Exhibit 5

U.S. Patent No. 11,347,657

Sources:

- Samsung, Samsung Solid State Drive White Paper, available at: https://download.semiconductor.samsung.com/resources/white-paper/Samsung_SSD_White_Paper.pdf (herein "Samsung SSD White Paper").
- Samsung, Samsung PM1733 NVMe SSD Product Brief, available at: https://download.semiconductor.samsung.com/resources/brochure/PM1733%20NVMe%20SSD.pdf (herein "Samsung PM1733 Product Brief").
- Samsung, Over-Provisioning Benefits for Samsung Data Center SSDs, available at: https://download.semiconductor.samsung.com/resources/white-paper/S190311-SAMSUNG-Memory-Over-Provisioning-White-paper_10129514063996.pdf (herein "Samsung Over-Provisioning White Paper").
- Jalil Boukhobza et al., Flash Memory Integration (ISTE Press Ltd 2017) (herein "Boukhobza").
- Hanyeoreum Bae et al., What You Can't Forget: Exploiting Parallelism for Zoned Namespaces, available at: https://www.hotstorage.org/2022/slides/hotstorage22-paper24-presentation_slides.pdf (herein "Bae").
- Zoned Storage, SSDs with NVMe Zoned Namespace (ZNS) Support, available at: https://zonedstorage.io/docs/introduction/zns (herein "Zoned Storage Introduction").
- NVM Express, Zoned Namespace Command Set Specification, Revision 1.1d, (herein "ZNS Specification").
- NVM Express, NVM Command Set Specification, Revision 1.1, (herein "NVM Command Set Specification").
- NVM Express, NVM Express Base Specification, Revision 2.0d, (herein "NVMe Base Specification").
- Samsung, Over-provisioning Maximize the Lifetime and Performance of Your SSD with Small Effect to Earn More, available at: https://download.semiconductor.samsung.com/resources/others/Samsung_SSD_845DC_04_Over-provisioning.pdf (herein "Samsung Application Note").
- Microsoft, Denali the Next Generation High Density Storage Interface, March 2018 OCP Summit (herein "Microsoft Denali Presentation").

- Kyuhwa Han et al., ZNS+: Advanced Zoned Namespace Interface for Supporting In-Storage Zone Compaction, 15th USENIX Symposium on Operating Systems Design and Implementation (herein "Han").
- Jeongdong Choe, EETimes, Samsung's 3D V-NAND 32L vs 48L–Just Vertical Expansion?, available at: https://www.eetimes.com/samsungs-3d-v-nand-32l-vs-48l-just-vertical-expansion/ (herein "EETimes V-NAND Analysis").
- Anton Shilov, AnandTech, Samsung Starts Mass Production of 9th Generation V-NAND: 1Tb 3D TLC NAND, available at: https://www.anandtech.com/show/21365/samsung-starts-mass-production-of-9th-generation-vnand-1tb-3d-tlc-nand (herein "Anand V-NAND Analysis").
- Samsung, Next-Generation QLC V-NAND Increases Data Center Profitability, available at: https://semiconductor.samsung.com/news-events/tech-blog/next-generation-qlc-v-nand-increases-data-center-profitability/ (herein "Samsung Tech Blog").
- Matias Bjørling et al., USENIX, Getting Zoned Storage out of the Friend Zone, available at https://www.usenix.org/publications/loginonline/getting-zoned-storage-out-friend-zone (herein "Bjørling").
- Samsung's U.S. Patent Application Publication No. 2021/0349781 (herein "Samsung's '781 Patent Application").
- Samsung's U.S. Patent Application Publication No. 2024/0319890 (herein "Samsung's '890 Patent Application").
- Samsung's Chinese Patent Application Publication No. CN117,055,816A (herein "Samsung's '816 Patent Application").
- Samsung's U.S. Patent No. 11,334,250 (herein "Samsung's '250 Patent").

<table>
<tr><th>No.</th><th>Claim Language</th><th>Samsung's Accused Instrumentalities</th></tr>
<tr><td>1pre</td><td>A storage device comprising:</td><td>To the extent the preamble is considered a limitation, Samsung's Accused Instrumentalities contain a storage device.</td></tr>
<tr><td>1a</td><td>flash memory comprising erase units; and</td><td>Samsung's Accused Instrumentalities comprise a flash memory comprising erase units.

