Technical Proposal ID	4076b – Zoned Random Write Area
Change Date	2023-02-01
Builds on Specification	NVM Express Zoned Namespaces Command Set Specification 1.1a
Refers to Technical Proposals	NVM Express 2.0 ECN 001

Technical Proposal Author(s)

Name	Company
Judy Brock, Bill Martin, Javier Gonzalez, Klaus B. Jensen	Samsung
Fred Knight	NetApp
Yoni Shternhell, Matias Bjørling	Western Digital
Paul Suhler	Kioxia

The Zoned Namespace Command Set added support for Zoned Namespaces (ZNS) which divides the logical address space of a namespace into zones, where each zone provides an LBA range that shall be written sequentially, and before being written again shall be explicitly reset (overwrites are not allowed).This technical proposal builds on the above.

This proposal defines a new area with a set of assigned LBAs which start at the write pointer for a given zone, called a Zone Random Write Area (ZRWA), in which the logical blocks that are mapped to that area may be written in non-sequential order as well as overwritten.

Address-Specific Write Commands to LBAs in the ZRWA may arrive out-of-order and are not required to be serialized (i.e., more than one may be outstanding concurrently). Logical blocks that have been written in the ZRWA are flushed to the associated zone using an implicit ZRWA Flush operation or explicit ZRWA Flush operation.

Revision History

Revision Date	Change Description
December 12, 2019	Initial version
January 05, 2020	Integrated feedback from Yoni (WDC) and David (Dell/EMC)

February 23, 2020	• Added optional support for Implicit Commit • Modified Explicit Commit to operate on one single zone rather than an array of zones • Sync up with current version of TP 4053 (last draft was based on fairly old ZNS TP draft) • Made Random Write Area Commit Granularity (ZRWACG) a 0's based value • Finishing a zone will not return an error if there were uncommitted logical blocks in the ZRWA. • Commit operation will not return an error if there were unwritten LBAs in the commit range. • Added break-out section on Commit Operations. • Zone Excursion will trigger autonomous commit. • Added Zone Random Write Area Capability (ZRWACAP) field to Identify Controller
March 23, 2020	• Sync up with 03/23/20 version of TP 4053 (member review candidate) • Moved Random Write Area Commit Granularity (ZRWACG) out of Identify Controller and into Zoned NS specific Identify NS because it's in blocks and has to change based on LBAF • Add optional Maximum Implicit Commit Write Size (MICWS) field. • The controller shall perform the Implicit Commit whether the write operation completes successfully or unsuccessfully. • indicate that the controller shall autonomously commit ZRWA when transitioning to states that release the ZRWA (Full, Empty, etc.) • Removed contradiction regarding handling of ZRWA allocation fail case. Eliminated text which allows this to succeed.
March 30,2020	• Specify that Append is optional if ZRWA is supported • Changed title block to Zoned Random Write Area (was Zoned Namespaces) • March 23rd draft changes accepted (except any outstanding comments) • Integrated Bill's notes from March 23 concall – closed on action required by some of them (some still to-do) • Added Figure TBD – Commit Operation.
April 10, 2020	• Added Kioxia as co-author. • Fixed two figure numbers. • Other minor editorial fixes. Accepted all format changes. • Changed "Implicit Commit Zone range" to "Implicit Commit Range".

April 14, 2020	• Permit zone in ZSIO:Implicitly Opened state to perform Commit operations due to following possible transition: Zone Open with ZRWAA=1 --> ZM:Close --> Write command causes transition to ZSIO:Implicitly Opened with ZRWAA=1 • Delete the RWANVMS field • MRWZ definition clarification (association is 1:1, not 1-to-many) • Added text indicating that write operations and commit operations are prohibited from crossing ZRWA boundary only if Implicit Commit is not enabled (i.e, limitation is for Explicit Commit only) • Removed action to add a log to report # of ZRWA resources available. Not possible in a multi-host environment as data can be stale by the time the log is read. • Permit a zone to acquire an association with a ZRWA when transitioning from ZSIO:Implicitly Opened w/no ZRWA to ZSEO:Explicitly Opened w/ZRWA.
April 20, 2020	• Moved figures into the order in which they appear in TP 4053. • Titled figures for Identify data structures as they are in TP 4053 Member Review revision. • Added headings for all modified sections. • Assigned the new section (Zone Random Write Area) heading 5.TBD. • Made Fig. 10 ZAMDS field optional, for consistency with Zone Append command being optional.
April 21, 2020	• Accepted all formatting changes to date. • Reviewed and accepted most of Pauls' 04/20 changes above • Editorial changes
April 28, 2020	• Made Implicit Commit mandatory. • Added references to Implicit Commit in sections that were written when only Explicit Commit existed. • Deleted support bit for Implicit Commit (IMPCOMSUP) as it is now mandatory to support it. • Reversed bit positions in Zoned Random Write Area Capabilities field per WG feedback • General cleanup/editorial changes • Added requirement for zone Empty state or WP Commit Granularity on allocation of a ZRWA (when zone is in Explicit Open or Implicit Open state)
May 12, 2020	• Major rewrite of Implicit Commit. • Removed Maximum Implicit Commit Write Size (MICWS) • Created definition for ZRWA boundary • Defined new "Invalid Operation Requested" status code • Replace Implicit Commit calculation of new WP with NetApp formula • Added ZRWA allocation error cases (E.G., cannot go from ZSEO w/o ZRWA to ZSEO w/ZRWA or vice versa; will result in Invalid Zone State Transition error.)
June 09, 2020	• Accepted most changes from May 12 draft version • Integrated comments from 05/26 ZNS WG meeting • Added additional Samsung contributors to Proposal Author(s) List • Changed new "Invalid Operation Requested" error status to "Invalid Zone Operation Request" • Modified definition of Commit

June 16, 2020	• Updated definition of Commit with version arrived at over the reflector by David, Fred, and myself.

June 23, 2020	• Incorporated a lot of feedback from 06/16 ZNS concall:
	  ○ Further massaging of definition of Commit
	  ○ Deleted ZRWA Boundary definition – talk about the end of the ZRWA rather than the ZRWA Boundary
	  ○ Rewrite of ZRWAA bit text
	  ○ Sync-up to ratified version of TP 4053
	  ○ Incorporate various other feedback

July 14, 2020	• Incorporated feedback from 06/23 meeting (last review) • Pull in comments from Matias • Made ZRWA pool namespace-based • Made all ZRWA-related capabilities namespace-based • Added new fields to support ZRWA pool resource management • Added new Feature ID for ZRWA pool resource management
July 21, 2020	• Accepted reviewed changes from last week • Incorporated some changes related to re-organizing text based on feedback from Matias
July 28, 2020	• Incorporated feedback from SKHynix (Santosh) and Seagate (Jim) • clarified text stating that a write command with a Starting LBA field that is not within the combined ZRWA and ICR address ranges shall be aborted. • Added text regarding the reading of ZRWA area • Removed text around zone transition to Empty state after autonomous commit (not valid) • Renamed Maximum Random Write Zones (MRWZ) to Number ZRWA (NUMZRWA) and simplified definition. • Delete Number ZRWAs Allocated to Namespace (NZAN) field – not needed. • Per team discussion, moved Zone Random Write Area Capability (ZRWACAP) back to Zoned Namespace Identify Controller data structure • general cleanup
August 4, 2020	• Accepted changes from last week. • Removed fields in Identify Namespace related to dynamic ZRWA Resource configuration. • Removed placeholder for Pool Configuration (Feature Identifier FID_TBDh • Per discussion with NetApp and WDC, moved Zone Random Write Area Capability (ZRWACAP) back to Zoned Namespace Identify Namespace data structure • Renamed Total Namespace ZRWAs to Number ZRWAs (NUMZRWA) • Renamed Zone Management Send Action "Commit Zone" to "Commit ZRWA"
August 5, 2020	• Accepted 8/4 changes above • Added comments from 08/04 WG review

