NVM Express Technical Errata

Errata ID	**008**
Change Date	**11/6/2013**
Affected Spec Ver.	**NVM Express 1.0 and NVM Express 1.1a**
Corrected Spec Ver.	

Submission info

Name	Company	Date
Jason Gao	Western Digital	8/22/2013
John Carroll	Intel	9/4/2013
Peter Onufryk	PMC	9/6/2013
Michael Xing	Microsoft	9/11/2013

The definition of reserved has been refined to clarify cases of value coded fields.

The minor version number of the 1.1 specification has been clarified.

The Command Processing section has been clarified related to arbitration.

Recommendations have been added for asynchronous events that may persist and for setting temperature threshold values.

PRP entry and PRP List requirements are clarified.

Several editorial fixes are also included.

Description of the specification technical flaw:

Modify portions of section 1.7.5 as shown below:

1.7.5 Reserved

A keyword referring to ~~indicating reserved~~ bits, bytes, words, fields, and opcode values that are set-aside for future standardization. Their use and interpretation may be specified by future extensions to this or other specifications. A reserved bit, byte, word, field, or register shall be cleared to zero, or in accordance with a future extension to this specification. The recipient is not required to ~~shall not~~ check reserved bits, bytes, words, or fields. Receipt of reserved coded values in defined fields in commands shall be reported as an error. Writing a reserved coded value into a controller register field produces undefined results.

Modify Figure 10 as shown below:

Figure 10: Command Dword 0

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td>Command Identifier (CID): This field specifies a unique identifier for the command when combined with the Submission Queue identifier.</td></tr>
<tr><td>15</td><td>PRP or SGL for Data Transfer (PSDT): This field specifies whether PRPs or SGLs are used for any data transfer associated with the command. If cleared to ‘0’, the command uses PRPs for any associated data or metadata transfer. If set to ‘1’, the command uses SGLs for any associated data or metadata transfer. PRPs shall be used for all Admin commands.</td></tr>
<tr><td>14:10</td><td>Reserved</td></tr>
<tr><td>09:08</td><td>Fused Operation (FUSE): In a fused operation, a complex command is created by “fusing” together two simpler commands. Refer to section 6.1. This field specifies whether this command is part of a fused operation and if so, which command it is in the sequence.
<table>
<tr><th>Value ~~Bits~~</th><th>Definition</th></tr>
<tr><td>00b</td><td>Normal operation</td></tr>
<tr><td>01b</td><td>Fused operation, first command</td></tr>
<tr><td>10b</td><td>Fused operation, second command</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>07:00</td><td>Opcode (OPC): This field specifies the opcode of the command to be executed.</td></tr>
</table>

Modify a portion of section 3.1.2.1 as shown below:

3.1.2 Offset 08h: VS – Version

This register indicates the major and minor version of the NVM Express specification that the controller implementation supports. ~~The upper two bytes represent the major version number, and the lower two bytes represent the minor version number. Example: Version 3.12 would be represented as 00030102h.~~ Valid versions of the specification are: 1.0 and 1.1.

3.1.2.1 VS Value for 1.0 Compliant Controllers

Bit	Type	Reset	Description
31:16	RO	0001h	**Major Version Number (MJR):** Indicates the major version is “1”
15:0~~800~~	RO	~~00~~00h	**Minor Version Number (MNR):** Indicates the minor version is “0”.
07:00	RO	00h	Reserved

3.1.2.1 VS Value for 1.1 Compliant Controllers

Bit	Type	Reset	Description
31:16	RO	0001h	**Major Version Number (MJR):** Indicates the major version is “1”

15:08	RO	01h	**Minor Version Number (MNR):** Indicates the minor version is "1".
07:00	RO	00h	Reserved

Modify steps 3 and 4 in section 7.2.1 on Command Processing as shown below:

3. The controller fetches the command(s) in the Submission Queue from host memory for future execution. Arbitration is the method used to determine the Submission Queue from which the controller starts processing the next command, refer to section 4.8.

4. The controller then proceeds with execution of the next command. Commands may complete out of order (the order submitted or started execution). ~~The controller performs command arbitration. The controller may execute commands out of order. The command arbitration mechanism selects the next command to execute from commands that have been previously fetched. The controller then proceeds with execution of the command.~~

Modify the last two paragraphs of section 5.2 (Asynchronous Event Request command) as shown below:

The following event types are defined:

- Error event: Indicates a general error that is not associated with a specific command. To clear this event, host software reads the Error Information log using the Get Log Page command.
- SMART / Health Status event: Indicates a SMART or health status event. To clear this event, host software reads the SMART / Health Information log using Get Log Page. The SMART / Health conditions that trigger asynchronous events may be configured in the Asynchronous Event Configuration feature using the Set Features command (see section 5.12).
- I/O Command Set events: Events that are defined by an I/O command set.
    - NVM Command Set Events:
        - Reservation Log Page Available event: Indicates that one or more Reservation Notification log pages are available.
- Vendor Specific event: Indicates a vendor specific event. To clear this event, host software reads the indicated vendor specific log page using Get Log Page.

