
<a href="#">702</a>	<a href="#">Write Pointer[31](7:0)</a>
<a href="#">703</a>	<a href="#">Write Pointer[31](15:8)</a>
<a href="#">704</a>	<a href="#">Write Pointer[31](23:16)</a>
<a href="#">705</a>	<a href="#">Write Pointer[31](31:24)</a>
<a href="#">706</a>	<a href="#">Write Pointer[31](39:32)</a>
<a href="#">707</a>	<a href="#">Write Pointer[31](47:40)</a>
<a href="#">708</a>	<a href="#">Reserved</a>
...	
<a href="#">1 022</a>	

**Figure 381 – Queued Error Log data structure definition**Write Pointer Valid

Bit n is set to 1 if Write Pointer[n] contains a valid write pointer corresponding to the command with tag n. Bit n is cleared to 0 if Write Pointer[n] does not contain a valid write pointer. The Write Pointer Valid bits are cleared to zero unless the device is in state DFPDMAQ13: WaitforClear. For any outstanding media access commands, the device shall set the corresponding bits to one in the WRITE POINTER VALID field.

Write Pointer[n]

The current value of the write pointer in the zone accessed by the operation with tag n.

Write Pointer Checksum

The write pointer checksum is the 2's complement of the sum of bytes 512..1 022 in the data structure. Each byte shall be added with unsigned arithmetic and overflow shall be ignored. The sum of bytes 512..1023 is zero if the checksum is correct.