August 7, 2020	• Incorporated changes and added comments to be reviewed by the team. • Removed early assignment of values and replaced with TBD's. • Updated wording to allow the Seq. Write Required zone type to be defined as is, before adding ZRWA to the logic. • Modified text to add ZRWA resources as a common resource pool similarly defined as Active and Open Resources. • Reordered fields in the Identify Namespace data structure. • Renamed Number ZRWAs to "Maximum ZRWA Resources" and aligned the description. • Moved bit to identify ZRWA being support in the OZCS field. • Updated Attach ZRWA description in the Zone Management Send command and move into generic Zone Send Action Specific Features field. • Propose to move ZRWA resource text to zone resource section.
August 10, 2020	• Added and updated drawings and related text. • Fixed incorrect numbers in existing figure captions. • Figure 21: Added preceding paragraph (missing) from TP 4053 preratification revision. • Figure 29: Changed definition of SLBA field to a sub-table, per Matias's request.
August 11, 2020	• Add definition for ZRWA • Retain definition for Commit • Reject "Renamed Number ZRWAs to "Maximum ZRWA Resources" and aligned the description.". Number ZRWA definition much cleaner. • Reject change to support ZRWA Allocation during Set Zone Extension Action, not needed
August 12, 2020	• Integrate feedback from August 11 conference call • significant changes or additions to multiple sections (Open Zone ZRWA Allocation, Manage Resources, Zone Random Write Area)
August 23, 2020	• Lots of cleanup (changes accepted, comments deleted) of sections we have reviewed/discussed as a group.
August 25, 2020	• Renamed Zone Management Send Action "Commit ZRWA" to "ZRWA Flush", Implicit Commit to implicit ZRWA Flush, Explicit Commit to explicit ZRWA Flush, and ZRWA Commit Granularity to ZRWA Flush Granularity. • Rewrote ZRWA Flush Granularity (formerly ZRWA Commit Granularity) definition for clarity
August 27, 2020	• Accepted changes from August 25 plus new WG comments • Renamed EXPCOMSUP to EXPFLUSHSUP • Eliminated all references to "commit", "committed", etc. – replaced with references to ZRWA flush • Corrected the text indicating when implicit flush is triggered: it is triggered by write beyond the start - not the end - of the Implicit ZRWA Flush Range.
September 2, 2020	• Merge in latest feedback from Matias minus various comments (to be discussed in conference call)
September 6, 2020	• Integrate more feedback from Matias
September 21, 2020	• Still more feedback processing
September 22, 2020	• Disposition of comments labelled "Phase 2"

September 23, 2020	• Moved EXPFLUSHSUP in ZRWACAP field to bit 0 (was previously at bit 1)
October 06, 2020	• Incorporated WG comments from 09/23 call • Removed support for transition from ZSEO: Explicitly Opened w/o ZRWA to ZSEO: Explicitly Opened with ZRWA • Corrected one "Commit Zone" to "ZRWA Flush" • Changed 2.5 heading to match TP 4053a. • Fixed headings • Updated figures from "commit" to "flush". • In equation, changed "CG" to "FG".
October 10, 2020	• Removed "**Reading in Sequential Write Required Zones"** section as ZRWA considerations have now been addressed in common text TP 4053a • Rewrite of Open Zone ZRWA Allocation/ ZRWAA bit text section. • Removed text which allowed transitioning from Closed state w/o ZRWA to Explicitly Opened state with ZRWA. This was a remnant from the fact that previously a zone could have been opened without a ZRWA, then closed, and then explicitly opened with a ZRWA. Since we are disallowing open w/o ZRWA transition to open w/ZRWA, we should disallow the hop also. • Integrate comments from 10/07 conference call. • Modified text describing Send Action ZRWA Flush or Open w/ZRWA when ZRWA Support bit is cleared to '0'
October 14, 2020	• Integrated comments/modifications from (and reviewed in) 10/13/20 walk-thru
October 27, 2020	• Integrated comments/modifications from 10/20/20 walk-thru • Integrated feedback from Mike Allison (Intel)
November 10, 2020	• Sync up with TP 4053a dated 2020.11.03a • Incorporate feedback from NetApp (Fred Knight) • Added definition: **IOCSSIN Data Structure**
November 13, 2020	• Changes from 2020-11-13 *ad hoc* meeting: • Rewrite of the implicit ZRWA flush section. • Added figure showing ZRWA at the end of the zone.
November 17, 2020	• Changes from 2020-11-17 ZNS Task Group meeting: • Editorial changes to section 5.TBD.3.
December 01.2020	• Incorporated changes from 11-10-20 meeting from Judy's local draft • Accepted various changes from past reviews
December 08.2020	• Lots of misc. cleanup
December 15, 2020	• Integration of comments from 12/08 ZNS Task group meeting; additional comments from WDC
January 05, 2021	• Accepted misc. reviewed changes
January 11, 2021	• Added comments from Seagate (Gerry Houlder) • Added revised figures from Kioxia (Paul Suhler) • Incorporated feedback from 2021-01-05 ZNS WG concall

January 19, 2021	• Split 2.5.1 into 2 sections – one that covers active/open and one that covers ZRWA • Integrated feedback from 2021-01-12 ZNS WG call • Lots of cleanup, resolved lots of comments • Added definition for ceiling
February 01, 2021	• Integrate Jan 19 feedback • Created definition for ceiling
February 16, 2021	• Integrate Feb 01 and Feb 09 feedback from ongoing walk-thru
February 17, 2021	• Modify Figure 31 and Figure 47 • Integrate Feb 16 comments from ZNS WG call
February 23, 2021	• Start walkthrough of Phase 3 comments
March 02, 2021	• Incorporate feedback from Feb 23 ZNS WG call • Continue walkthrough of Phase 3 comments
March 09, 2021	• Renamed Zone Management Send Action "ZRWA Flush" to "Flush Explicit ZRWA Range" to eliminate overloaded terminology ("ZRWA Flush" was both an umbrella term encompassing both implicit and explicit ZRWA Flush as well as a specific Zone Send Action) • Incorporated feedback from March 02 ZNS WG call • Closed various long-outstanding comments
March 16, 2021	• Incorporate feedback from March 09 ZNS WG call • Make NumZRWA 0-based to align with MOR, MAR • Defined IZFR (Implicit ZRWA Flush Region) • Pulled in Matias' comments; addressed most of them (not all yet)
April 5, 2021	• Added changes to section A.5 Capacity and Sizes • Incorporate feedback from March 16 ZNS WG call
April 9, 2021	• Incorporate feedback from April 5 ZNS WG call
April 20, 2021	• Incorporate feedback from Matias
April 27, 2021	• Started to convert to ZNS Command Set Spec v1.1 (section #'s, figure #'s, etc.) – still in progress
May 4,2021	• Incorporate most of section **2.1.1.2.1.1** rewrite from Matias; accept changes from 04/27 meeting
May 5, 2021	• Accept changes from 05/04 meeting • Ported doc to sync up to baseline ZNS Command Set specification v1.1, date 05/04. Was previously based on TP 4053a. • Changed copyright to 2020-2021
May 11,2021	• Integrated feedback from Mike A. (Intel) • Deleted resolved comments • Accepted all changes in preparation of member review draft
June 10, 2021	• Resolved various member review comments, addressed editorial and technical feedback • ZSC to ZSIO on explicit ZRWA Flush operation • Added ability to associate with both ZRWA and Zone Descriptor Extension concurrently
June 14, 2021	• Reorganized changes to two sentences in section **"Open Zone ZRWA Allocation"** from Fred