Samsung's Accused Instrumentalities contain NAND flash memory.

For example, Samsung's PM1733 contains NAND flash memory. See:

Samsung PM1733 specifications
<table><tr><td>Form factor</td><td>U.2 / 2.5"</td></tr><tr><td>Capacity</td><td>1.92 TB, 3.84 TB, 7.68TB, and 15.36TB</td></tr><tr><td>Host interface</td><td>PCIe Gen 3/4 x4</td></tr><tr><td>Spec Compliance</td><td>NVMe spec rev. 1.3
PCI Express base specification rev. 4.0</td></tr><tr><td>NAND flash memory</td><td>Samsung V-NAND®</td></tr></table>
Source: Samsung PM1733 Product Brief (annotated).

The flash memory in Samsung's Accused Instrumentalities comprises blocks (erase units). See:</td></tr>
</table>

Writing and Erasing NAND

Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept

Source: Samsung SSD White Paper, at 17.

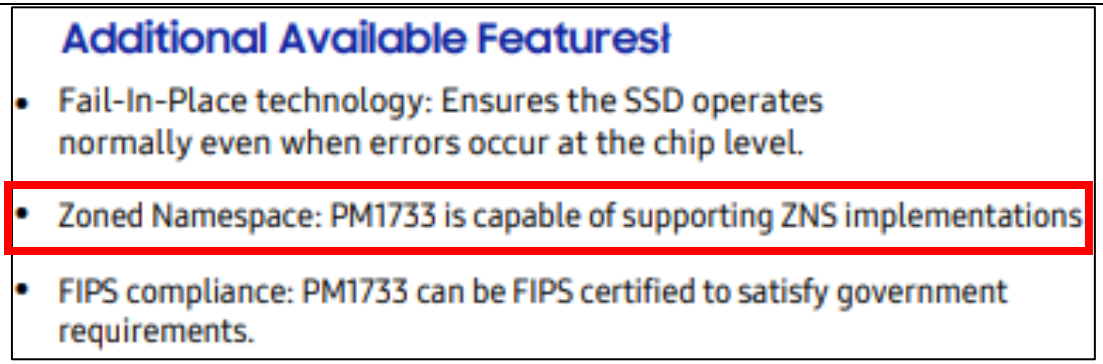
Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP).

Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block.

Source: Boukhobza, at 22 (highlighted).

		2.2.1. *Erase-before-write constraint* Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: *Id.* at 24.
1b	circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is accompanied by incoming address information;	Samsung's Accused Instrumentalities contain circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is accompanied by incoming address information. Samsung's Accused Instrumentalities are able to receive and execute a Write command from a host. A Write command is a data write request. A Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. See: **3.3.6 Write command** The Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. The host may also specify protection information to include as part of the operation. Source: NVM Command Set Specification, at 53. The Write command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which data is to be written. The NSID and SLBA together are incoming address information. See:



Source: Samsung PM1733 Product Brief (annotated).

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVM Command Set Specification, at 107.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2

Source: ZNS Specification, at 18 (annotated).

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

Samsung's Accused Instrumentalities are able to receive and execute a Zone Management Send command specifying a Reset Zone action (erase request) from the host. See:

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

Source: ZNS Specification, at 29.

3.4.3.1.4 Reset Zone

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

a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. The Zone Management Send command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which the Zone Management Send command applies. The NSID and SLBA together are incoming address information. See:

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

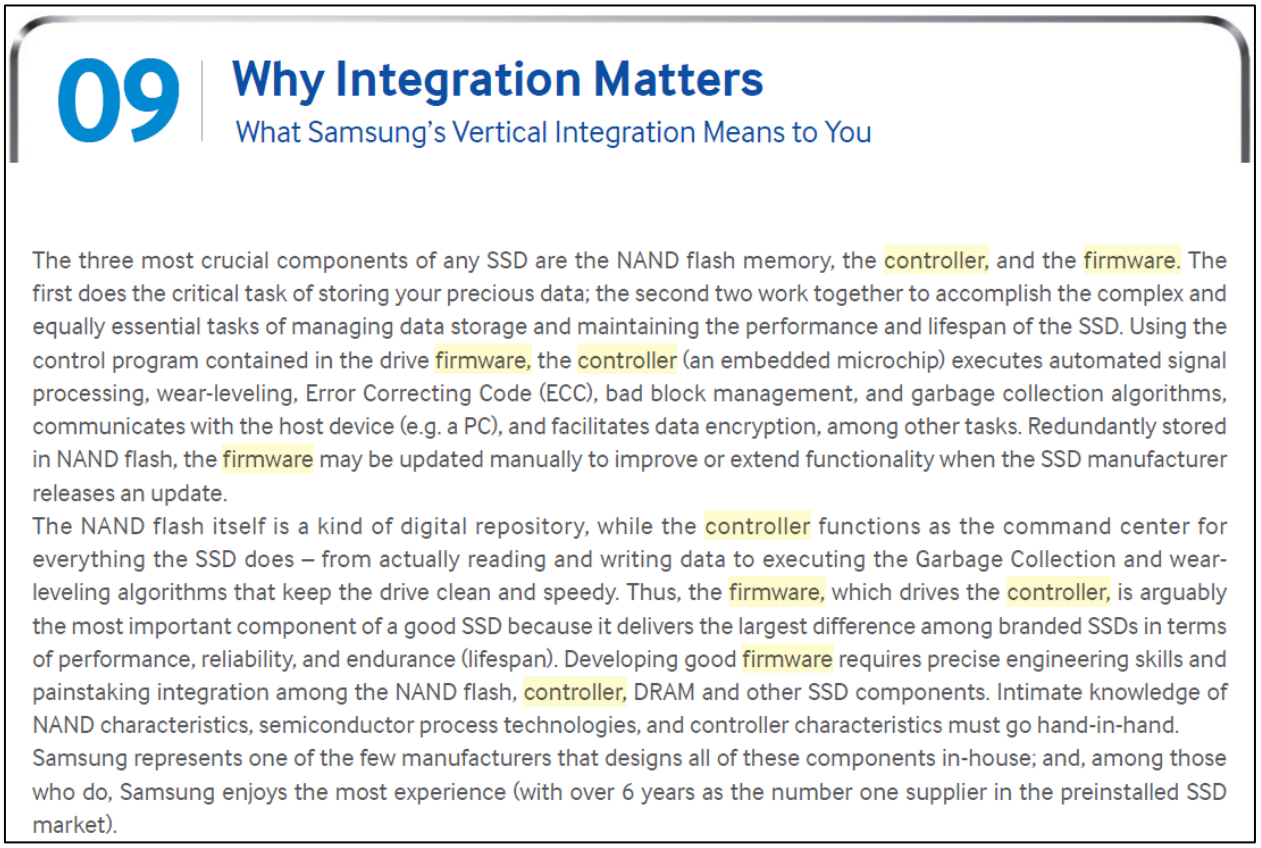
Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVM Command Set Specification, at 107. **3.4.3 Zone Management Send command** The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11, and Command Dword 13 fields. All other command specific fields are reserved. **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. Source: ZNS Specification, at 29.
1c	wherein said circuitry is to,	Samsung's Accused Instrumentalities comprise circuitry to, for each of the incoming requests, perform what is described in elements [1d] through [1g]. See below.

	for each of the incoming requests:	Samsung's Accused Instrumentalities contain a processor or memory controller that executes firmware (circuitry). See: 

