LEGAL NOTICE:

This NVM Express revision 1.3 technical proposal is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this NVM Express revision 1.3 technical proposal subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **"© 2007 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

NVM Express Workgroup
c/o VTM Group
3855 SW 153rd Drive
Beaverton, OR 97003 USA
info@nvmexpress.org

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**4032**
Change Date	**2018-07-23**
Builds on Specification	**NVM Express 1.3** **TP 4000a**

Technical Proposal Author(s)

Name	Company
John Maroney, Paul Suhler	Micron Technology
Peter Onufryk	Microsemi

This technical proposal defines a mechanism for a controller to indicate to a host to (i) prevent congestion in a PCI Express fabric due to PMR PCIe write requests and (ii) determine the time it will take for a PCI Express read request associated with a PMR write barrier mechanism to complete.

Revision History

Revision Date	Change Description
2018-01-31	Initial version
2018-02-22	Added explanatory text to 4.8
2018-02-23	Incorporate changes from 22 February Technical WG meeting.
2018-03-26	Reworded read bypass discussion to refer to memory reads and writes, rather than to commands. Removed 2^n clarifications of KB, MB, etc. as those abbreviations are defined as powers of two. Clarified definition of PMRSWTV field.
2018-04-10	Incorporated changes from 5 April Technical WG meeting: • Clarify “conflicting” reads and writes • Define MB and GB, as those are not in the conventions section; only KB is in conventions.
2018-04-23	Incorporated changes from 19 April Technical WG meeting: • Delete reference to zero-length reads in text describing conflicting reads and writes. • Add definitions of MB and GB to Conventions, and use MB and GB in the proposal.
2018-04-26	Incorporated changes from 26 April Technical WG meeting: • Fixed typos This version is intended for thirty-day member review.

2018-05-31	Changes from thirty-day review: • Corrected name of related field mentioned in definition of PMRSWTV field.
2018-07-17	Corrected oversight discovered after integration: • Added two new registers defined in this TP to the register definition figure in section 3.1. • Corrected abbreviation PMRWSTP to PMRSWTP to match name.
2018-07-23	Ratified

Description for NVMe 1.4 Changes Document

PMR Elasticity Buffer Status

- Two read-only registers are added, for the controller to indicate the PMR elasticity buffer size and PMR sustained throughput.
- Explanatory text is added to the PMR section describing the usage of the size and throughput.
- Added definitions for MB and GB to Conventions section.
- References:
  - NVMe 1.3 sections 1.5, 3.1, and 4.8
  - Technical Proposal 4032
  - Technical Proposal 4032a

Description of Specification Changes

Insert two paragraphs in Section 1.5 as shown below:

When a size is stated in the document as KB, the convention used is 1KB = 1024 bytes.

When a size is stated in the document as MB, the convention used is 1MB = 1024 * 1024 bytes.

When a size is stated in the document as GB, the convention used is 1GB = 1024 * 1024 * 1024 bytes.

The ^ operator is used to denote the power to which that number, symbol, or expression is to be raised.

Insert the following row in the register definition table, in section 3.1:

Start	End	Symbol	Description
00h	07h	CAP	Controller Capabilities
…			
50h	DFFh	Reserved	Reserved
E00h	E03h	PMRCAP	Persistent Memory Capabilities (Optional)
E04h	E07h	PMRCTL	Persistent Memory Region Control (Optional)
E08h	E0Bh	PMRSTS	Persistent Memory Region Status (Optional)
E0Ch	E0Fh	PMREBS	Persistent Memory Region Elasticity Buffer Size
E10h	E13h	PMRSWTP	Persistent Memory Region Sustained Write Throughput
~~E0Ch~~E14h	EFFh	Reserved	Reserved
…			

< Editor's Note: Register addresses before the changes in this table correspond to those in NVMe 1.NEXTc.>

Insert the following sections after Section 3.1.18 (TP 4000a) as shown below:

3.1.19 Offset E0Ch: PMREBS – Persistent Memory Region Elasticity Buffer Size.

This optional register identifies to the host the size of the PMR elasticity buffer. A value of zero in this register indicates to the host that no information regarding the presence or size of a PMR elasticity buffer is available.

