## NVM Express Technical Errata

| | |
|---|---|
| **Errata ID** | **006** |
| **Change Date** | **6/20/2013** |
| **Affected Spec Ver.** | **NVM Express 1.0 and 1.1** |
| **Corrected Spec Ver.** | |

Submission info

| Name | Company | Date |
|---|---|---|
| Barrett Mayes, Amber Huffman | Intel | 04/30/2013 |
| Ken Okin | Virident | 05/08/2013 |
| Santosh Singh | Samsung | 05/08/2013 |
| Olivier Mallinger | IP Maker | 05/08/2013 |
| Peter Onufryk | IDT | 05/14/2013 |
| Adam Geml | HGST | 05/16/2013 |
| Dave Landsman | SanDisk | 05/31/2013 |

Section 5.12.1.8 and 5.12.1.9 are unclear on interrupt coalescing and vector configuration behavior when the device is not configured for MSI-x or when changing interrupt modes. Clarifications are made on usage in different interrupt modes when switching interrupt modes.

The specification lists the PCI Express capability as optional. There is not a practical way you can build an NVMe device without implementing it. The conventional PCI spec is over a decade old and is no longer being revised. Additionally, OS bus drivers require PCIe to participate in IO technologies like VTd. Based on these factors, the capability is marked as mandatory.

A clarification is added that if IO commands are outstanding to a namespace when a Format NVM is issued for that namespace, then the Format NVM command may fail.

Clarifications were made regarding the maximum number of IO queues.

A clarification was made that the Maximum Data Transfer Size includes lengths associated with SGL Bit Buckets if used. The Number of Logical Blocks parameter was also clarified.

The use of the “Number of LBA Ranges” field in the LBA Range Type Feature was clarified.

A few updates for Write and Write Zeroes status codes were made and other typos were fixed.

A clarification was made that writes larger than the Atomic Write Unit (Normal or Power Fail) have no atomicity guarantees.

A clarification was made that any write to an invalid doorbell register results in a particular asynchronous event (whether Submission Queue or Completion Queue doorbells).

Description of the specification technical flaw:

***Add the following paragraph before Figure 100 in section 5.12.1.8:***

This Feature is valid when the device is configured for Pin Based, MSI, Multiple MSI or MSI-X interrupts. There is no requirement for the device to persist these settings if interrupt modes are changed. It is recommended that the host re-issue this Feature after changing interrupt modes.

**Figure100: Interrupt Coalescing – Command Dword 11**

| Bit | Description |
|---|---|
| 31:16 | Reserved |
| 15:08 | **Aggregation Time (TIME):** Specifies the recommended maximum time in 100 microsecond increments that a controller may delay an interrupt due to interrupt coalescing. A value of 0h corresponds to no delay (i.e., disabling this capability). The controller may apply this time per interrupt vector or across all interrupt vectors. The reset value of this setting is 0h. |
| 07:00 | **Aggregation Threshold (THR):** Specifies the desired minimum number of completion queue entries to aggregate per interrupt vector before signaling an interrupt to the host. This is a 0's based value. The reset value of this setting is 0h. |

***Add the following paragraph before Figure 101 in section 5.12.1.9:***

Prior to issuing this Feature, the host shall configure the specified Interrupt Vector with a valid I/O Completion Queue. Violation of this requirement or specifying an out of range Interrupt Vector results in undefined behavior.

**Figure 101: Interrupt Vector Configuration – Command Dword 11**

| Bit | Description |
|---|---|
| 31:17 | Reserved |
| 16 | **Coalescing Disable (CD):** If set to '1', then any interrupt coalescing settings shall not be applied for this interrupt vector. If cleared to '0', then interrupt coalescing settings apply for this interrupt vector. |
| 15:00 | **Interrupt Vector (IV):** This field specifies the interrupt vector for which the configuration settings are applied. |

***Modify the title of section 2.5 as shown below:***

**2.5 PCI Express Capability ~~(Optional)~~**

***Modify paragraph 3 in section 5.13 as shown below:***

The Format NVM command shall fail if the controller is in an invalid security state. See the TCG SIIS reference. The Format NVM command may fail if there are outstanding IO commands to the namespace specified to be formatted.

