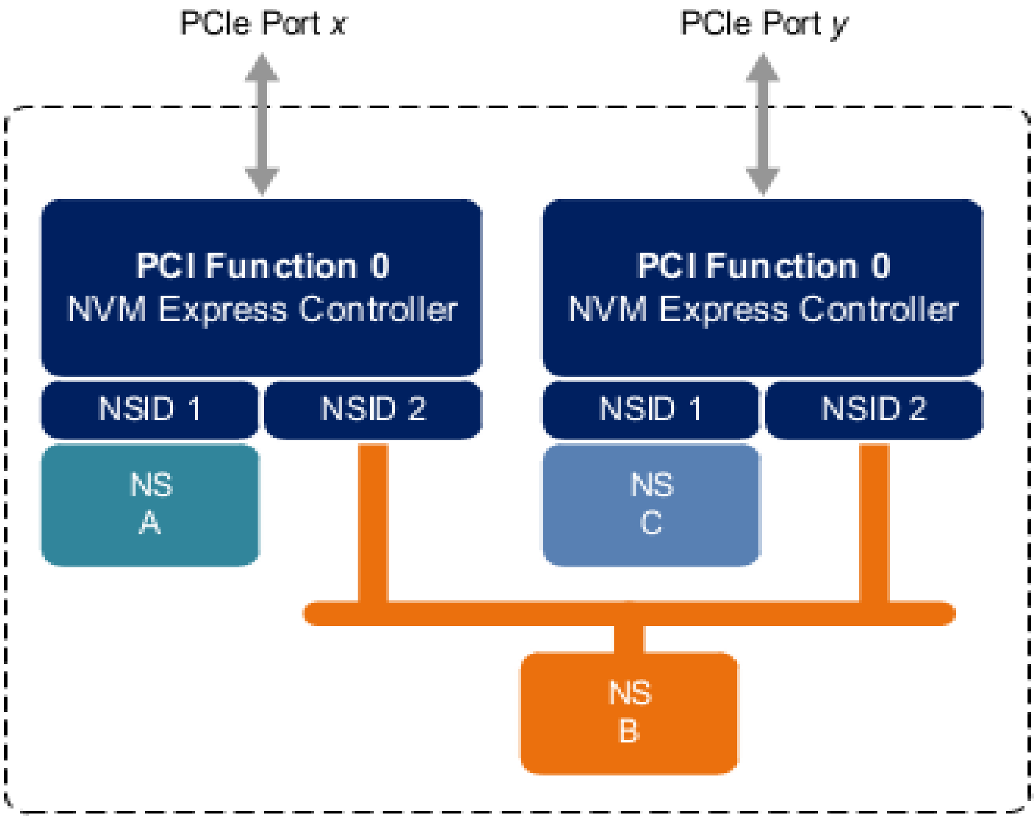
Upgradation of NVMe1.0 spec to NVMe1.1

Major features to be upgraded

- Multi path I/O and Namespace Sharing
- Scatter gather list (SGL)

Multi path I/O and Namespace Sharing

- Multi-path I/O refers to two or more completely independent physical PCI Express paths between a single host and a namespace
- while namespace sharing refers to the ability for two or more hosts to access a common shared namespace using different NVM Express controllers.
- Both multi-path I/O and namespace sharing require that the NVM subsystem contain two or more controllers.
- Concurrent access to a shared namespace by two or more hosts requires some form of coordination between hosts.
- The procedure used to coordinate these hosts is outside the scope of this specification.



- The above figure illustrates an NVM Subsystem with two PCI Express ports each with an associated controller.Both controllers map to PCI Function 0 of the corresponding port. Each PCI Express port in this example is completely independent and has its own PCI Express Fundamental Reset and reference clock input.

- A reset of a port only affects the controller associated with that port and has no impact on the other controller, shared namespace, or operations performed by the other controller on the shared namespace.
- There is a unique Identify Controller data structure for each controller and a unique Identify Namespace data structure for each namespace.
- Controllers with access to a shared namespace return the Identify Namespace data structure associated with that shared namespace.
- Each controller supports a single private namespace and access to shared namespace B. The namespace ID shall be the same in all controllers that have access to a particular shared namespace. In this example both controllers use namespace ID 2 to access shared namespace B.