June 15, 2021	• Minor editorial fixes (extra spaces corrected) • Modified transition ZSC:ZSIO to include reference to explicit ZRWA Flush operation as one of the causes of such a transition
June 17, 2021	• Integrated editorial feedback from 6/15 walk-thru • Accepted Fred's re-org of "Managing Resources" section
June 24, 2021	• Accept all changes in preparation for re-start of 30 day review
July 22, 2021	• Integrated feedback from Mike A. • Updated Figure FigFlush: Flush Operation with new figure from Paul S. to correct acronym for ZRWA Size
July 27, 2021	• Integrate feedback from ZNS WG 07/27 walk-thru of 07/22 draft • Accept all changes in preparation for integration
August 11, 2021	• Integrated into the NVMe Zoned Namespaces Command Set Specification, revision 1.1.
August 17, 2021	• Corrected the bytes allotted to the ZRWACAP field to be 1 byte. • Corrected the value for the Invalid Zone Operation Request status code in figure 30.
October 20, 2021	Start TP4076a: • Added the word "resources" in the Number of ZRWA Resources field definition. • Changed "Zone Send Action command" to "Zone Send Action" in SLBA field of Zone Management Send • Abort the Zone Management Send command if ZRWA Supported bit is cleared to '0' and the Zone Send Action field specifies Set Zone Descriptor Extension with the ZRWAA bit set to '1' • In ZRWASUP field, replace "feature" with "capability" • Wordsmith text on setting the value of the write pointer at command completion for Flush Explicit ZRWA Range • Added detailed Description of Changes section
November 17, 2021	• Removed the first instance of the word "command" in sentence starting with "On successful command completion of a command for which …" • Added missing period • Accepted all changes in preparation for 30 day member review
November 29, 2021	• Addressed Mike Allison's member review comments
January 19, 2022	• Removed all Comments.
October 09, 2022	Start TP4076b: • Correct off-by-one error in reference to last LBA in IZFR • Modify section 2.1.1.3 to take into consideration whether there was a ZRWA attached at the time the state machine for a zone is initialized/re-initialized • Update footer to 2022 (was 2021) • Editorial updates from WG review
January 25, 2023	• Integrated
February 01, 2023	• Addressed Mike Allison and Judy Brock review comments

Description of changes for NVM Express Zoned Namespace Command Set Specification 1.1a:

- Added new definitions for Zone Random Write Area (ZRWA) and Ceiling
- Zone Append may be used to write to logical blocks in a zone only if no ZRWA is associated with the zone
- For a zone associated with a ZRWA, command completion processing is defined in a new section
- Added another ZSC:ZSIO transition case – when an explicit ZRWA Flush operation is completed
- Added a new type of zone resource called ZRWA resource
- Added resource management rules for ZRWA resources
- Added two new Command Specific status codes: Invalid Zone Operation Request and ZRWA Resources Unavailable
- Added new Zone Descriptor Data Structure field called Zone Random Write Area Valid (ZRWAV)
- Added new Zone Management Send field called Zone Send Action Specific Option which controls the allocation of a ZRWA to a zone.
- Added a new Zone Send Action type in Zone Management Send command called Flush Explicit ZRWA Range.
- Added new Open Zone sub-section called Open Zone ZRWA Allocation
- Zone Management Send may return two new command specific status codes: Invalid Zone Operation Request and ZRWA Resources Unavailable
- Added new section to describe Flush Explicit ZRWA Range
- Added new ZRWA Supported (ZRWASUP) bit in the OZCS field of the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set
- Added the following new fields to the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set:
  - Number of ZRWA Resources (NUMZRWA)
  - ZRWA Flush Granularity (ZRWAFG)
  - ZRWA Size (ZRWASZ)
  - ZRWA Capability (ZRWACAP)
    - EXPFLUSHSUP bit within this field which defines support for explicit ZRWA Flush operations
- Added a model clause section for Zone Random Write Area (see section 5.TBD)

- **New requirement / incompatible change:**
  - A Zone Append command shall not be sent to a zone while that zone is associated with a ZRWA
  - Definition of Starting LBA (SLBA) field in Zone Management Send command is changed if the Zone Send Action is Flush Explicit ZRWA Range
  - Added new requirement to Reset Zone: the Zone Random Write Area Valid bit shall be cleared to '0'
  - New qualification: writes are required to start at the valid write pointer address only if a ZRWA is not associated with a zone
- **References:**
  - Section 1.4
  - Section 2.1.1.2.1.1
  - Section 2.1.1.3
  - Section 2.1.1.4
  - Section 3.1.2
  - Section 3.4.1
    - Figure 30
  - Section 3.4.2.2.3
    - Figure 37
  - Section 3.4.3.1.3.TBD
  - Section 4.1.5.1
    - Figure 48
  - TP 4076a

Markup Conventions:

Black: Unchanged (however, hot links are removed)

~~Red Strikethrough~~:   Deleted

Blue: New

Green: Moved

<mark>Highlighted</mark>:   TBD values, anchors, and links to be inserted.

Orange Bracketed: **Notes to editor**

1.4   Definitions

Add the following sections as shown below:

<mark>1.4.TBD_1</mark> Zone Random Write Area (ZRWA)

An area of non-volatile storage with a sliding set of assigned LBAs which start at the write pointer for the associated zone.

<mark>1.4.TBD_2</mark> Ceiling

A function where, for real number x, ceiling(x) is equal to x rounded up to the nearest integer value. If x is an integer then ceiling(x) is equal to x.

2   Zoned Namespace Command Set Model

[…snip…]

Modify section 2.1.1.2.1.1 Writing in Sequential Write Required Zones as follows:

2.1.1.2.1.1 Writing in Sequential Write Required Zones

~~User Data Out~~ The following commands may be used to write to logical blocks in a specific zone of zone type Sequential Write Required~~.~~:

- Write;
- Write Zeroes;
- Write Uncorrectable;
- Copy; and
- Zone Append, if no ZRWA is associated with the zone.

A write pointer is maintained for each zone in the zoned namespace that indicates the ~~next~~ lowest-numbered writeable logical block address in that zone. The write pointer is valid for a subset of the zone states as defined in Figure 4.

 […snip…]

The host may use the Zone Management Receive command to determine the current write pointer for a zone.

