NVM Express Technical Errata

Errata ID	**007**
Change Date	**8/22/2013**
Affected Spec Ver.	**NVM Express 1.0 and 1.1**
Corrected Spec Ver.	

Submission info

Name	Company	Date
Judy Brock	Samsung	5/29/2013
Dave Landsman	SanDisk	7/24/2013
Matthew Wilcox	Intel	7/24/2013
Mikyeong Kang	Samsung	7/24/2013
Edward Hsieh	Silicon Motion	8/21/2013

Section 7.4.1 on I/O Submission and Completion Queue creation has confusing wording. This erratum clarifies the meaning of “allocated” and “created” in this material.

The Active Firmware Info in the Firmware Slot Information log is clarified to show it is the slot that the active firmware was loaded from.

In section 5.11, a buffer of namespace IDs is returned. This erratum adds a Namespace List data structure definition to clarify the description of this structure.

The wording in the SGL Bit Bucket description and a few other SGL related items were clarified.

Typos were corrected in section 7.7 to correct Figure number references.

Description of the specification technical flaw:

Modify portions of section 7.4.1 as shown below:

7.4.1 Queue ~~Allocation~~ Setup and Initialization

To setup and initialize ~~allocate~~ I/O Submission Queues and I/O Completion Queues for use, host software follows these steps:

1. Configures ~~Allocates~~ the Admin Submission and Completion Queues by ~~configuring~~ initializing the Admin Queue Attributes (AQA), Admin Submission Queue Base Address (ASQ), and Admin Completion Queue Base Address (ACQ) registers appropriately.
2. Submits a Set Features command ~~for~~ with the Number of Queues attribute ~~in order~~ to request the desired number of I/O Submission Queues and I/O Completion Queues ~~desired~~. The completion queue entry for ~~of~~ this Set Features command indicates the number of I/O Submission Queues and I/O Completion Queues allocated by the controller.
3. Determines the maximum number of entries supported per queue (CAP.MQES) and whether the queues are required to be physically contiguous (CAP.CQR).
4. Creates ~~Allocates~~ the desired I/O Completion Queues within the limitations of the number allocated by the controller and the queue attributes supported (maximum entries and physically contiguous requirements) by using the Create I/O Completion Queue command.
5. Creates ~~Allocates~~ the desired I/O Submission Queues within the limitations of the number allocated by the controller and the queue attributes supported (maximum entries and physically contiguous requirements) by using the Create I/O Submission Queue command.

At the end of this process, the desired I/O Submission Queues and I/O Completion Queues have been setup and initialized ~~allocated~~ and may be used to complete I/O commands.

Modify the second and third paragraph of section 5.12.1.7 as shown below:

If a Set Features or Get Features command is submitted for this Feature, the attributes specified in Figure 99 are returned in Dword 0 of the completion queue entry for that command.

Figure 1: Number of Queues – Command Dword 11

Bit	Description
31:16	**Number of I/O Completion Queues Requested (NCQR):** Indicates the number of I/O Completion Queues requested by software. This number does not include the Admin Completion Queue. A minimum of one shall be requested, reflecting that the minimum support is for one I/O Completion Queue. This is a 0's based value.
15:00	**Number of I/O Submission Queues Requested (NSQR):** Indicates the number of I/O Submission Queues requested by software. This number does not include the Admin Submission Queue. A minimum of one shall be requested, reflecting that the minimum support is for one I/O Submission Queue. This is a 0's based value.

~~The number of queues allocated is returned in Dword 0 of the command completion queue entry. The definition is shown in Figure 99.~~

Modify byte 00 of Figure 76 as shown below:

Bytes	Description
00	**Active Firmware Info (AFI):** Specifies information about the active firmware revision. Bit 7 is reserved. Bits 6:4 indicates the firmware slot that is going to be activated at the next controller reset. If this field is 0h, then the controller does not indicate the firmware slot that is going to be activated at the next controller reset. Bit 3 is reserved. Bits 2:0 indicates the firmware slot from which ~~that contains~~ the actively running firmware revision was loaded.

Add section 4.7 as shown below (preceding the section on Fused Operations):

4.7 Namespace List

A Namespace List, defined in Figure TBD, is an ordered list of namespace IDs. Unused entries are zero filled.

Figure TBD: Namespace List Format

Bytes	Description
3:0	**Identifier 0:** This field contains the lowest namespace ID in the list or 0h if the list is empty.
7:4	**Identifier 1:** This field contains the second lowest namespace ID in the list or 0h if the list contains less than two entries.
…	…
(N*4+3): (N*4)	**Identifier N:** This field contains the N+1 lowest namespace ID in the list or 0h if the list contains fewer than N entries.