***Modify the beginning of section 1.4 as shown below.***

NVM Express is a scalable host controller interface designed to address the needs of Enterprise and Client systems that utilize PCI Express based solid state drives. The interface provides optimized command submission and completion paths. It includes support for parallel operation by supporting up to 65,535 ~~64K~~ I/O Queues with up to 64K outstanding commands per I/O Queue. Additionally, support has been added for

many Enterprise capabilities like end-to-end data protection (compatible with T10 DIF and SNIA DIX standards), enhanced error reporting, and virtualization.
The interface has the following key attributes:

- Does not require uncacheable / MMIO register reads in the command submission or completion path.
- A maximum of one MMIO register write is necessary in the command submission path.
- Support for up to 65,535 ~~64K~~ I/O queues, with each I/O queue supporting up to 64K outstanding commands.
- Priority associated with each I/O queue with well-defined arbitration mechanism.
- All information to complete a 4KB read request is included in the 64B command itself, ensuring efficient small I/O operation.
- Efficient and streamlined command set.
- Support for MSI/MSI-X and interrupt aggregation.
- Support for multiple namespaces.
- Efficient support for I/O virtualization architectures like SR-IOV.
- Robust error reporting and management capabilities.
- Support for multi-path I/O and namespace sharing.

***Modify Figure 27 in section 4.6 as shown below:***

**Figure 27: Completion Queue Entry: DW 3**

| Bit | Description |
|---|---|
| 31:17 | **Status Field (SF):** Indicates status for the command that is being completed. Refer to section 4.6.1. |
| 16 | **Phase Tag (P):** Identifies whether a Completion Queue entry is new. The Phase Tag values for all Completion Queue entries shall be initialized to '0' by host software prior to setting CC.EN to '1'. When the controller places an entry in the Completion Queue, it shall invert the phase tag to enable host software to discriminate a new entry. Specifically, for the first set of completion queue entries after CC.EN is set to '1' all Phase Tags are set to '1' when they are posted. For the second set of completion queue entries, when the controller has wrapped around to the top of the Completion Queue, all Phase Tags are cleared to '0' when they are posted. The value of the Phase Tag is inverted each pass through the Completion Queue. |
| 15:00 | **Command Identifier (CID):** Indicates the identifier of the command that is being completed. This identifier is assigned by host software when the command is submitted to the Submission Queue. The combination of the SQ Identifier and Command Identifier uniquely identifies the command that is being completed. The maximum number of requests outstanding at one time is 64K ~~for an I/O Submission Queue and 4K for the Admin Submission Queue~~. |

***Modify Figure 98 in section 5.12.1.7 as shown below:***

**Figure 98: 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. The maximum value that may be specified is 65,534 (indicating 65,535 I/O Completion Queues). |
| 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 maximum value that may be specified is 65,534 (indicating 65,535 I/O Submission Queues). |

***Modify byte 77 of Figure 82 as shown below:***

| | | |
|---|---|---|
| 77 | M | **Maximum Data Transfer Size (MDTS):** This field indicates the maximum data transfer size between the host and the controller. The host should not submit a command that exceeds this transfer size. If a command is submitted that exceeds the transfer size, then the command is aborted with a status of Invalid Field in Command. The value is in units of the minimum memory page size (CAP.MPSMIN) and is reported as a power of two (2^*n*). A value of 0h indicates no restrictions on transfer size. The restriction includes metadata if it is interleaved with the logical block data.<br><br>If SGL Bit Bucket descriptors are supported, their lengths shall be included in determining if a command exceeds the Maximum Data Transfer Size for destination data buffers. Their length in a source data buffer is not included for a Maximum Data Transfer Size calculation. |

***Modify Figure 93 as shown below:***

**Figure 93: LBA Range Type – Command Dword 11**

| Bit | Description |
|---|---|
| 31:06 | Reserved |
| 05:00 | **Number of LBA Ranges (NUM):** This field specifies the number of LBA ranges specified in this command. This is a 0's based value. This field is used for the Set Features command only and is ignored for the Get Features command for this Feature. |

***Modify Figure 33 as shown below:***

**Figure 33: Status Code – Command Specific Status Values, NVM Command Set**

| Value | Description | Commands Affected |
|---|---|---|
| 80h | Conflicting Attributes | Dataset Management, Read, Write |
| 81h | Invalid Protection Information | Compare, Read, Write, Write Zeroes |
| 82h | Attempted Write to Read Only Range | Write, Write Zeroes |
| 83h - BFh | Reserved | |