Scatter Gather List (SGL)

- A Scatter Gather List (SGL) is a data structure in memory address space used to describe a data buffer.
- A data buffer is either a source buffer or a destination buffer. There is no alignment requirement for the data buffer. An SGL contains one or more SGL segments.
- An SGL segment is a Qword aligned data structure in a contiguous region of physical memory describing all, part of, or none of a data buffer and the next SGL segment, if any.
- An SGL segment consists of an array of one or more SGL descriptors.
- Only the last descriptor in an SGL segment may be an SGL Segment descriptor or an SGL Last Segment descriptor.

Figure 17: SGL Segment

Bytes	Description
15:00	**SGL Descriptor 0**
31:16	**SGL Descriptor 1**
…	…
((n*16)+15): (n*16)	**SGL Descriptor n**

Figure 18: SGL Descriptor Format

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>14:00</td><td><b>Descriptor Type Specific</b></td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>03:00</td><td>Descriptor Type Specific</td></tr>
<tr><td>07:04</td><td>SGL Descriptor Type</td></tr>
</table>
</td></tr>
</table>

Figure 19: SGL Descriptor Type

Code	Descriptor
0h	SGL Data Block descriptor
1h	SGL Bit Bucket descriptor
2h	SGL Segment descriptor
3h	SGL Last Segment descriptor
4h – Eh	Reserved
Fh	Vendor specific

Figure 20: SGL Data Block descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>7:0</td><td><b>Address:</b> The Address field specifies the starting 64-bit memory byte address of the data block.</td></tr>
<tr><td>11:8</td><td><b>Length:</b> The Length field specifies the length in bytes of the data block. A Length field set to 00000000h specifies that no data is transferred. An SGL Data Block descriptor specifying that no data is transferred is a valid SGL Data Block descriptor.
If the value in the Address field plus the value in the Length field is greater than 1_00000000_00000000h then the SGL Data Block descriptor shall be processed as having an error.</td></tr>
<tr><td>14:12</td><td>Reserved</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>03:00</td><td>Zero: The Zero field shall have the value of 0h. An SGL Data Block descriptor containing a Zero field set to a value other than 0h shall be processed as having an error.</td></tr>
<tr><td>07:04</td><td>SGL Descriptor Type: 0h as specified in Figure 19.</td></tr>
</table>
</td></tr>
</table>

Figure 21: SGL Bit Bucket descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>7:0</td><td>Reserved</td></tr>
<tr><td>11:8</td><td><b>Length:</b> If the SGL describes a destination data buffer, then the Length field specifies the number of bytes of the source data not to be transferred (i.e., the number of bytes to be discarded). A Length field set to 00000000h specifies that no source data shall be discarded. An SGL Bit Bucket descriptor specifying that no source data be discarded is a valid SGL Bit Bucket descriptor.
If the SGL describes a source data buffer (i.e., a write from host memory to the controller) then the Length field shall be ignored and no data shall be discarded from the source data or destination data. An SGL Bit Bucket descriptor specifying that no data be discarded shall not be processed as having an error.</td></tr>
<tr><td>14:12</td><td>Reserved</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>03:00</td><td>Zero: The Zero field shall have the value of 0h. An SGL Bit Bucket descriptor containing a Zero field set to a value other than 0h shall be processed as having an error.</td></tr>
<tr><td>07:04</td><td>SGL Descriptor Type: 1h as specified in Figure 19.</td></tr>
</table>
</td></tr>
</table>

Figure 22: SGL Segment descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>7:0</td><td><b>Address:</b> The Address field specifies the starting 64-bit memory byte address of the next SGL segment, which is a SGL segment.</td></tr>
<tr><td>11:8</td><td><b>Length:</b> The Length field specifies the length in bytes of the next SGL segment. The Length field shall be a non-zero value and a multiple of 16.
If the value in the Address field plus the value in the Length field is greater than 1_00000000_00000000h, then the SGL Segment descriptor shall be processed as having an error.</td></tr>
<tr><td>14:12</td><td>Reserved</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>03:00</td><td>Zero: The Zero field shall have the value of 0h. An SGL Segment descriptor containing a Zero field set to a value other than 0h shall be processed as having an error.</td></tr>
<tr><td>07:04</td><td>SGL Descriptor Type: 2h as specified in Figure 19.</td></tr>
</table>
</td></tr>
</table>