<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>31:8</td><td>RO</td><td>Impl Spec</td><td>PMR Elasticity Buffer Size Base (PMRWBZ): Indicates the size of the PMR elasticity buffer. The actual size of the PMR elasticity buffer is equal to the value in this field multiplied by the value specified by the PMR Elasticity Buffer Size Units field.</td></tr>
<tr><td>7:5</td><td>RO</td><td>0h</td><td>Reserved</td></tr>
<tr><td>4</td><td>RO</td><td>Impl Spec</td><td>Read Bypass Behavior: If a memory read does not conflict with any memory write in the PMR Elasticity Buffer (i.e., if the set of memory addresses specified by a read is disjoint from the set of memory addresses specified by all writes in the PMR Elasticity Buffer), and this bit is:
a) set to ‘1’, then memory reads not conflicting with memory writes in the PMR Elasticity Buffer shall bypass those memory writes; and
b) cleared to ‘0’, then memory reads not conflicting with memory writes in the PMR Elasticity Buffer may bypass those memory writes.</td></tr>
<tr><td>3:0</td><td>RO</td><td>Impl Spec</td><td>PMR Elasticity Buffer Size Units (PMRSZU): Indicates the granularity of the PMR Elasticity Buffer Size field.
<table>
<tr><th>Value</th><th>Granularity</th></tr>
<tr><td>0h</td><td>Bytes</td></tr>
<tr><td>1h</td><td>One KB</td></tr>
<tr><td>2h</td><td>One MB</td></tr>
<tr><td>3h</td><td>One GB</td></tr>
<tr><td>7h – Fh</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

3.1.20 Offset E10h: PMRSWTP – Persistent Memory Region Sustained Write Throughput

This optional register identifies to the host the maximum PMR sustained write throughput. A value of zero in this register indicates to the host that no information regarding the PMR sustained write throughput is available.

< Editor's note: Correct the abbreviation in the heading and in the figure title. >

<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>31:8</td><td>RO</td><td>Impl Spec</td><td>PMR Sustained Write Throughput (PMRSWTV): Indicates the sustained write throughput of the PMR at the maximum PCIe TLP payload size, as specified in the Max_Payload_Size (MPS) field of the PCIe Express Device Control (PXDC) register. The actual sustained write throughput of the PMR is equal to the value in this field multiplied by the units specified by the PMR Sustained Write Throughput Units field.</td></tr>
<tr><td>7:4</td><td>RO</td><td>0h</td><td>Reserved</td></tr>
<tr><td>3:0</td><td>RO</td><td>Impl Spec</td><td>PMR Sustained Write Throughput Units (PMRSWTU): Indicates the granularity of the PMR Sustained Write Throughput field.
<table>
<tr><th>Value</th><th>Granularity</th></tr>
<tr><td>0h</td><td>Bytes/second</td></tr>
<tr><td>1h</td><td>One KB/second</td></tr>
<tr><td>2h</td><td>One MB/second</td></tr>
<tr><td>3h</td><td>One GB/second</td></tr>
<tr><td>7h – Fh</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

Make the following addition to section 4.8:

4.8 Persistent Memory Region

…

The Persistent Memory Region (PMR) is an optional region of general purpose PCI Express read/write persistent memory that may be used for a variety of purposes. The controller indicates support for the PMR by setting CAP.PMRS to '1' and indicates whether the controller supports command data and metadata transfers to or from the PMR by setting support flags in the PMRCAP register. When command data and metadata transfers to or from PMR are supported, all data and metadata associated with a particular command shall be either entirely located in the Persistent Memory Region or outside the Persistent Memory Region.