***Modify Figure 166 as shown below:***

**Figure 166: Write – Command Specific Status ~~Errors~~ Values**

| Value | Description |
|---|---|
| 80h | **Conflicting Attributes:** The attributes specified in the command are conflicting. |
| 81h | **Invalid Protection Information:** The Protection Information settings specified in the command are invalid. |
| 82h | **Attempted Write to Read Only Range:** The LBA range specified contains read-only blocks. |

***Modify Figure 173 as shown below:***

**Figure 173: Write Zeroes – Command Specific Status Values**

| Value | Description |
|---|---|
| 81h | **Invalid Protection Information:** The Protection Information settings specified in the command are invalid. |
| 82h | **Attempted Write to Read Only Range:** The LBA range specified contains read-only blocks. |

***Modify bytes 529:526 of Figure 82 as shown below:***

| | | |
|---|---|---|
| 527:526 | M | **Atomic Write Unit Normal (AWUN):** This field indicates the atomic write size for the controller during normal operation. This field is specified in logical blocks and is a 0's based value. If a write is submitted of this size or less, the host is guaranteed that the write is atomic to the NVM with respect to other read or write operations. If a write is submitted that is greater than this size, there is no guarantee of atomicity.<br><INSERT BLANK LINE><br>A value of FFFFh indicates all commands are atomic as this is the largest command size. It is recommended that implementations support a minimum of 128KB (appropriately scaled based on LBA size). |
| 529:528 | M | **Atomic Write Unit Power Fail (AWUPF):** This field indicates the atomic write size for the controller during a power fail condition. This field is specified in logical blocks and is a 0's based value. If a write is submitted of this size or less, the host is guaranteed that the write is atomic to the NVM with respect to other read or write operations. If a write is submitted that is greater than this size, there is no guarantee of atomicity. |

***Modify Figure 44 as shown below:***

**Figure 44: Asynchronous Event Information – Error Status**

| Value | Description |
|---|---|
| 0h | **Write to Invalid Doorbell Register ~~Invalid Submission Queue~~:** Host software wrote the doorbell of a queue that was not created. |
| 1h | **Invalid Doorbell Write Value:** Host software attempted to write an invalid doorbell value. Some possible causes of this error are:<br>• the value written was out of range of the corresponding queue's base address and size,<br>• the value written is the same as the previously written doorbell value,<br>• host software attempts to add a command to a full Submission Queue, and<br>• host software attempts to remove a completion queue entry from an empty Completion Queue. |
| 2h | **Diagnostic Failure:** A diagnostic failure was detected. This may include a self test operation. |
| 3h | **Persistent Internal Device Error:** A failure occurred within the device that is persistent or the device is unable to isolate to a specific set of commands. If this error is indicated, then the CSTS.CFS bit may be set to '1' and the host should perform a reset as described in section 7.3. |
| 4h | **Transient Internal Device Error:** A transient error occurred within the device that is specific to a particular set of commands and operation may continue. |
| 5h | **Firmware Image Load Error:** The firmware image could not be loaded. The controller reverted to the previously active firmware image or a baseline read-only firmware image. |
| 6h - FFh | Reserved |

***Modify Figure 21 as shown below:***

**Figure 21: SGL Bit Bucket descriptor**

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>7:0</td><td>Reserved</td></tr>
<tr><td>11:8</td><td>Length: If the SGL describes a destination data buffer, then the Length field specifies the number of bytes of the source data 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.<br><br>If the SGL describes a source data buffer (i.e., 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.<br><br>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.</td></tr>
<tr><td>14:12</td><td>Reserved</td></tr>
<tr><td>15</td><td>SGL Identifier: The definition of this field is described in the table below.<br><br>
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>03:00</td><td>Zero: The Zero field shall have the value of 0h. An SGL Bit Bucket descriptor containing a Zero field set to a value other than 0h shall be processed as having an error.</td></tr>
<tr><td>07:04</td><td>SGL Descriptor Type: 1h as specified in Figure 19.</td></tr>
</table>
</td></tr>
</table>

***Modify the first paragraph of section 4.4.1 as shown below:***

Figure 24 shows an example of a data read request using SGLs. In the example, the logical block size is 512B. The total length of the logical blocks accessed on the device is 13KB, of which only 11KB is transferred to the host. The Number of Logical Blocks (NLB) field in the command shall specify 26, indicating the total length of the logical blocks accessed on the device is 13KB. There are three SGL segments describing the locations in host memory where the logical block data is transferred. ~~In the example, the logical block size is 512B.~~

***Modify bytes 79:78 in Figure 82 as shown below:***

| | | |
|---|---|---|
| 79:78 | M | **Controller ID (CNTLID):** Contains the NVM subsystem unique controller identifier associated with the controller. Refer to section ~~7.7~~ 7.8 for unique identifier requirements. |

***Modify the second paragraph of section 3.1.11 as shown below:***

The host should not read the doorbell registers. If a doorbell register is read, the value returned is vendor specific. Writing to a non-existent ~~or unallocated~~ Submission Queue Tail Doorbell has undefined results.