Asynchronous events may be reported due to a single instance (e.g., Invalid Doorbell Write Value) or a persistent condition (e.g., Temperature Above Threshold). If the asynchronous event is triggered due to a persistent condition, the host should modify the event threshold or mask the event before issuing another Asynchronous Event Request command. If the host clears the event without taking these recommended actions for a persistent condition, then the persistent condition may cause repeated reporting of asynchronous events.

When the controller needs to report an event and there are no outstanding Asynchronous Event Request commands, the controller queues the event internal to the controller and reports it when an Asynchronous Event Request command is received.

Modify the first two paragraphs of section 5.12.1.4 as shown below:

5.12.1.4 Temperature Threshold (Feature Identifier 04h)

This Feature indicates the threshold for the temperature of the overall device (controller and NVM included) in units of Kelvin. If this temperature is exceeded, then an asynchronous event may be issued to the host. ~~The host should configure this feature prior to enabling asynchronous event notification for the temperature exceeding the threshold.~~ The attributes are indicated in Command Dword 11.

If the default threshold value is set to 0h or FFFFh, then the host should not enable reporting of asynchronous notifications for temperature (refer to section 5.12.1.11).

If a Get Features command is submitted for this Feature, the attributes specified in Figure 95 are returned in Dword 0 of the completion queue entry for that command.

Modify the first two paragraphs of section 5.12.1.11 as shown below:

5.12.1.11 Asynchronous Event Configuration (Feature Identifier 0Bh)

This Feature controls the events that trigger an asynchronous event notification to the host. This Feature may be used to disable reporting events in the case of a persistent condition (refer to section 5.2). The attributes are indicated in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 103 are returned in Dword 0 of the completion queue entry for that command.

Figure 103: Asynchronous Event Configuration – Command Dword 11

Bit	Description
31:08	Reserved
07:00	**SMART / Health Critical Warnings:** This field determines whether an asynchronous event notification is sent to the host for the corresponding Critical Warning specified in the SMART / Health Information Log (refer to Figure 75). If a bit is set to '1', then an asynchronous event notification is sent when the corresponding critical warning bit is set to '1' in the SMART / Health Information Log. If a bit is cleared to '0', then an asynchronous event notification is not sent when the corresponding critical warning bit is set to '1' in the SMART / Health Information Log.

Modify bytes 23:16 of Figure 12 as shown below:

<table>
<tr><td>23:16</td><td>If CDW0[15] is cleared to '0', then the definition of this field is:

<table><tr><td>23:16</td><td>Metadata Pointer (MPTR): This field contains the address of a contiguous physical buffer of metadata. This field is only used if metadata is not interleaved with the logical block data, as specified in the Format NVM command. This value <s>field</s> shall be Dword aligned.</td></tr></table>
If CDW0[15] is set to '1', then the definition of this field is:

<table><tr><td>23:16</td><td>Metadata SGL Segment Pointer (MSGLP): This field contains the address of an SGL segment which describes the metadata to transfer. This field is only used if metadata is not interleaved with the logical block data, as specified in the Format NVM command. This value <s>field</s> shall be Qword aligned. Refer to section 4.4.</td></tr></table></td></tr>
</table>

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 value ~~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 Namespace Identifier field in bytes 07:04 of Figure 12 as shown below:

07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace is not used for the command, then this field shall be cleared to 0h. If a command shall be applied to all namespaces on the device, then this value shall be set to FFFFFFFFh. Unless otherwise noted, specifying an inactive namespace ID in a command that uses the namespace ID shall cause the controller to abort the command with status Invalid Field in Command. Specifying an invalid namespace ID in a command that uses the namespace ID shall cause the controller to abort the command with status Invalid Namespace or Format.

Update the last two paragraphs of section 4.3 as shown below:

Dependent on the command definition, the ~~The~~ first PRP entry contained within the command may have a non-zero offset within the memory page. The first PRP List entry (i.e. the first pointer to a memory page containing additional PRP entries) that if present is typically contained in the PRP Entry 2 location within the command, shall be Qword aligned and may also have a non-zero offset within the memory page.
<ADD BLANK LINE>
PRP entries contained within a PRP List shall have a memory page offset of 0h. If a second PRP entry is present within a command, it shall have a memory page offset of 0h. In both cases, the entries are memory page aligned based on the value in CC.MPS.
~~All other PRP and PRP List entries shall have a memory page offset of 0h, i.e. the entries are memory page aligned based on the value in CC.MPS. The last entry within a memory page, as indicated by the memory page size in the CC.MPS field, shall be a PRP List pointer if there is more than a single memory page of data to be transferred~~.