The PCI Express address range and size of the PMR is defined by the PCI Base Address register (BAR) indicated by PMRCAP.BIR. The PMR consumes the entire address region exposed by the BAR and supports all the required features of the PCI express programming model (i.e., it in no way restricts what is otherwise permitted by PCI Express).

The contents of data written to the PMR while the PMR is ready persists across power cycles, NVM subsystem resets, controller resets, and disabling of the PMR. The mechanism used to make a write to the PMR persistent is implementation specific. For example, in one implementation this may mean that a write to non-volatile memory has completed while in another implementation this may mean that the write has been stored in a non-volatile write buffer and is written to non-volatile memory at some later point.

A PMR implementation has a maximum sustained write throughput. The PMR implementation may also have an optional write elasticity buffer used to buffer writes from PMR PCIe write requests. When the PMR sustained write throughput is less than the PCI Express link throughput, then such a write elasticity buffer allows PCIe write request burst throughput to exceed the PMR sustained write throughput without back pressuring into the PCI Express fabric.

The time required to transfer data from the write elasticity buffer to nonvolatile media is the amount of data written to the elasticity buffer divided by the Persistent Memory Region Sustained Write Throughput (refer to 3.1.20). The time to transfer the entire contents of the write elasticity buffer is the Persistent Memory Region Elasticity Buffer Size (refer to 3.1.19) divided by the Persistent Memory Region Sustained Write Throughput.

The host enables the PMR by setting PMRCTL.EN to '1'. Once enabled, the controller indicates that the PMR is ready by setting PMRSTS.NRDY to '0'. It is not necessary to enable the controller to enable the PMR. Restoring and saving the contents of the PMR may take time to complete. When the host modifies the value of PMRCTL.EN, the host should wait for at least the time interval specified in PMRCAP.PMRTO for PMRSTS.NRDY to reflect the change.

When the PMR is not ready, PMR reads complete successfully and return an undefined value while PMR writes complete normally, but do not update memory (i.e., the contents of the PMR address written remains unchanged). The undefined value returned by a PMR read following a sanitize operation is such that recovery of any previous user data from any cache or the non-volatile media is not possible.

When the PMR becomes read-only or unreliable, then a critical warning is reported in the SMART/Health Information Log which may be used to trigger an NVMe asynchronous event. Since reporting of an asynchronous event may occur an unspecified amount of time after the PMR health status has changed, the host should assume that all operations to the PMR have been affected since the last time normal operation was reported in PMRSTS.HSTS.

PMRWBM enumerates supported PMR write barrier mechanisms. At least one mechanism shall be supported. An implementation may optionally support a mechanism where a PCI Express read of any size to the PMR, including a "zero-length read," ensures that all previous memory writes (i.e., Posted PCI Express requests) to the PMR have completed and are persistent. An implementation may optionally support a write barrier mechanism that utilizes a read of the PMRSTS register. When supported, a read of the PMRSTS register allows a host to:

- ensure that previously issued memory writes to the PMR have completed; and
- determine whether the PMR updates associated with those writes have completed without error and are persistent.

A PMR memory write error may be the result of a poisoned PCI Express TLP, an NVM subsystem internal error, or a PMR health status issue.

Regardless of the supported PMR write barrier mechanisms, a host may periodically read PMRSTS to ensure that reads to the PMR have returned valid data. For example, if a read to the PMRSTS register indicates that the PMR is operating normally is then followed by a series of reads, and finally a second read to the PMRSTS register that indicates the PMR is unreliable, then one or more of the reads between the two PMRSTS reads may have returned invalid data. Such polling of the PMRSTS register may be unnecessary if the host handles poisoned TLPs and/or poisoned TLP error reporting is enabled.

The PMR write elasticity buffer size along with the PMR sustained write throughput allows a host to determine the amount of time for a read associated with a persistent memory region write barrier mechanism to complete.

< end>