LEGAL NOTICE:

This NVM Express revision 1.4 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 1.4 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 - 2019 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, Inc.
3855 SW 153rd Drive
Beaverton, OR 97003 USA
info@nvmexpress.org

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**4059a – CMB Write Elasticity Status**
Change Date	**2020-07-20**
Builds on Specification	**NVM Express 1.4**
Ratified Technical Proposals Referenced	

Technical Proposal Author(s)

Name	Company
Paul Suhler, John Maroney	Micron Technology
Peter Onufryk	Microchip
Mike Allison	Intel

This technical proposal defines a mechanism for a controller to indicate to a host information used to prevent congestion in a PCI Express fabric due to CMB PCIe write requests.

Revision History

Revision Date	Change Description
2019-03-06	Initial version
2019-04-02	Notes from 2019-03-07 meeting of Technical WG: • Corrected KB, etc. to KiB, etc. • Deleted two instances of “actual”. • Removed barrier mechanism, as it is redundant. • Fixed section references. • Updated base spec from 1.3c to 1.3d
2019-04-04	Changes from 2019-04-04 Technical WG meeting: • Added captions for figures. • Reworked register and field abbreviations for consistency. • Added note to reconsider in phase 3 whether the time can be expressed without referring to an elasticity buffer.
2019-05-06	Made changes suggested by Mike Allison in e-mail. • Updated referenced spec to NVMe 1.4. • Added note for editor to align new registers on four-byte boundaries. • Fixed typos. • Added comment on phrase that may need re-wording.
2019-05-09	Additional updates to NVMe 1.4: • Updated section 3.1 and Figure 68 to match NVMe 1.4. • Updated existing section 4.7 text to that in NVMe 1.4. Made changes from 2019-05-09 Technical WG meeting: • Resolved questions and deleted comments.
2019-07-03	Integration
2019-07-08	Review of Integration lead author’s requested CMBWBV field be changed to CMBWBZ to align with equivalent field for PMR**.**
2019-07-22	Ratified
4059a	

2020-04-16	Marking new NVMe registers as optional in section 3.1.1.
2020-04-20	Updates description of changes to indicate the registers being added are optional.
2020-04-30	Fixed figure numbering in section 3.1 to match NVMe 1.4.
2020-07-20	Integrated into the NVM Express Base Specification.

Description for NVMe 1.4 Changes Document

CMB Write Elasticity Status (optional)

- Two optional read-only registers are added, for the controller to indicate the CMB elasticity buffer size and CMB sustained throughput.
- Explanatory text is added to the CMB section describing the usage of the size and throughput.
- References:
  - NVMe 1.4 sections 3.1 and 4.7
  - Technical Proposal 4059

Description of Specification Changes

Markup Conventions:

Black:	Unchanged (however, hot links are removed)
~~Red Strikethrough~~:	Deleted
Blue:	New
Blue Highlighted:	TBD values, anchors, and links to be inserted in new text.
<Green Bracketed>:	Notes to editor

Modify portions of NVMe 1.4 as shown below:

3 Controller Registers

3.1 Register Definition

Figure 68 describes the register map for the controller.

The Vendor Specific address range starts after the last doorbell supported by the controller and continues to the end of the BAR0/1 supported range. The start of the Vendor Specific address range starts at the same location and is not dependent on the number of allocated doorbells.

Figure 68: Register Definition

Start	End	Symbol	Description
		…	
50h	57h	CMBMSC	Controller Memory Buffer Memory Space Control (Optional)
58h	5Bh	CMBSTS	Controller Memory Buffer Status (Optional)
5Ch	5Fh	CMBEBS	Controller Memory Buffer Elasticity Buffer Size (Optional)
60h	63h	CMBSWTP	Controller Memory Buffer Sustained Write Throughput (Optional)
~~5Ch~~ 64h	D~~E~~FFh	Reserved	Reserved
		…	

< **Note to Editor**: The two registers added above are on four-byte boundaries. >

…

3.1.TBD Offset 5Ch: CMBEBS – Controller Memory Buffer Elasticity Buffer Size

This optional register identifies to the host the size of the CMB elasticity buffer. A value of 0h in this register indicates to the host that no information regarding the presence or size of a CMB elasticity buffer is available.