		Source: Samsung PM1733 Product Brief.  09 Why Integration Matters What Samsung's Vertical Integration Means to You The three most crucial components of any SSD are the NAND flash memory, the controller, and the firmware. The first does the critical task of storing your precious data; the second two work together to accomplish the complex and equally essential tasks of managing data storage and maintaining the performance and lifespan of the SSD. Using the control program contained in the drive firmware, the controller (an embedded microchip) executes automated signal processing, wear-leveling, Error Correcting Code (ECC), bad block management, and garbage collection algorithms, communicates with the host device (e.g. a PC), and facilitates data encryption, among other tasks. Redundantly stored in NAND flash, the firmware may be updated manually to improve or extend functionality when the SSD manufacturer releases an update. The NAND flash itself is a kind of digital repository, while the controller functions as the command center for everything the SSD does – from actually reading and writing data to executing the Garbage Collection and wear-leveling algorithms that keep the drive clean and speedy. Thus, the firmware, which drives the controller, is arguably the most important component of a good SSD because it delivers the largest difference among branded SSDs in terms of performance, reliability, and endurance (lifespan). Developing good firmware requires precise engineering skills and painstaking integration among the NAND flash, controller, DRAM and other SSD components. Intimate knowledge of NAND characteristics, semiconductor process technologies, and controller characteristics must go hand-in-hand. Samsung represents one of the few manufacturers that designs all of these components in-house; and, among those who do, Samsung enjoys the most experience (with over 6 years as the number one supplier in the preinstalled SSD market). Source: Source: Samsung SSD White Paper, at 30.
1d	derive, from the accompanying incoming address information, a first address portion and a	Samsung's Accused Instrumentalities contain said circuitry to derive, from the accompanying incoming address information, a first address portion and a second address portion. Samsung's Accused Instrumentalities are able to derive, from the accompanying incoming address information, a Namespace Identifier or NSID (a first address portion) and a starting LBA (a second address portion). See:

second address portion

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVMe Base Specification, at 107.

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

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.

Source: ZNS Specification, at 29.

1e	identify an addressed block device from the first address portion,	Samsung's Accused Instrumentalities contain said circuitry to identify an addressed block device from the first address portion. Samsung's Accused Instrumentalities are able to identify an addressed namespace (block device) from the Namespace Identifier or NSID (the first address portion). See:

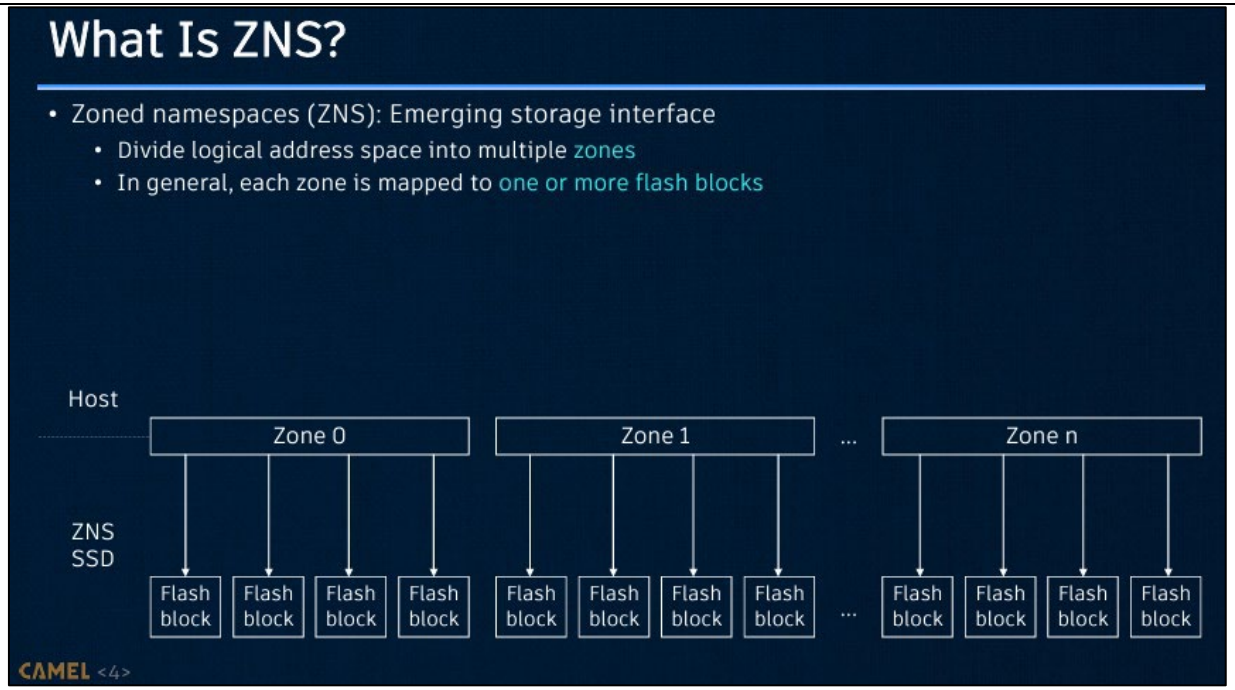
Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
If	use a first division operation on the second address portion, to identify an addressed segment within the addressed block device, wherein the addressed segment corresponds to a subset of the	Samsung's Accused Instrumentalities contain said circuitry to use a first division operation on the second address portion, to identify an addressed segment within the addressed block device, wherein the addressed segment corresponds to a subset of the erase units of the flash memory. Samsung's Accused Instrumentalities contain zones (segments) that correspond to subsets of the erase units of the flash memory. See:

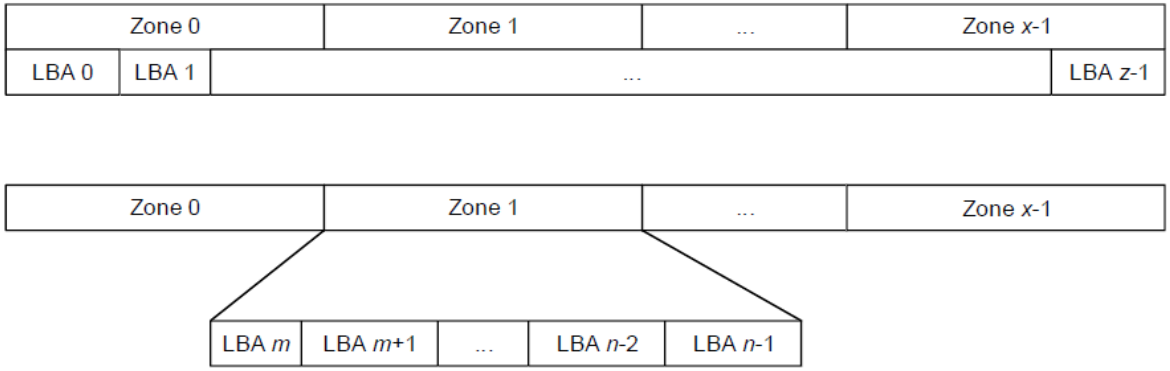
	erase units of the flash memory, and	 Source: Bae, at 4. This new attribute was introduced to allow for the zone size to remain a power of two number of logical blocks (facilitating logical block to zone number conversions) while allowing optimized mapping of a zone storage capacity to the underlying media characteristics. For instance, in the case a flash based device, a zone capacity can be aligned to the size of flash erase blocks without requiring that the device implements a power-of-two sized erased block. Source: Zoned Storage Introduction.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

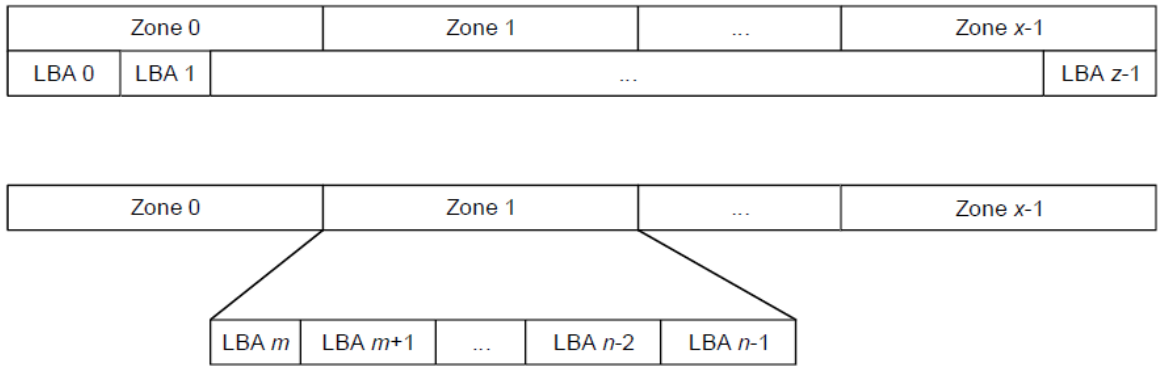
Samsung's Accused Instrumentalities are able to use a first division operation on the starting LBA (second address portion) to identify an addressed zone (segment) within the addressed namespace (block device). All zones are equally sized with contiguous non-overlapping ranges of LBAs. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.*

3.3.7 Write command

The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

3.3.7.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19.