PRP Lists shall be minimally sized with packed entries starting with entry 0. If more PRP List pages are required, then the last entry of the PRP List page is a pointer to the next PRP List page. The next PRP List page shall be memory page aligned. The total number of PRP entries is implied by the command parameters and memory page size.

In the paragraph after Figure 183 in section 8.4, modify the text as follows:

The host may dynamically modify the power state using the Set Features command and determine the current power state using the Get Features command. The host may directly transition between any two supported power states. The Entry Latency (ENTLAT) field in the power management descriptor indicates the maximum amount of time in microseconds that it takes to enter that power state and the Exit Latency (~~EXTLAT~~ EXLAT) field indicates that maximum amount of time in microseconds that it takes to exit that state. The maximum amount of time to transition between any two power states is equal to the sum of the old state’s exit latency and the new state’s entry latency. The host is not required to wait for a previously submitted power state transition to complete before initiating a new transition. The maximum amount of time

for a sequence of power state transitions to complete is equal to the sum of transition times for each individual power state transition in the sequence.

Modify section 7.3.2 as shown below:

There are five primary controller level reset mechanisms:

- NVM Subsystem Reset
- Conventional Reset (PCI Express Hot, Warm, or Cold reset)
- PCI Express transaction layer Data Link Down status
- Function Level Reset (PCI reset)
- Controller Reset (CC.EN transitions from '1' to '0')

When any of the above resets occur, the following actions are performed:

- The controller stops processing any outstanding Admin or I/O commands.
- All I/O Submission Queues are deleted.
- All I/O Completion Queues are deleted.
- ~~All outstanding I/O commands shall be processed as aborted by host software.~~
- ~~All outstanding Admin commands shall be processed as aborted by host software.~~
- The controller is brought to an Idle state. When this is complete, CSTS.RDY is cleared to '0'.
- The Admin Queue registers (AQA, ASQ, or ACQ) are not reset as part of a controller reset. All other controller registers defined in section 3 and internal controller state are reset.

In all cases except a Controller Reset, the PCI register space is reset as defined by the PCI Express base specification. Refer to the PCI Express specification for further details.

To continue after a reset, the host shall:

- Update register state as appropriate.
- Set CC.EN to '1'.
- Wait for CSTS.RDY to be set to '1'.
- Configure the controller using Admin commands as needed.
- Create I/O Completion Queues and I/O Submission Queues as needed.
- Proceed with normal I/O operations.

Note that all cases except a Controller Reset result in the controller immediately losing communication with the host. In these cases, the controller is unable to indicate any aborts or update any completion queue entries.

Modify section 2.2.3 as shown below:

Bit	Type	Reset	Description
15	~~RO~~ RWC	0	**PME Status (PMES):** ~~Not supported by NVM Express.~~ Refer to the PCI SIG specifications.
14:13	~~RO~~ RW	0	**Data Scale (DSC):** ~~Not supported by NVM Express.~~ Refer to the PCI SIG specifications.
12:09	RO	0	**Data Select (DSE):** ~~Not supported by NVM Express.~~ Refer to the PCI SIG specifications.
08	~~RO~~ RWS	0	**PME Enable (PMEE):** ~~Not supported by NVM Express.~~ Refer to the PCI SIG specifications.
07:04	RO	0	Reserved
03	RO	1	**No Soft Reset (NSFRST):** A value of '1' indicates that the controller transitioning from D3hot to D0 because of a power state command does not perform an internal reset.
02	RO	0	Reserved
01:00	R/W	00	**Power State (PS):** This field is used both to determine the current power state of the controller and to set a new power state. The values are: 00 – D0 state 01 – D1 state 10 – D2 state 11 – $D3_{HOT}$ state When in the $D3_{HOT}$ state, the controller's configuration space is available, but the register I/O and memory spaces are not. Additionally, interrupts are blocked.

Disposition log

8/22/2013	Erratum captured.
9/4/2013	Added version number change. Modified formatting to match other ECNs.
9/6/2013	Clarified arbitration within the Command Processing section.
9/11/2013	Added clarifications for asynchronous events and temperature.
9/24/2013	Updated reserved definition and Version register.
10/1/2013	Added changes for the Version register.
10/17/2013	Updates based on discussion in 10/10 meeting.
10/24/2013	Updates based on discussion in 10/17 meeting.
10/31/2013	Updates based on discussion in 10/24 meeting.
11/6/2013	Updated PME Status and PME Enable to RWC and RWS, respectively.
12/17/2013	Erratum ratified.