~~The write pointer for a zone in the ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state shall be increased by the number of logical blocks written on successful completion of a write operation.~~
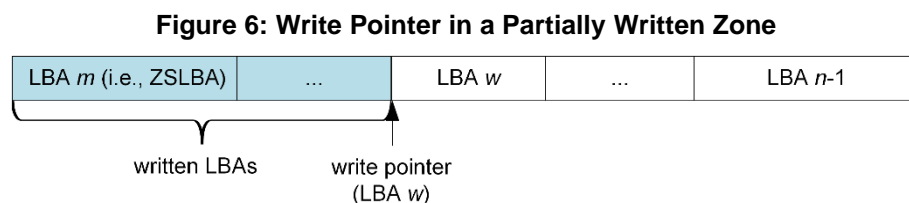
~~If the controller is not able to successfully write to all logical blocks specified by a User Data Out Command, then the write pointer shall:~~
~~a) be set to a value within the range of LBAs specified in that User Data Out Command ;~~
~~b) be set to one greater than the last LBA in the range of LBAs specified in that User Data Out Command; or~~
~~c) become invalid (i.e., due to transitioning to the ZSRO:Read Only state or the ZSO:Offline state, or due to a Zone Active Excursion (refer to section 5.6)).~~

The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone.

[…snip…]

Figure 6 shows an example of a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state, that has had ~~some~~ one or more logical blocks written. The write pointer, indicated by LBA w, is the lowest-numbered ~~unwritten~~ writeable LBA ~~(i.e., the next LBA to be written)~~ and n-1 is the highest-numbered LBA of the zone.

Figure 6: Write Pointer in a Partially Written Zone

LBA m (i.e., ZSLBA)	...	LBA w	...	LBA n-1

written LBAs   write pointer
(LBA w)

The controller shall abort a command that writes to a zone that is in the ZSF:Full state with a status code of Zone Is Full.

The controller shall abort a command that writes to a zone that is in the ZSRO:Read Only state, with a status code of Zone Is Read Only.

The controller shall abort a command that writes to a zone that is in the ZSO:Offline state, with a status code of Zone Is Offline.

~~For a zone in the ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state:~~

For a zone associated with a ZRWA (refer to section 5.TBD), commands shall be processed as defined in section 5.TBD.

For a zone not associated with a ZRWA and which has a valid write pointer, commands shall be processed as follows:

a) if the controller:
   a. successfully writes to all logical blocks specified by a write operation, then the write pointer shall be increased by the number of logical blocks written on successful completion of that write operation; and
   b. is not able to successfully write to all logical blocks specified by a write operation, then the write pointer shall:
      i. be set to a value within the range of LBAs specified in that write operation;
      ii. be set to one greater than the last LBA in the range of LBAs specified in that write operation; or
      iii. become invalid (i.e., due to transitioning to the ZSRO:Read Only state or the ZSO:Offline state, or due to a Zone Active Excursion (refer to section 5.6));

b) if an Address-Specific Write Command specifies a Starting LBA field that is not equal to the write pointer for that zone, then the controller shall abort that command with a status code of Zone Invalid Write; and

c) if a Zone Append command specifies a ZSLBA that is not the lowest logical block address in that zone, then that command is aborted with a status code of Invalid Field in Command, as described in section 3.4.1.

The controller shall abort a command that initiates a write operation that has a starting LBA in one zone for which the Number of Logical blocks exceeds the remaining number of logical blocks in that zone, with a status code of Zone Boundary Error.

Modify section 2.1.1.3 as shown below:

2.1.1.3 Zone State Machine

…

The initial state for each zone is the:

a) ZSE:Empty state if:~~,~~
   a) ~~if~~ the write pointer is valid;~~,~~
   b) the write pointer points to the lowest LBA in the zone;~~, and~~
   c) the Zone Descriptor Extension Valid bit is cleared to '0'; and
   d) the ZRWA, if any, associated with the zone contains no user data;

b) ZSC:Closed state:~~a)~~ if the write pointer is valid and:
   a) the write pointer does not point to the lowest LBA in the zone; ~~or~~
   b) ~~if the write pointer is valid and~~ the Zone Descriptor Extension Valid bit is set to '1'; or
   c) the ZRWA, if any, associated with the zone contains user data;

c) ZSF:Full state if:
   a) ~~if~~ the most recent state was the ZSF:Full; or
   b) ~~if~~ the zone state was transitioned to the ZSF:Full state as a result of the NVM Subsystem Reset;

d) ZSRO:Read Only state, if the most recent zone state was the ZSRO:Read Only state; and

e) ZSO:Offline state, if the most recent zone state was the ZSO:Offline state.

Modify section 2.1.1.3.4 Zone Resources as shown below:

2.1.1.3.4   ZSC:Closed state

…

Transition ZSC:ZSIO: The zone shall transition from the ZSC:Closed state to the ZSIO:Implicitly Opened state, if there are available resources as defined in section 2.1.1.4 and:

    a) a write operation writes one or more logical blocks of that zone; or
    b) an explicit ZRWA Flush operation is completed.

Modify section 2.1.1.4 Zone Resources as shown below:

2.1.1.4   Zone Resources

Zones may have associated Active Resources, and associated Open Resources, and associated ZRWA resources. The resources limit the number of zones that are allowed to be in each zone state. The resource relationship is defined in Figure 8.

Figure 8: Zone Resources

Resource	States	Comment
Active	ZSIO:Implicitly Opened, ZSEO:Explicitly Opened, ZSC:Closed	Zones in zone states associated with this resource are limited by the Maximum Active Resources field.
Open	ZSIO:Implicitly Opened, ZSEO:Explicitly Opened	Zones in zone states associated with this resource are limited by the Maximum Open Resources field.
ZRWA	ZSIO:Implicitly Opened, ZSEO:Explicitly Opened, ZSC:Closed	Zones in zone states associated with this resource are limited by the Number of ZRWA Resources field. Refer to section 5.TBD.

Zones that have associated Open Resources are a subset of the zones that are associated with Active Resources. The Maximum Open Resources field shall be less than or equal to the Maximum Active Resources field in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set (refer to Figure 48).

If the ZRWA Supported bit is set to '1' in the OZCS field in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set, then the Number of ZRWA Resources field in that data structure shall be less than or equal to the Maximum Active Resources field in that data structure.

Modify section 2.1.1.4.1 as shown below

2.1.1.4.1   Managing resources

The controller associates Active Resources with zones in the ZSIO:Implicitly Opened, the ZSEO:Explicitly Opened, and the ZSC:Closed states. The controller associates Open Resources with zones in the ZSIO:Implicitly Opened state and the ZSEO:Explicitly Opened state.

The rResource management for active and open resources is as follows:

    a) A transition from the ZSE:Empty state to the ZSIO:Implicitly Opened state or the
        ZSEO:Explicitly Opened state increases the resource usage of Active Resources and Open
        Resources by 1.

b) **...[snip]…**
g) A transition from the ZSE:Empty state to the ZSF:Full state shall not impact resource usage of Active Resources and Open Resources

A controller processing a command that requests a zone to transition to the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state shall, if resources are not available and:

a) the Maximum Active Resources field is greater than the Maximum Open Resources field and:
   i. as a result of the requested transition the resource that is not available is Active Resources, then abort the command with a status code of Too Many Active Zones; or
   ii. as a result of the requested transition the resource that is not available is Open Resources, then abort the command with a status code of Too Many Open Zones;

   or

b) the Maximum Active Resources field is equal to the Maximum Open Resources field and as a result of the requested transition the resource that is not available is Active Resources, then abort the command with a status code of Too Many Active Zones.

If the ZRWA Supported bit is set to '1' in the OZCS field in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set, then as described in Figure 8, the controller associates ZRWA Resources with zones in the ZSIO:Implicitly Opened, the ZSEO:Explicitly Opened, and the ZSC:Closed states.