Figure 19: Write – Command Specific Status Values

Value	Description
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.
BCh	**Zone Invalid Write:** The write to a zone was not at the write pointer.
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.

Source: *Id.* at 20.

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

If the command completes successfully, depending on the Zone Send Action field and the current states of the zones specified by the command, then that command may affect zones in various ways, including the following:

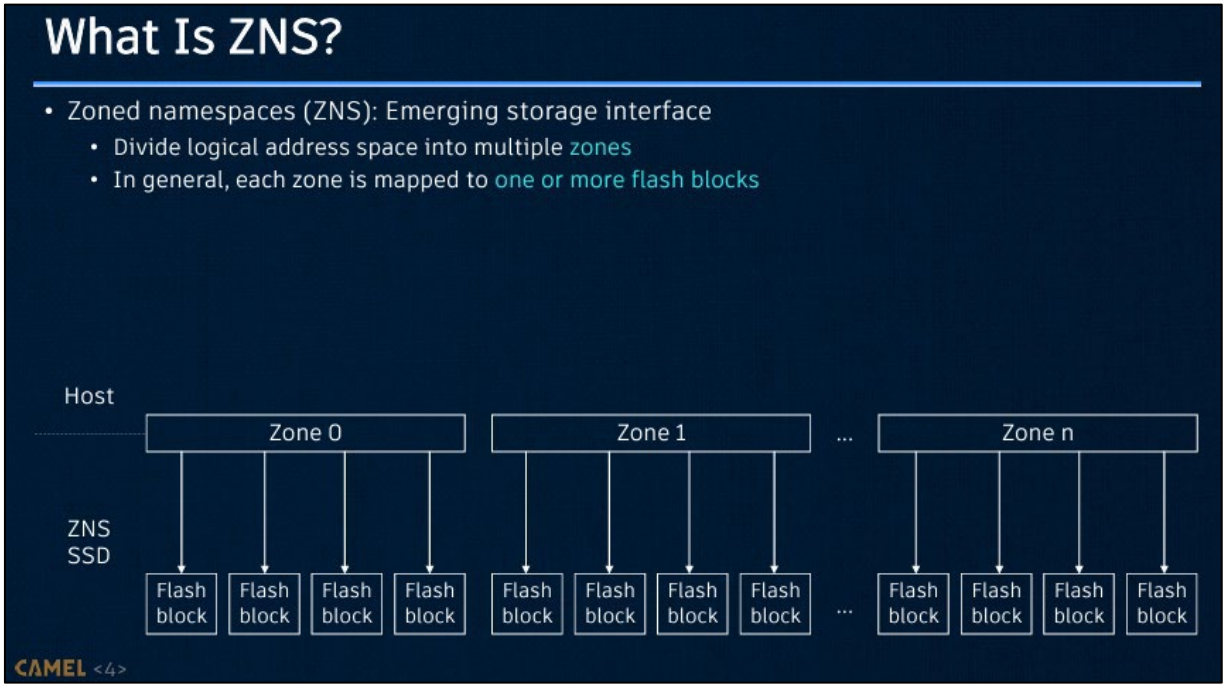
a) the zone state may change;
b) the Zone Descriptor Extension data may change; and
c) the Zone Descriptor Extension Valid zone attribute bit may change.

Source: *Id.* at 29.