Figure 23: SGL Last Segment descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>7:0</td><td><b>Address:</b> The Address field specifies the starting 64-bit memory byte address of the next and last SGL segment, which is a SGL segment.</td></tr>
<tr><td>11:8</td><td><b>Length:</b> The Length field specifies the length in bytes of the next and last SGL segment. The Length field shall be a non-zero value and a multiple of 16.
If the value in the Address field plus the value in the Length field is greater than 1_00000000_00000000h, then the SGL Last Segment descriptor shall be processed as having an error.</td></tr>
<tr><td>14:12</td><td>Reserved</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>03:00</td><td>Zero: The Zero field shall have the value of 0h. An SGL Last Segment descriptor containing a Zero field set to a value other than 0h shall be processed as having an error.</td></tr>
<tr><td>07:04</td><td>SGL Descriptor Type: 3h as specified in Figure 19.</td></tr>
</table>
</td></tr>
</table>

Admin Command Set

- Get Features command

  Select field :

  A Select field set to 000b (i.e., current) returns the current operating attribute value for the FeatureIdentifier specified.

  A Select field set to 001b (i.e., default) returns the default attribute value for the Feature Identifier specified

- A Select field set to 010b (i.e., saved) returns the last saved attribute value for the Feature Identifier specified (i.e., the last Set Features command completed without error, with the Save bit set to ‘1’ for the Feature Identifier specified.)
- A Select field set to 011b (i.e., supported capabilities) returns the capabilities supported for this Feature Identifier.
- The capabilities supported are returned in Dword 0 of the completion entry of the Get Features Command.
- If Dword 0 bit 0 of the completion entry of the Get Features command is set to ‘1’, then the Feature Identifier is saveable. If Dword 0 bit 0 of the completion entry of the Get Features command is cleared to ‘0’, then the Feature Identifier is not saveable.
- If Dword 0 bit 1 of the completion entry of the Get Features command is set to ‘1’, then the Feature Identifier is namespace specific and settings are applied to individual namespaces.
- If Dword 0 bit 1 of the completion entry of the Get Features command is cleared to ‘0’, then the Feature Identifier is not namespace specific and its settings apply to the entire controller.
- If Dword 0 bit 2 of the completion entry of the Get Features command is set to ‘1’, then the Feature Identifier is changeable. If Dword 0 bit 2 of the completion entry of the Get Features ommand is cleared to ‘0’, then the Feature Identifier is not changeable.

NVM Command Set

Reservation Acquire command

- The Reservation Acquire command is used to acquire a reservation on a namespace, preempt a reservation held on a namespace, and abort a reservation held on a namespace.
- The command uses Command Dword 10 and a Reservation Acquire data structure in host memory. If the command uses PRPs for the data transfer, then PRP Entry 1 and PRP Entry 2 fields are used.
- If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used.
- All other command specific fields are reserved.

Figure 145: Reservation Acquire – PRP Entries or SGL Entry 1

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>127:00</td><td>If CDW0[15] is cleared to ‘0’, then the definition of this field is:
<table>
<tr><td>127:64</td><td><b>PRP Entry 2 (PRP2):</b> This field contains the second PRP entry that specifies the location where data is transferred from (if there is a physical discontinuity). This field shall not be a pointer to a PRP List.</td></tr>
<tr><td>63:00</td><td><b>PRP Entry 1 (PRP1):</b> Indicates a data buffer where data is transferred from.</td></tr>
</table>
If CDW0[15] is set to ‘1’, then the definition of this field is:
<table>
<tr><td>127:00</td><td><b>SGL Entry 1 (SGL1):</b> This field contains the first SGL segment for the command, indicating the location of a data buffer where data is transferred from.</td></tr>
</table>
.</td></tr>
</table>

Figure 148: Reservation Type Encoding

Value	Description
0h	Reserved
1h	Write Exclusive Reservation
2h	Exclusive Access Reservation
3h	Write Exclusive - Registrants Only Reservation
4h	Exclusive Access - Registrants Only Reservation
5h	Write Exclusive - All Registrants Reservation
6h	Exclusive Access - All Registrants Reservation
07h-FFh	Reserved