Resource management for ZRWA Resources is as follows:

a) A successful allocation of a ZRWA to a zone as a result of the Zone Management Send command (refer to section 3.4.3) increases the resource usage of ZRWA Resources by 1.
b) For a zone with an associated ZRWA Resource, a transition to the ZSE:Empty state, the ZSF:Full state, the ZSRO:Read Only state, or the ZSO:Offline state decreases the resource usage of ZRWA Resources by 1.

A controller that supports ZRWA resources and that processes a command that requests a ZRWA Resource shall, if ZRWA Resources are not available due to the number of ZRWA Resources in use being equal to the Number of ZRWA Resources field, abort that command with a status code of ZRWA Resources Unavailable.

Zones that have associated Active Resources are transitioned to the ZSF:Full state when the zoned namespace becomes write protected. Refer to the Namespace Write Protection section in the NVMe Express Base sSpecification.

The controller may transition zones in the ZSIO:Implicitly Opened state to the ZSC:Closed state for resource management purposes.

The coordination of host software usage of resources associated with shared namespaces is outside the scope of this specification.

Modify section 3.1.2 as shown below

3.1.2   Command Specific Status Values

This specification supports the Command Specific status values defined in the NVMe Express Base Specification and in the NVM Command Set Specification. Command Specific status values that are specific to the Zoned Namespace Command Set Specification are defined in this section.

Figure 11 defines the status values specific to the Zoned Namespace Command Set.

Figure 11: Status Code – Command Specific Status Values, Zoned Namespace Command Set

Value	Description	Commands Affected
B6h	Invalid Zone Operation Request	Zone Append, Zone Management Send
B7h	ZRWA Resources Unavailable	Zone Management Send
B8h	Zone Boundary Error	Compare[1], Copy[1], Read[1], Verify[1], Write, Write Uncorrectable, Write Zeroes, Zone Append
B9h	Zone Is Full	Copy, Write, Write Uncorrectable, Write Zeroes, Zone Append
BAh	Zone Is Read Only	Copy, Write, Write Uncorrectable, Write Zeroes, Zone Append, Zone Management Send
BBh	Zone Is Offline	Compare, Copy, Read, Verify, Write, Write Uncorrectable, Write Zeroes, Zone Append, Zone Management Send
BCh	Zone Invalid Write	Copy, Write, Write Uncorrectable, Write Zeroes
BDh	Too Many Active Zones	Copy, Write, Write Uncorrectable, Write Zeroes, Zone Append, Zone Management Send
BEh	Too Many Open Zones	Copy, Write, Write Uncorrectable, Write Zeroes, Zone Append, Zone Management Send
BFh	Invalid Zone State Transition	Zone Management Send
NOTES:		
1. This command is affected if the Read Across Zone Boundaries bit is cleared to '0' in the Zoned Namespace Command Set specific Identify Namespace data structure (refer to section 4.1.5.1) is cleared to '0'.		

Modify section 3.4.1 Zone Append Command as follows:

3.4.1 Zone Append command

[…snip…]

If the ZSLBA field in the Zone Append command does not specify the lowest logical block for a zone, then the command shall be aborted with a status code of Invalid Field in Command.

If a Zone Append command is received for a zone while that zone is associated with a ZRWA, then the controller shall abort the command with a status code of Invalid Zone Operation Request.

The AWUN, NAWUN, NABSN, AWUPF, NAWUPF, NABSPF atomicity parameters apply as defined in the Atomic Operations section in the NVM Command Set Specification to the Zone Append command.

Modify Figure 30: Zone Append – Command Specific Status Values as follows:

Figure 30: Zone Append – Command Specific Status Values

Value	Description
B6h	**Invalid Zone Operation Request:** The operation requested is invalid due to having received a Zone Append command for a zone associated with a ZRWA.
B8h	**Zone Boundary Error:** The command specifies logical blocks in more than one zone.
B9h	**Zone Is Full:** The accessed zone is in the ZSF:Full state.
BAh	**Zone Is Read Only:** The accessed zone is in the ZSRO:Read Only state.
BBh	**Zone Is Offline:** The accessed zone is in the ZSO:Offline state.
BDh	**Too Many Active Zones:** The controller does not allow additional active zones.
BEh	**Too Many Open Zones:** The controller does not allow additional open zones.

Modify section 3.4.2.2.3 Zone Descriptor Data Structure as shown below:

3.4.2.2.3 Zone Descriptor Data Structure

Figure 37 defines the Zone Descriptor data structure.

Modify Figure 37: Zone Descriptor Data Structure as follows:

Figure 37: Zone Descriptor Data Structure

Bytes	Description
…	…
02	**Zone Attributes (ZA):** Indicates attributes for the Zone:

Bits	Description
7	**Zone Descriptor Extension Valid (ZDEV)**: If this bit is set to '1', then Zone Descriptor Extension data is associated with the zone. If this bit is cleared to '0', then no Zone Descriptor Extension data is associated with the zone. Refer to section 5.1.
6:4	Reserved
3	**Zone Random Write Area Valid (ZRWAV):** If this bit is set to '1', then there is a ZRWA (refer to section 5.TBD) associated with the zone. If this bit is cleared to '0', then there is no ZRWA associated with the zone.
2	**Reset Zone Recommended (RZR):** If this bit is set to '1', the controller recommends that the zone is reset. Refer to section 5.4
1	**Finish Zone Recommended (FZR):** If this bit is set to '1', then controller recommends that the host transitions this zone to the ZSF:Full state. Refer to section 5.5.
0	**Zone Finished by Controller (ZFC):** The controller shall set this bit to '1' when the controller autonomously transitioned the zone to ZSF:Full state. The controller shall clear this bit to '0' on any other zone state transition. Refer to section 5.6.

| … | … |

Modify section 3.4.3 Zone Management Send command as shown below:

3.4.3   Zone Management Send command

[…snip…]

Figure 38: Zone Management Send – Command Dword 10 and Command Dword 11

Bits	Description
63:00	**Starting LBA (SLBA):** ~~This field specifies the lowest LBA of the zone on which the Zone Send Action is performed. Command Dword 10 contains bits 31:00 of the SLBA; Command Dword 11 contains bits 63:32 of the SLBA.~~ The definition of this field depends on the Zone Send Action, as shown in the following table:

Zone Send Action	Description
Flush Explicit ZRWA Range	This field specifies the LBA of the highest-numbered logical block that is requested to be flushed. Command Dword 10 contains bits 31:00 of the SLBA; Command Dword 11 contains bits 63:32 of the SLBA (refer to section 3.4.3.1.TBD).
All others	This field specifies the LBA of the lowest-numbered logical block of the zone on which the Zone Send Action is performed. Command Dword 10 contains bits 31:00 of the SLBA; Command Dword 11 contains bits 63:32 of the SLBA.