Figure Fig_CMBEBS: Offset 5Ch: CMBEBS – Controller Memory Buffer Elasticity Buffer Size

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>31:8</td><td>RO</td><td>Impl Spec</td><td>CMB Elasticity Buffer Size Base (CMBWBZ): Indicates the size of the CMB elasticity buffer. The size of the CMB elasticity buffer is equal to the value in this field multiplied by the value specified by the CMB 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 CMB 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 CMB Elasticity Buffer), and this bit is:
a) set to ‘1’, then memory reads not conflicting with memory writes in the CMB Elasticity Buffer shall bypass those memory writes; and
b) cleared to ‘0’, then memory reads not conflicting with memory writes in the CMB Elasticity Buffer may bypass those memory writes.</td></tr>
<tr><td>3:0</td><td>RO</td><td>Impl Spec</td><td>CMB Elasticity Buffer Size Units (CMBSZU): Indicates the granularity of the CMB 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>1 KiB</td></tr>
<tr><td>2h</td><td>1 MiB</td></tr>
<tr><td>3h</td><td>1 GiB</td></tr>
<tr><td>4h – Fh</td><td>Reserved</td></tr>
</table></td></tr>
</table>

3.1.TBD+1 Offset 60h: CMBSWTP – Controller Memory Buffer Sustained Write Throughput

This optional register identifies to the host the maximum CMB sustained write throughput. A value of 0h in this register indicates to the host that no information regarding the CMB sustained write throughput is available.

Figure Fig_CMBSWTP: Offset 60h: CMBSWTP – Controller Memory Buffer Sustained Write Throughput

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>31:8</td><td>RO</td><td>Impl Spec</td><td>CMB Sustained Write Throughput (CMBSWTV): Indicates the sustained write throughput of the CMB 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 sustained write throughput of the CMB is equal to the value in this field multiplied by the units specified by the CMB 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>CMB Sustained Write Throughput Units (CMBSWTU): Indicates the granularity of the CMB 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>1 KiB/second</td></tr>
<tr><td>2h</td><td>1 MiB/second</td></tr>
<tr><td>3h</td><td>1 GiB/second</td></tr>
<tr><td>4h – Fh</td><td>Reserved</td></tr>
</table></td></tr>
</table>

4 Data Structures

…

4.7 Controller Memory Buffer

The Controller Memory Buffer (CMB) is a region of general purpose read/write memory on the controller. The controller indicates support for the CMB by setting CAP.CMBS to ‘1’. The host indicates intent to use the CMB by setting CMBMSC.CRE to ‘1’. Once this bit is set, the controller indicates the properties of the CMB via the CMBLOC and CMBSZ registers.

The CMB may be used for a variety of purposes. The controller indicates which purposes the memory may be used for by setting support flags in the CMBSZ register.

The CMB’s PCI Express address range is used for external memory read and write requests to the CMB. The PCI Express base address of the CMB is defined by the PCI Base Address Register (BAR) indicated by CMBLOC.BIR, and the offset indicated by CMBLOC.OFST. The size of the CMB is indicated by CMBSZ.SZ.

The controller uses the CMB’s controller address range to reference CMB with addresses supplied by the host. The PCI Express address range and the controller address range of the CMB may differ, but both ranges have the same size, and equivalent offsets within each range have a one-to-one correspondence. The host configures the controller address range via the CMBMSC register.

The host enables the CMB’s controller memory space via the CMBMSC.CMSE bit. When controller memory space is enabled, if host supplies an address referencing the CMB’s controller address range, then the controller directs memory read or write requests for this address to the CMB.

When the CMB’s controller memory space is disabled, the controller does not consider any host-supplied address to reference the CMB’s controller address range, and memory read and write requests are directed elsewhere (e.g., to memory other than the CMB).

To prevent possible misdirection of the controller’s memory requests, before host software enables the CMB’s controller memory space, it should configure the CMB’s controller address range to so that it does not overlap any address that host software intends to use for DMA.

In earlier versions of this specification, for a controller that supports the CMB, the CMB’s controller address range is fixed to be equal to its PCI Express address range, and the CMB’s controller memory space is always enabled whenever the controller is enabled. To prevent misdirection of controller memory requests when such a controller is assigned to a virtual machine, host software (on the hypervisor or host OS) should not enable translation of the CMB’s PCI Express address range, and it should ensure that this address range does not overlap any range of pre-translated addresses that the virtual machine may use for DMA.