Figure 146: Reservation Acquire – Command Dword 10

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td>Reserved</td></tr>
<tr><td>15:08</td><td><b>Reservation Type (RTYPE):</b> This field specifies the type of reservation to be created. The field is defined in Figure 148.</td></tr>
<tr><td>07:04</td><td>Reserved</td></tr>
<tr><td>03</td><td><b>Ignore Existing Key (IEKEY)</b>: If this bit is set to a ‘1’, then the Current Reservation Key (CRKEY) check is disabled and the command shall succeed regardless of the CRKEY field value.</td></tr>
<tr><td>02:00</td><td><b>Reservation Acquire Action (RACQA):</b> This field specifies the action that is performed by the command.
<table>
<tr><th>RACQA Value</th><th>Description</th></tr>
<tr><td>000b</td><td>Acquire</td></tr>
<tr><td>001b</td><td>Preempt</td></tr>
<tr><td>010b</td><td>Preempt and Abort</td></tr>
<tr><td>011b - 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

Figure 147: Reservation Acquire Data Structure

Bytes	O/M	Description
7:0	M	**Current Reservation Key (CRKEY):** The field specifies the current reservation key associated with the host. If the IEKEY bit is set to ‘1’ in the command, then the CRKEY check succeeds regardless of the value in this field.
15:8	M	**Preempt Reservation Key (PRKEY):** If the Reservation Acquire Action is set to 001b (i.e., Preempt) or 010b (i.e., Preempt and Abort), then this field specifies the reservation key to be unregistered from the namespace. For all other Reservation Acquire Action values, this field is reserved.

Reservation Register command

- The Reservation Register command is used to register, unregister, or replace a reservation key.
- The command uses Command Dword 10 and a Reservation Register data structure in host memory.
- If the command uses PRPs for the data transfer, then PRP Entry 1 and PRP Entry 2 fields are used.
- If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

Figure 150: Reservation Register – Command Dword 10

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:30</td><td><b>Change Persist Through Power Loss State (CPTPL):</b> This field allows the Persist Through Power Loss state associated with the namespace to be modified as a side effect of processing this command.
<table>
<tr><th>CPTPL Value</th><th>Description</th></tr>
<tr><td>00b</td><td>No change to PTPL state</td></tr>
<tr><td>01b</td><td>Reserved</td></tr>
<tr><td>10b</td><td>Set PTPL state to ‘0’. Reservations are released and registrants are cleared on a power on.</td></tr>
<tr><td>11b</td><td>Set PTPL state to ‘1’. Reservations and registrants persist across a power loss.</td></tr>
</table></td></tr>
<tr><td>29:04</td><td>Reserved</td></tr>
<tr><td>03</td><td><b>Ignore Existing Key (IEKEY)</b>: If this bit is set to a ‘1’, then Reservation Register Action (RREGA) field values that use the Current Reservation Key (CRKEY) shall succeed regardless of the value of the Current Reservation Key field in the command (i.e., the current reservation key is not checked).</td></tr>
<tr><td>02:00</td><td><b>Reservation Register Action (RREGA):</b> This field specifies the registration action that is performed by the command.
<table>
<tr><th>RREGA Value</th><th>Description</th></tr>
<tr><td>000b</td><td>Register Reservation Key</td></tr>
<tr><td>001b</td><td>Unregister Reservation Key</td></tr>
<tr><td>010b</td><td>Replace Reservation Key</td></tr>
<tr><td>011b - 111b</td><td>Reserved</td></tr>
</table></td></tr>
</table>

Figure 151: Reservation Register Data Structure

Bytes	O/M	Description
7:0	M	**Current Reservation Key (CRKEY):** If the Reservation Register Action is 001b (i.e., Unregister Reservation Key) or 010b (i.e., Replace Reservation Key), then this field contains the current reservation key associated with the host. For all other Reservation Register Action values, this field is reserved. The controller ignores the value of this field when the Ignore Existing Key (IEKEY) bit is set to ‘1’.
15:8	M	**New Reservation Key (NRKEY):** If the Reservation Register Action is 000b (i.e., Register Reservation Key) or 010b (i.e., Replace Reservation Key), then this field contains the new reservation key associated with the host. For all other Reservation Register Action values, this field is reserved.