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:~~09~~10	Reserved

Bits	Description
09	**Zone Send Action Specific Option (ZSASO):** <table><tr><td>**Zone Send Action**</td><td>**Description**</td></tr><tr><td>Open Zone</td><td rowspan="2">**ZRWA Allocation (ZRWAA):** This bit controls the allocation of a Zone Random Write Area to a zone. Allocation of a ZRWA resource operates as described in section 3.4.3.1.3.TBD.</td></tr><tr><td>Set Zone Descriptor Extension</td></tr><tr><td>All others</td><td>Reserved</td></tr></table>
08	**Select All**: If this bit is set to '1', then the SLBA field shall be ignored. If this bit is cleared to '0', then the SLBA field specifies the lowest logical block of the zone. Refer to section 3.4.3.1 for specific behavior for each Zone Send Action.
07:00	**Zone Send Action (ZSA):** Defines the zone action to be performed for Zone Management Send. <table><tr><td>**Value**</td><td>**Description**</td><td>**Refer to section**</td></tr><tr><td>00h</td><td>Reserved</td><td></td></tr><tr><td>01h</td><td>**Close Zone:** Close one or more zones.</td><td>3.4.3.1.1</td></tr><tr><td>02h</td><td>**Finish Zone:** Finish one or more zones.</td><td>3.4.3.1.2</td></tr><tr><td>03h</td><td>**Open Zone:** Open one or more zones.</td><td>3.4.3.1.3</td></tr><tr><td>04h</td><td>**Reset Zone:** Reset one or more zones.</td><td>3.4.3.1.4</td></tr><tr><td>05h</td><td>**Offline Zone:** Offline one or more zones.</td><td>3.4.3.1.5</td></tr><tr><td>06h to 0Fh</td><td>Reserved</td><td></td></tr><tr><td>10h</td><td>**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.</td><td>3.4.3.1.6</td></tr><tr><td>11h</td><td>**Flush Explicit ZRWA Range:** Flushes a range of logical blocks from a ZRWA to a zone.</td><td>3.4.3.1.TBD</td></tr><tr><td>~~11h~~ 12h to FFh</td><td>Reserved</td><td></td></tr></table>

[…snip…]

If there are insufficient available Active Resources, ~~or~~ insufficient available Open Resources, or insufficient available ZRWA resources, then the command shall be aborted as defined in section 2.1.1.4, and no zone state transition shall occur.

If the command SLBA field does not specify the starting logical block for a zone in the specified zoned namespace, the Zone Send Action does not specify Flush Explicit ZRWA Range, and the Select All bit is cleared to '0', then the command shall be aborted with a status code of Invalid Field in Command.

If the Zone Send Action field specifies Set Zone Descriptor Extension, and the Zone Descriptor Extension Size field value in the Zoned Namespace Command Set specific Identify Namespace data structure is cleared to 0h, then the command shall be aborted with a status code of Invalid Field in Command.

If the ZRWA Supported bit is cleared to '0' in the OZCS field, then the command shall be aborted with a status code of Invalid Zone Operation Request if the Zone Send Action field specifies:

   a) Open Zone with the ZRWAA bit set to '1';
   b) Set Zone Descriptor Extension with the ZRWAA bit set to '1'; or
   c) Flush Explicit ZRWA Range.

If the Zone Send Action field specifies Flush Explicit ZRWA Range and the EXPFLUSHSUP bit is cleared to '0' in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set, then the controller shall abort the command with a status code of Invalid Zone Operation Request.

For a shared zoned namespace, the method used by hosts to coordinate Zone Management Send commands is outside the scope of the specification.

Modify section 3.4.3.1.3 Open Zone as follows:

If the Select All bit is cleared to '0' in Command Dword 13 ~~is cleared to '0',~~ and the zone specified by the SLBA field is in:

a) the ZSE:Empty state, the ZSIO:Implicitly Opened state or the ZSC:Closed state, then the zone should be transitioned to the ZSEO:Explicitly Opened state;
b) the ZSEO:Explicitly Opened state, then no change shall be made to the zone state; and
c) the ZSF:Full state, the ZSRO:Read Only state or the ZSO:Offline state, then the controller shall abort the command with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the ZSC:Closed state should be transitioned to the ZSEO:Explicitly Opened state.

If there are insufficient available Active Resources or insufficient available Open Resources, then the command shall be aborted as described in section 2.1.1.4 and no zone state transition shall occur.

Add new Open Zone sub-section 3.4.3.1.3.TBD Open Zone ZRWA Allocation as follows:

3.4.3.1.3.TBD   Open Zone ZRWA Allocation

The Zone Random Write Area Allocation (ZRWAA) bit in the Zone Send Action Specific Features field controls the allocation of a ZRWA to a zone in a Zone Management Send command when the Zone Send Action is set to Open Zone or set to Set Zone Descriptor Extension.

If the ZRWAA bit is cleared to '0' or a ZRWA is associated with this zone, then the processing of the command shall have no effect on ZRWA resources.

If the ZRWAA bit is set to '1' and there is no ZRWA associated with the zone, then:

- if no ZRWA resources are available, then the controller shall abort the command with a status code of ZRWA Resources Unavailable and no zone state transition shall occur; and
- if the zone is not in the ZSE:Empty state, then the controller shall abort the command with a status code of Invalid Zone Operation Request and no zone state transition shall occur.

A ZRWA shall be allocated to the specified zone upon successful completion of the Zone Management Send command if:

- the ZRWAA bit is set to '1';
- a ZRWA resource is available (refer to Number of ZRWA Resources field and section 2.1.1.4.1;
- no ZRWA is currently associated with the specified zone;
- the Zone Send Action is set to Open Zone or set to Set Zone Descriptor Extension; and
- the zone is in the ZSE:Empty state.

Modify section 3.4.3.1.4 Reset Zone as follows:

3.4.3.1.4 Reset Zone

[…snip…]

If the command completes successfully, then for each affected zone:

a) the Write Pointer zone attribute in the Zone Descriptor shall be set to the ZSLBA of the zone; and
b) the following zone attribute bits in the Zone Descriptor shall be cleared to '0':

    a) Zone Descriptor Extension Valid;
    b) Finish Zone Recommended;
    c) Reset Zone Recommended; ~~and~~
    d) Zone Finished by Controller ~~.~~;and
    e) Zone Random Write Area Valid.

Modify Figure 40 (Zone Management Send – Command Specific Status Values) as follows:

Figure 40: Zone Management Send – Command Specific Status Values

Value	Description
B6h	**Invalid Zone Operation Request:** The operation requested is invalid. This may be due to various conditions, including: • attempting to allocate a ZRWA when a zone is not in the ZSE:Empty state; or • invalid Flush Explicit ZRWA Range Send Zone Action operation.
B7h	**ZRWA Resources Unavailable:** No ZRWAs are available.
BAh	**Zone Is Read Only:** Zone is in the ZSRO:Read Only state. This may have occurred during the processing of the command.
BBh	**Zone Is Offline:** Zone is in the ZSO:Offline state. This may have occurred during the processing of the command.
BDh	**Too Many Active Zones:** The controller does not allow additional active zones.
BEh	**Too Many Open Zones:** The controller does not allow additional open zones.
BFh	**Invalid Zone State Transition:** The request is not a valid zone state transition.

Add the following section as shown below:

3.4.3.1.TBD Flush Explicit ZRWA Range

The Zone Send Action of Flush Explicit ZRWA Range is a request to move data and metadata to the zone specified by the SLBA field. The logical blocks in the specified LBA range are written from the ZRWA associated with that zone to the zone.

The range of logical blocks to be flushed starts at the write pointer and ends at the LBA specified by the SLBA field in Command Dword 10 and Command Dword 11. On successful completion of a command for which the SLBA is not the highest-numbered LBA of the zone, the write pointer for that zone shall be set to one greater than the value in the SLBA field.

The number of logical blocks in the specified LBA range shall be a multiple of ZRWAFG. If the number of logical blocks to be flushed is not an integral multiple of ZRWAFG, then the controller shall abort the command with a status of Invalid Field in Command.