***Modify the second paragraph of section 3.1.12 as shown below:***

The host should not read the doorbell registers. If a doorbell register is read, the value returned is vendor specific. Writing to a non-existent ~~or unallocated~~ Completion Queue Head Doorbell has undefined results.

***Update Figure 68 as shown below:***

**Figure 68: Get Features – Feature Identifiers**

| Description | Section Defining Format of Attributes Returned |
|---|---|
| Arbitration | Section 5.12.1.1 |
| Power Management | Section 5.12.1.2 |
| LBA Range Type | Section 5.12.1.3 |
| Temperature Threshold | Section 5.12.1.4 |
| Error Recovery | Section 5.12.1.5 |
| Volatile Write Cache | Section 5.12.1.6 |
| Number of Queues | Section 5.12.1.7 |
| Interrupt Coalescing | Section 5.12.1.8 |
| Interrupt Vector Configuration | Section 5.12.1.9 |
| Write Atomicity | Section 5.12.1.10 |
| Asynchronous Event Configuration | Section 5.12.1.11 |
| Autonomous Power State Transition | Section 5.12.1.12 |
| **NVM Command Set Specific** | |
| Software Progress Marker | ~~Section 5.12.1.12~~<br>Section 5.12.1.13 |
| Host Identifier | ~~Section 5.12.1.13~~<br>Section 5.12.1.14 |
| Registration Notification Mask | ~~Section 5.12.1.14~~<br>Section 5.12.1.15 |
| Reservation Presistence | ~~Section 5.12.1.15~~<br>Section 5.12.1.16 |

***Update the first paragraph of section 7.6.1.1 as shown below:***

The Software Progress Marker feature, defined in section ~~5.12.1.12~~ 5.12.1.13, indicates the number of times pre-boot software has loaded prior to the OS successfully loading. If the pre-boot software load count becomes large, it may indicate there are issues with cached data within the NVM since the OS driver software has not set this field to 0h recently. In this case, the OS driver software may choose to use the NVM more conservatively (e.g., not utilize cached data).

***Update section 8.8.1 as shown below:***

There are three types of reservation notifications: registration preempted, reservation released, and reservation preempted. Conditions that cause a reservation notification to occur are described in the following sections. A Reservation Notification log page is created whenever an unmasked reservation notification occurs on a namespace associated with the controller (see section 5.10.1.4.1). Reservation notifications may be masked from generating a reservation log page on a per reservation notification type and per namespace ID basis through the Reservation Notification Mask feature (see section ~~5.12.1.14~~ 5.12.1.15). A host may use the Asynchronous Event Request command to be notified of the presence of one or more available Reservation Notification log pages (see section 5.2).

***Update the second paragraph section 8.8.2 as shown below:***

Registering a reservation key with a namespace creates an association between a host and a namespace. A host that is a registrant of a namespace may use any controller with which it is associated (i.e., that has the same Host Identifier, refer to section ~~5.12.1.13~~ 5.12.1.14) to access that namespace as a registrant. Thus, a host need only register on a single controller in order to become a registrant of the namespace on all controllers in the NVM Subsystem that have access to the namespace and are associated with the host.

Disposition log

| | |
|---|---|
| 5/8/2013 | Erratum created. |
| 5/14/2013 | Included Atomic Write Unit Normal and Power Fail atomicity clarifications. Added invalid queue as an item for discussion. |
| 5/16/2013 | Modified Async Event for invalid doorbell register write to cover Submission and Completion Queue doorbells. |
| 5/31/2013 | Added clarification of NLB parameter when Bit Buckets are used, corrected a few typos, and fixed cross references. |
| 6/20/2013 | Clarifications for SGL Bit Bucket and number of commands outstanding. |
| 7/29/2013 | Erratum ratified. |

*Technical input submitted to the NVMHCI Workgroup is subject to the terms of the NVMHCI Contributor's agreement.*