Reservation Release command

- The Reservation Release command is used to release or clear a reservation held on a namespace.
- The command uses Command Dword 10 and a Reservation Release data structure in host memory.
- If the command uses PRPs for the data transfer, then PRP Entry 1 and PRP Entry 2 fields are used.
- If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

Figure 152: Reservation Release – PRP Entries or SGL Entry 1

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>127:00</td><td>If CDW0[15] is cleared to ‘0’, then the definition of this field is:
<table>
<tr><td>127:64</td><td><b>PRP Entry 2 (PRP2):</b> This field contains the second PRP entry that specifies the location where data is transferred from (if there is a physical discontinuity). This field shall not be a pointer to a PRP List.</td></tr>
<tr><td>63:00</td><td><b>PRP Entry 1 (PRP1):</b> Indicates a data buffer where data is transferred from.</td></tr>
</table>
If CDW0[15] is set to ‘1’, then the definition of this field is:
<table>
<tr><td>127:00</td><td><b>SGL Entry 1 (SGL1):</b> This field contains the first SGL segment for the command, indicating the location of a data buffer where data is transferred from.</td></tr>
</table>
.</td></tr>
</table>

Figure 153: Reservation Release – Command Dword 10

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td>Reserved</td></tr>
<tr><td>15:08</td><td><b>Reservation Type (RTYPE):</b> If the Reservation Release Action is 00b (i.e., Release), then this field specifies the type of reservation that is being released. The reservation type in this field shall match the current reservation type. This field is defined in Figure 148.</td></tr>
<tr><td>07:04</td><td>Reserved</td></tr>
<tr><td>03</td><td><b>Ignore Existing Key (IEKEY)</b>: If this bit is set to a ‘1’, then the Current Reservation Key (CRKEY) check is disabled and the command succeeds regardless of the CRKEY field value.</td></tr>
<tr><td>02:00</td><td><b>Reservation Release Action (RRELA):</b> This field specifies the registration action that is performed by the command.
<table>
<tr><th>RRELA Value</th><th>Description</th></tr>
<tr><td>000b</td><td>Release</td></tr>
<tr><td>001b</td><td>Clear</td></tr>
<tr><td>001b - 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

Figure 154: Reservation Release Data Structure

Bytes	O/M	Description
7:0	M	**Current Reservation Key (CRKEY):** The field specifies the current reservation key associated with the host. If the IEKEY bit is set to ‘1’ in the command, then the CRKEY check succeeds regardless of the value in this field.

Reservation Report command

- The Reservation Report command returns a Reservation Status data structure to host memory that describes the registration and reservation status of a namespace.
- The size of the Reservation Status data structure is a function of the number of controllers in the NVM Subsystem that are associated with hosts that are registrants of the namespace (i.e., there is a Registered Controller data structure for each such controller).
- The command uses Command Dword 10. If the command uses PRPs for the data transfer, then PRP Entry 1 and PRP Entry 2 fields are used.
- If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used.
- All other command specific fields are reserved.

Figure 156: Reservation Report – Command Dword 10

Bit	Description
31:00	**Number of Dwords (NUMD):** This field specifies the number of Dwords of the Reservation Status data structure to transfer. This is a 0's based value. If this field corresponds to a length that is less than the size of the Reservation Status data structure, then only that specified portion of the data structure is transferred. If this field corresponds to a length that is greater than the size of the Reservation Status data structure, then the entire contents of the data structure are transferred and no additional data is transferred.

Write Zeroes command

- The Write Zeroes command is used to set a range of logical blocks to zero. After successful completion of this command, the value returned by subsequent reads of logical blocks in this range shall be zeroes until a write occurs to this LBA range.
- The fields used are Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 14, and Command Dword 15 fields.

Figure 169: Write Zeroes – Command Dword 10 and Command Dword 11

Bit	Description
63:00	**Starting LBA (SLBA):** This field indicates the 64-bit address of the first logical block to be written as part of the operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:32.