Host software may configure CMBMSC so that CMB operates when the controller is assigned to a virtual machine that only supports versions 1.3 and earlier of this specification. To prevent that virtual machine from unintentionally clearing CMBMSC, the contents of CMBSMC are preserved across Controller Reset and Function Level Reset.

Submission Queues in host memory require the controller to perform a PCI Express read from host memory in order to fetch the queue entries. Submission Queues in controller memory enable host software to directly write the entire Submission Queue Entry to the controller's internal memory space, avoiding one read from the controller to the host. This approach reduces latency in command execution and improves efficiency in a PCI Express fabric topology that may include multiple switches. Similarly, PRP Lists or SGLs require separate fetches across the PCI Express fabric, which may be avoided by writing the PRP or SGL to the Controller Memory Buffer. Completion Queues in the Controller Memory Buffer may be used for peer to peer or other applications. For writes of small amounts of data, it may be advantageous to have the host write the data and/or metadata to the Controller Memory Buffer rather than have the controller fetch it from host memory.

The contents of the Controller Memory Buffer are initially undefined. Host software should initialize any memory before it is referenced (e.g., a Completion Queue shall be initialized by host software in order for the Phase Tag to be used correctly).

A CMB implementation has a maximum sustained write throughput. The CMB implementation may also have an optional write elasticity buffer used to buffer writes from CMB PCIe write requests. When the CMB 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 CMB sustained write throughput without back pressuring into the PCI Express fabric.

The time required to transfer data from the write elasticity buffer to the CMB is the amount of data written to the elasticity buffer divided by the Controller Memory Buffer Sustained Write Throughput (refer to section 3.1.TBD+1). The time to transfer the entire contents of the write elasticity buffer is the Controller Memory Buffer Elasticity Buffer Size (refer to section 3.1.TBD) divided by the Controller Memory Buffer Sustained Write Throughput.

A controller memory based queue is used in the same manner as a host memory based queue – the difference is the memory address used is located within the controller's own memory rather than in the host memory. The Admin or I/O Queues may be placed in the Controller Memory Buffer. If the CMBLOC.CQMMS bit (refer to Figure 84) is cleared to '0', then for a particular queue, all memory associated with it shall reside in either the Controller Memory Buffer or outside the Controller Memory Buffer.

If the CMBLOC.CQPDS bit (refer to Figure 84) is cleared to '0', then for all queues in the Controller Memory Buffer, the queue shall be physically contiguous.

The controller may support PRP Lists and SGLs in the Controller Memory Buffer. If the CMBLOC.CDPMLS bit (refer to Figure 84.) is cleared to '0', then for a particular PRP List or SGL associated with a single command, all memory associated with the PRP List or SGL shall be either entirely located in the Controller Memory Buffer or entirely located outside the Controller Memory Buffer.

PRP Lists and SGLs associated with a command may be placed in the Controller Memory Buffer if that command is present in a Submission Queue in the Controller Memory Buffer. If:

a) CMBLOC.CDPCILS bit (refer to Figure 84) is cleared to '0'; and
b) a command is not present in a Submission Queue in the Controller Memory Buffer,

then the PRP Lists and SGLs associated with that command shall not be placed in the Controller Memory Buffer.

The controller may support data and metadata in the Controller Memory Buffer. If the CMBLOC.CDMMMS bit (refer to Figure 84) is cleared to '0', then all data and metadata, if any, associated with a particular command shall be either entirely located in the Controller Memory Buffer or entirely located outside the Controller Memory Buffer.

If the requirements for the Controller Memory Buffer use are violated by the host, the controller shall fail the associated command with Invalid Use of Controller Memory Buffer status.

The address region allocated for the CMB shall be 4 KiB aligned. It is recommended that a controller allocate the CMB on an 8 KiB boundary. The controller shall support burst transactions up to the maximum payload size, support byte enables, and arbitrary byte alignment. The host shall ensure that all writes to the CMB that are needed for a command have been sent before updating the SQ Tail doorbell register. The Memory Write Request to the SQ Tail doorbell register shall not have the Relaxed Ordering bit set, to ensure that it arrives at the controller after all writes to the CMB.

...

<End of changes>