If the range of logical block addresses to be flushed by the Flush Explicit ZRWA Range Send Action contains a logical block address that is not contained in the ZRWA, then the command shall be aborted with a status code of Invalid Zone Operation Request.

Logical blocks which were not written to the ZRWA prior to being flushed by the Flush Explicit ZRWA Range Send Action shall be treated as unwritten logical blocks when read (refer to the Deallocated or Unwritten Logical Blocks section in the NVM Command Set Specification).

If a Flush Explicit ZRWA Range Send Zone Action is requested and the specified zone is not:

    a) in the ZSEO:Explicitly Opened state, the ZSIO:Implicitly Opened state, or the ZSC:Closed state; and

    b) associated with a ZWRA,

then the controller shall abort the command with a status code of Invalid Zone Operation Request.

If the Select All bit is set to '1', the controller shall abort the command with a status code of Invalid Field in Command.

[…snip…]

Modify section 4.1.5.1 as shown below

4.1.5.1   I/O Command Set Specific Identify Namespace Data Structure for the Zoned Command Set (CNS 05h, CSI 02h)

Figure <mark>48</mark> defines the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set.

Figure 48: I/O Command Set Specific Identify Namespace Data Structure for the Zoned Namespace Command Set

Bytes	O/M [1]	Description
01:00	O	**Zone Operation Characteristics (ZOC):** This field indicates the zone operation characteristics of the zoned namespace. | Bits	Description
---	---
15:2	Reserved
1	**Zone Active Excursions:** If set to '1', then a controller may transition a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state to the ZSF:Full state due to a vendor specific excursion event. If cleared to '0', then a controller shall not transition a zone due to a vendor specific excursion event. Refer to section <mark>5.6.</mark>
0	**Variable Zone Capacity:** if set to '1', then the capacity for a zone may change without a change to the format of the zoned namespace. If cleared to '0', then the capacity for a zone does not change without a change to the format of the zoned namespace. Refer to Figure <mark>37</mark>.	
03:02	O	**Optional Zoned Command Support (OZCS):** This field defines optional features of the zoned namespace. | Bits	Description
---	---
15:~~1~~2	Reserved
1	ZRWA Supported (ZRWASUP): If set to '1' indicates that the namespace supports the ZRWA capability. If cleared to '0', then the namespace does not support the ZRWA capability. Refer to section <mark>5.TBD</mark>
0	**Read Across Zone Boundaries:** If set to '1', then any command is allowed to perform read operations that specify an LBA range containing logical blocks in more than one zone. If cleared to '0', then any command that performs a read operation that specifies an LBA range containing logical blocks in more than one zone is aborted as described in section <mark>2.1.1.2.1.2</mark>	
07:04	M	**Maximum Active Resources (MAR):** This field defines the maximum number of concurrently active zones in the zoned namespace. A value of FFFFFFFh indicates that there is no limit. This is a 0's based value
11:08	M	**Maximum Open Resources (MOR):** This field defines the maximum number of concurrently open zones in the zoned namespace. A value of FFFFFFFFh indicates that there is no limit. This is a 0's based value
…	…	….
43:40	O	**Finish Recommended Limit 3 (FRL3):** If the zone attribute Finish Zone Recommended Time Limit field is set to 11b, then this field indicates the number of seconds before the NVM subsystem may perform the vendor specific action on a zone after the Finish Zone Recommended zone attribute is set to '1' for that zone. If this field is cleared to 0h, then no Finish Recommended Limit is reported. Refer to section <mark>5.5</mark>.

Figure 48: I/O Command Set Specific Identify Namespace Data Structure for the Zoned Namespace Command Set

Bytes	O/M [1]	Description
47:44	O	**Number of ZRWA Resources (NUMZRWA):** This field indicates the total number of ZRWA resources in this namespace. Refer to section 5.TBD. This field is a 0's based value. If the ZRWA Supported bit is cleared to '0' in the OZCS field, then this field shall be ignored.
49:48	O	**ZRWA Flush Granularity (ZRWAFG):** This field specifies the granularity of ZRWA Flush operations (refer to section 5.TBD.1) in logical blocks for this namespace. Unless otherwise specified, data shall be flushed from logical blocks in a ZRWA to its associated zone in integral multiples of the value in this field. If the ZRWA Supported bit is cleared to '0' in the OZCS field, then this field shall be cleared to 0h. This value may change if the namespace is reformatted.
51:50	O	**ZRWA Size (ZRWASZ):** This field indicates the total size in logical blocks of each ZRWA for this namespace. The value in this field shall be a multiple of the non-zero value of the ZRWA Flush Granularity (ZRWAFG) field. If the ZRWA Supported bit is cleared to '0' in the OZCS field, then this field shall be cleared to 0h. Refer to section 5.TBD. This value may change if the namespace is reformatted.
52	O	**ZRWA Capability (ZRWACAP):** This field provides information about the ZRWA capability. If the ZRWA Supported bit is cleared to '0' in the OZCS field, then this field shall be cleared to 0h. <table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:1</td><td>Reserved</td></tr><tr><td>0</td><td>**EXPFLUSHSUP:** If this bit is set to '1', then the controller supports explicit ZRWA Flush operations (refer to section 5.TBD.2). If this bit is cleared to '0', then the controller does not support explicit ZRWA Flush operations.</td></tr></table>
2815:2053		Reserved
2831:2816	M	**LBA Format 0 Extension (LBAFE0):** This field indicates the LBA format Extension 0 that is supported by the controller. The Zone format field is defined in Figure 49
2847:2832	O	**LBA Format 1 Extension (LBAFE1):** This field indicates the LBA format 1 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 49.
...	...	....
4095:3840	O	Vendor Specific

NOTES:
1. O/M definition: O = Optional, M = Mandatory.

5 Extended Capabilities

[…snip…]

5.TBD Zone Random Write Area

A ZRWA may be thought of as being analogous to a type of non-volatile cache.

Support for ZRWA is optional. If assignment of a ZRWA to a zone is supported, then the ZRWASUP bit shall be set to '1' in the Zoned Namespace specific Identify Namespace data structure. If the ZRWASUP bit is cleared to '0' in the OZCS field in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set, then assignment of a ZRWA to a zone in the specified namespace is not supported.

The Zone Random Write Area Size (ZRWASZ) field (refer to Figure 8) indicates the size of the ZRWA in logical blocks. The lowest-numbered LBA of the ZRWA is the write pointer of the zone with which the ZRWA is associated. The highest-numbered LBA of the ZRWA is the write pointer of the zone with which the ZRWA is associated plus ZRWASZ minus 1. For a given zone, a ZRWA of size n ends at (write pointer + n - 1). ZRWASZ shall be a multiple of ZRWA Flush Granularity (ZRWAFG) (refer to Figure 8), as shown in Figure FigWPAZ.

The ZRWA advances when the write pointer advances.

Figure FigWPAZ: Write Pointer in a Partially Written Zone with Associated ZRWA
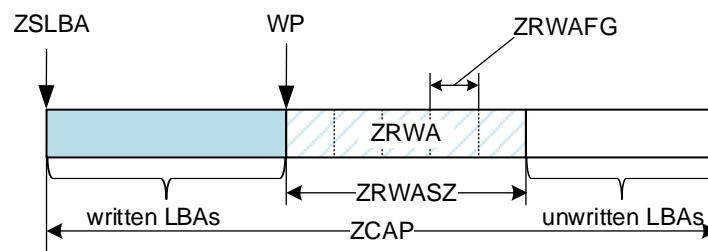
A ZRWA is allocated to a zone using the Zone Send Management command with a Zone Send Action of Open Zone or Set Zone Descriptor Extension. Allocating a ZRWA to a zone creates an association between the ZRWA and that zone. A ZRWA is associated with at most one zone at a time.