Figure 170: Write Zeroes – Command Dword 12

Bit	Description
31	**Limited Retry (LR):** If set to '1', the controller should apply limited retry efforts. If cleared to '0', the controller should apply all available error recovery means to write the data to the NVM.
30	**Force Unit Access (FUA):** This field indicates that the data shall be written to non-volatile media before indicating command completion. There is no implied ordering with other commands.
29:26	**Protection Information Field (PRINFO):** Specifies the protection information action and check field, as defined in Figure 123.
25:16	Reserved
15:00	**Number of Logical Blocks (NLB):** This field indicates the number of logical blocks to be written. This is a 0's based value.

Figure 171: Write Zeroes – Command Dword 14

Bit	Description
31:00	**Initial Logical Block Reference Tag (ILBRT):** This field indicates the Initial Logical Block Reference Tag value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

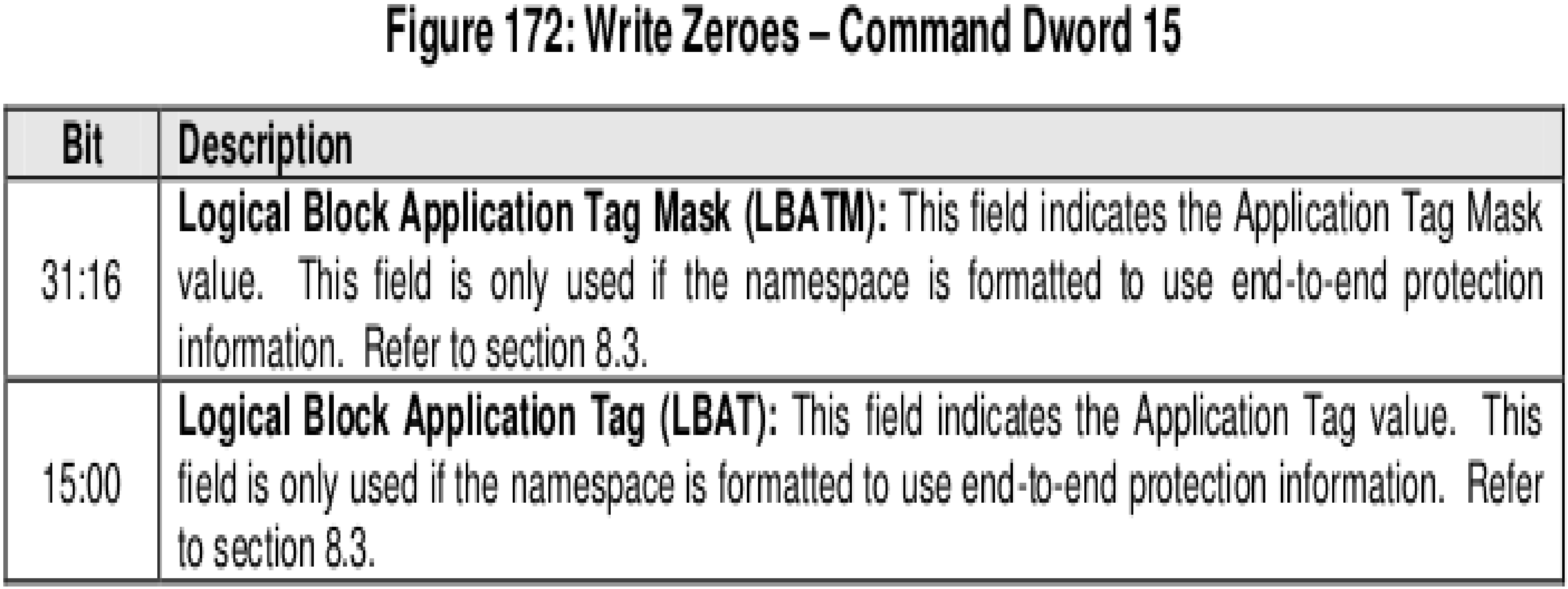


Figure 172: Write Zeroes – Command Dword 15

Bit	Description
31:16	**Logical Block Application Tag Mask (LBATM):** This field indicates the Application Tag Mask value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.
15:00	**Logical Block Application Tag (LBAT):** This field indicates the Application Tag value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.