Modify the second paragraph of section 5.11 as shown below:

When the host specifies active namespaces to be returned, then the buffer is filled with a Namespace List as defined in Figure TBD that contains up to 1024 active namespaces ~~a list of 1024 namespace IDs is returned containing active namespace IDs in increasing order~~ that are greater than the value specified in the Namespace Identifier (CDW1.NSID) field of the command. ~~If there are fewer than 1024 active namespaces greater than the value in the CDW1.NSID field, then the unused portion of the list is zero filled. If there are 1024 or more active namespaces with namespace identifiers greater than the value in the CDW1.NSID field, then the last Namespace Identifier in the list is non-zero.~~

Modify Figure 81 as shown below:

Figure 81: Identify – Command Dword 10

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:02</td><td>Reserved</td></tr>
<tr><td>01:00</td><td>Controller or Namespace Structure (CNS): This field specifies the information to be returned to the host.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>The Identify Namespace data structure is returned to the host for the namespace specified in the Namespace Identifier (CDW1.NSID) field.</td></tr>
<tr><td>01b</td><td>The Identify Controller data structure is returned to the host.</td></tr>
<tr><td>10b</td><td>A Namespace List ~~list~~ of up to 1024 active namespace IDs is returned to the host containing active namespaces with a namespace identifier greater than the value specified in the Namespace Identifier (CDW1.NSID) field.</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table></td></tr>
</table>

Modify the Length field of Figure 21 as shown below:
(Note: *This version of Figure 21 is from ratified NVMe 1.1 ECN 006.)*

11:8	**Length:** If the SGL describes a destination data buffer (e.g., a read from the controller to host memory), then the Length field specifies the number of bytes of the source data to be discarded (i.e., not transferred to the destination data buffer) ~~not to be transferred (i.e., the number of bytes to be discarded)~~. A Length field set to 00000000h specifies that no source data shall be discarded. An SGL Bit Bucket descriptor specifying that no source data be discarded is a valid SGL Bit Bucket descriptor. If the SGL describes a source data buffer (~~i.e.,~~ e.g., a write from host memory to the controller) then the Length field shall be ignored ~~and no data shall be discarded from the source data or destination data. An SGL Bit Bucket descriptor specifying that no data be discarded shall not be processed as having an error~~. If SGL Bit Bucket descriptors are supported, their length in a destination data buffer shall be included in the Number of Logical Blocks (NLB) parameter specified in NVM Command Set data transfer commands. Their length in a source data buffer is not included in the NLB parameter.

Modify the fourth paragraph of section 4.4 as shown below:

A command shall be aborted if:

~~• an SGL segment contains greater than one SGL Segment descriptors, and/or SGL Last Segment descriptors; or~~

• an SGL segment contains an SGL Segment descriptor or an SGL Last Segment descriptor in other than the last descriptor in the segment; or

• a last SGL segment contains an SGL Segment descriptor, or an SGL Last Segment descriptor.

Modify the Metadata (SGL Segment Pointer) definition in Figure 124 (Compare), Figure 137 (Read), and Figure 159 (Write) as shown below:

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>63:00</td><td>If CDW0[15] is cleared to ‘0’, then the definition of this field is:

<table><tr><td>63:00</td><td>Metadata Pointer (MPTR): This field contains the address of a contiguous physical buffer of metadata, if applicable. This field shall be Dword aligned.</td></tr></table>
If CDW0[15] is set to ‘1’, then the definition of this field is:

<table><tr><td>63:00</td><td>Metadata SGL Segment Pointer (MSGLP): This field contains the address of an SGL segment containing exactly one SGL Descriptor that describes the metadata to transfer, if applicable.</td></tr></table></td></tr>
</table>

Modify the second paragraph of section 7.7 as shown below:

If bit 4 is set to ‘1’ in the Optional NVM Command Support field of the Identify Controller ~~D~~data structure in Figure ~~65~~ 82 then for each Feature, there are three settings: default, saveable, and current. If bit 4 is cleared to ‘0’ in the Optional NVM Command Support field of the Identify Controller ~~D~~data structure in Figure ~~65~~ 82 then the controller only supports a current and default value for each Feature. In this case, the current value may be persistent across power states based on the information specified in Figure 89 and Figure 90.

Disposition log

5/29/2013	Erratum captured.
6/19/2013	Updated based on 6/6 meeting.
7/24/2013	Added Firmware Slot clarification, Namespace List, and further SGL bit bucket clarifications.
8/1/2013	Changes based on 7/25 meeting.
8/21/2013	Corrected Figure number in section 7.7 and clarified section 5.12.1.7.
9/26/2013	Erratum ratified.