A ZRWA remains associated with a zone until that zone transitions to the ZSF:Full state, ZSE:Empty state, ZSRO:Read Only state, or ZSO:Offline state. A transition to the ZSF:Full state, ZSE:Empty state, ZSRO:Read Only state, or ZSO:Offline state terminates the association, releases the ZRWA associated with that zone, and increases the number of ZRWA resources available for allocation by 1 (refer to the Number of ZRWA Resources field in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set).

For a zone that is associated with a ZRWA:

a) logical blocks within the ZRWA are not required to be written sequentially; a host may write data to logical block addresses within the ZRWA in any order;
b) the Starting LBA for an Address-Specific Write Command to the ZRWA is required to be greater than or equal to the write pointer of the zone;
c) the host is not limited to a single outstanding Address-Specific Write Command at one time to the ZRWA; and
d) the host may overwrite logical blocks in the ZRWA.

If a zone is associated with a ZRWA, then any Address-Specific Write Command that attempts to address an LBA that is less than the write pointer of that zone shall be aborted with a status code of Invalid Zone Write.

A ZRWA Flush operation (refer to section 5.TBD.1) is used to flush logical blocks and metadata from a ZRWA to the zone associated with that ZRWA.

The controller shall perform a ZRWA Flush operation prior to transitioning that zone to the ZSF:Full or ZSRO:Read Only state.

Address-Specific Write Commands to logical block addresses in a ZRWA do not cause the associated zones' write pointer to be updated unless that Address-Specific Write Command causes an implicit ZRWA Flush operation to occur.

If a Zone Append command is received for a zone while that zone is associated with a ZRWA, then the controller shall abort the command as described in section 3.4.1.

All writes that start within the ZRWA or the IZFR (refer to section 5.TBD.3) are allowed to specify LBAs up to the end of the writeable capacity of the zone.

The coordination of host usage of ZRWAs associated with shared namespaces is outside the scope of this specification.

5.TBD.1   ZRWA Flush Operations

A ZRWA Flush operation flushes user data from a ZRWA (refer to section 1.4.TBD_1) to a zone and advances the write pointer as described in section 5.TBD.3.1. There is a one to one mapping between the source LBA in the ZRWA and the implied destination LBA in the associated zone. Logical blocks which have been flushed are not able to be written again until the zone transitions to the ZSE:Empty state.

When a ZRWA Flush operation that affects a logical block occurs, if that logical block is unwritten, then that logical block is flushed to the zone as an unwritten logical block (refer to section Deallocated or Unwritten Logical Blocks in the NVM Command Set Specification).

ZRWA Flush operations may be explicit or implicit. Both explicit and implicit ZRWA Flush operations:

   a)   flush logical blocks in multiples of ZRWA Flush Granularity (ZRWAFG) from the ZRWA to the zone; and
   b)   advance the write pointer by the number of logical blocks flushed to the zone in step a).

When an explicit ZRWA Flush operation flushes the logical block that causes the zone to reach its writeable zone capacity, the zone shall transition to the ZSF:Full state.

Figure FigFlush shows the effects of a ZRWA Flush operation on the write pointer and the mapping of the ZRWA to the logical blocks in the zone.

Figure FigFlush: Flush Operation

If the ZRWA is at the end of the zone (i.e., WP >= ZCAP – ZRWASZ; refer to Figure FigEndOfZone), then there is no IZFR (refer to section 5.TBD.3) and implicit ZRWA Flush operations do not occur. If the ZRWA is at the end of the zone, to initiate a ZRWA Flush, host software should send a Zone Management Send command specifying either:

    a.   a Zone Send Action of Finish Zone; or

    b.   a Zone Send Action of Flush Explicit ZRWA Range.

Figure FigEndOfZone: ZRWA at End of Zone

For a namespace that is able to support a ZRWA, ZCAP for all zones on that namespace shall be an integral multiple of ZRWAFG.

5.TBD.2   Explicit ZRWA Flush

An explicit ZRWA Flush operation is performed when a Zone Send Management command with the Zone Send Action field set to Flush Explicit ZRWA Range (refer to section 3.4.3.1_TBD) completes successfully. The EXPFLUSHSUP bit shall be set to '1' in the I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set if the Flush Explicit ZRWA Range Zone Send Action is supported.

5.TBD.3   Implicit ZRWA Flush

As a result of an Address-Specific Write Command, multiple implicit ZRWA Flush operations may occur. If one or more implicit ZRWA Flush operations occur as a result of a given Address-Specific Write Command, then at the completion of that Address-Specific Write Command, the write pointer

has been updated and all logical blocks up to the new write pointer have been flushed to the associated zone.

The range of logical blocks that starts at the first logical block beyond the end of the ZRWA and is the same size as the ZRWA (ZRWASZ) is called the Implicit ZRWA Flush Range (IZFR) as shown in Figure FigImpZRWAFlushRange.

FigImpZRWAFlushRange: Implicit ZRWA Flush Range

Start of Implicit ZRWA Flush Range	End of Implicit ZRWA Flush Range
Write Pointer + ZRWA Size	Write Pointer + (2 * ZRWA Size) – 1

The controller performs an implicit ZRWA Flush when the controller processes an Address-Specific Write Command that specifies one or more logical blocks in the IZFR.

The variables in Figure FigImpFluVars are used to describe an implicit ZRWA Flush operation.

Figure FigImpFluVars: Implicit ZRWA Flush Variables

Variable	Definition
New_WP	New Write Pointer
WP	Initial Write Pointer
IZFR_Start_LBA	Start of IZFR (i.e., WP + ZRWASZ)
Last_LBA	Ending LBA in Write Operation
FG	ZRWA Flush Granularity (ZRWAFG)

The range of LBAs flushed by an implicit ZRWA Flush operation begins with the write pointer. The number of logical blocks flushed is an integral multiple of the ZRWAFG and is at least one times ZRWAFG.

If an implicit ZRWA Flush operation occurs, then the value of the write pointer is determined by the following formula:

$$\text{if} \quad \left((Last\_LBA - IZFR\_Start\_LBA) \bmod FG \right) > 0$$

$$\text{then} \quad New\_WP = WP + Ceiling\left(\frac{Last\_LBA - IZFR\_Start\_LBA}{FG} \right) * FG$$

$$\text{else} \quad New\_WP = WP + \left(Ceiling\left(\frac{Last\_LBA - IZFR\_Start\_LBA}{FG} \right) + 1 \right) * FG$$

If an Address-Specific Write Command specifies a Starting LBA that is greater than or equal to *IZFR_Start_LBA* + ZRWASZ (i.e., the first LBA after the IZFR) then the controller shall abort that command with a status code of Zone Invalid Write, and shall not perform any implicit ZRWA Flush operations.

If an implicit ZRWA Flush operation is to be performed, then that implicit ZRWA Flush operation shall be performed whether or not the Address-Specific Write Command completes successfully.

[…snip…]

Modify section A.2 Writing to Zones as follows:

A.2 Writing to Zones

In a Sequential Write Required Zone, writes are required to start at the valid write pointer address if a ZRWA is not associated with a zone. Refer to section 5.TBD for more details.