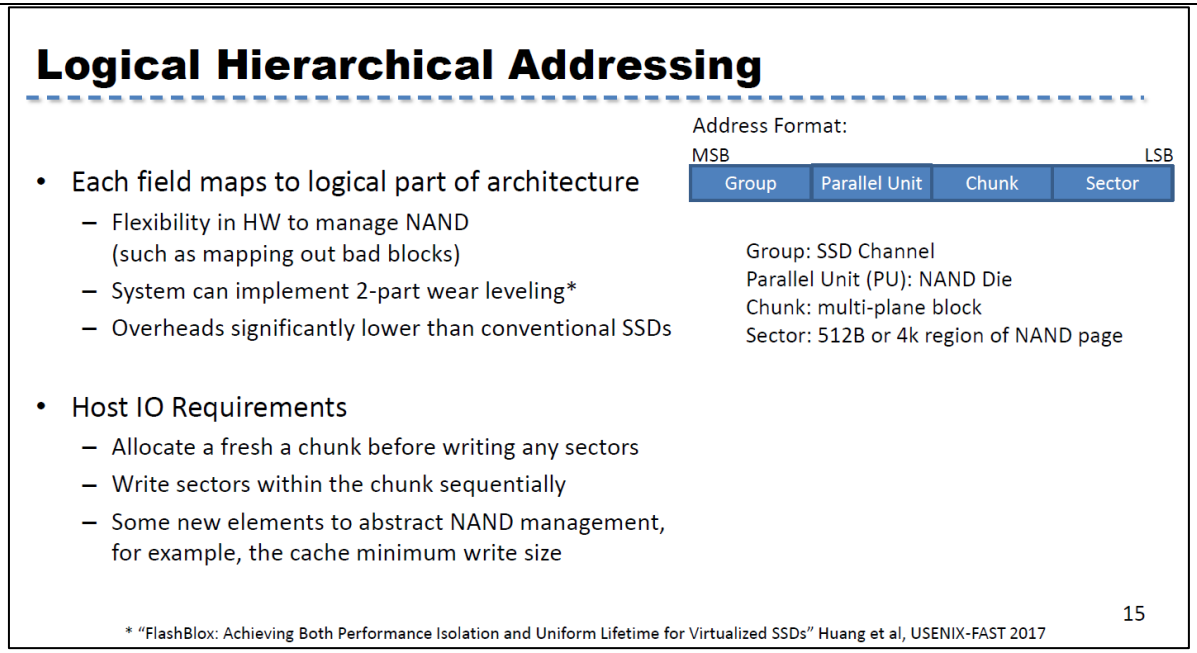
Figure 40: Zone Management Send – Command Specific Status Values

Value	Description
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.

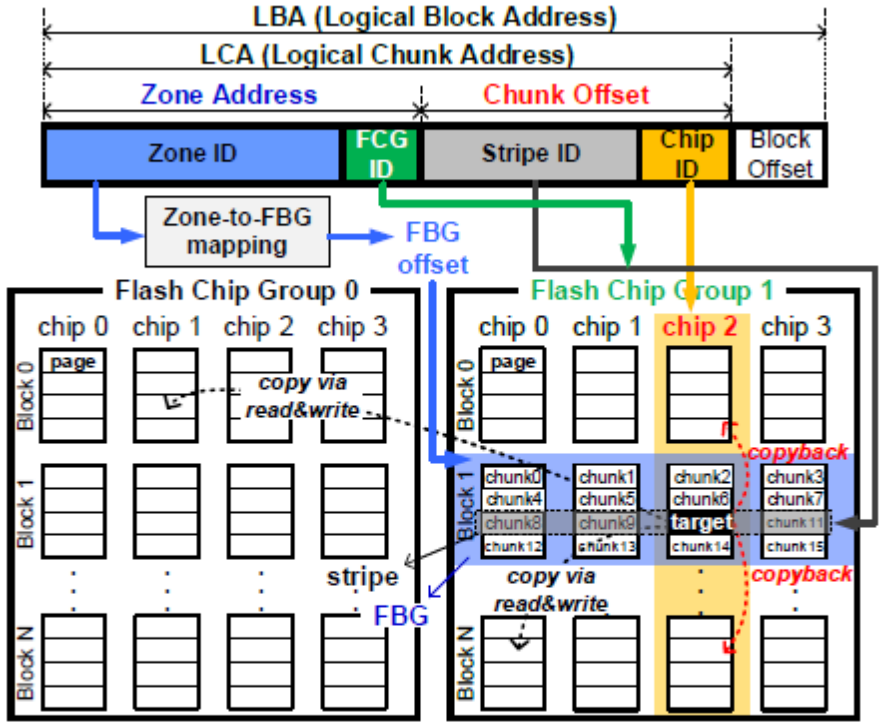
Source: *Id.* at 32.



Source: Bae, at 4.



Source: Microsoft Denali Presentation, at 15.

		 Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted. Source: Han, at 149.
1g	for each of the data write requests, identify from	Samsung's Accused Instrumentalities contain said circuitry to, for each of the data write requests, identify from the first division operation a third address portion, and use a second division operation on the third address portion to identify an address value to select one of the erase units from the subset which

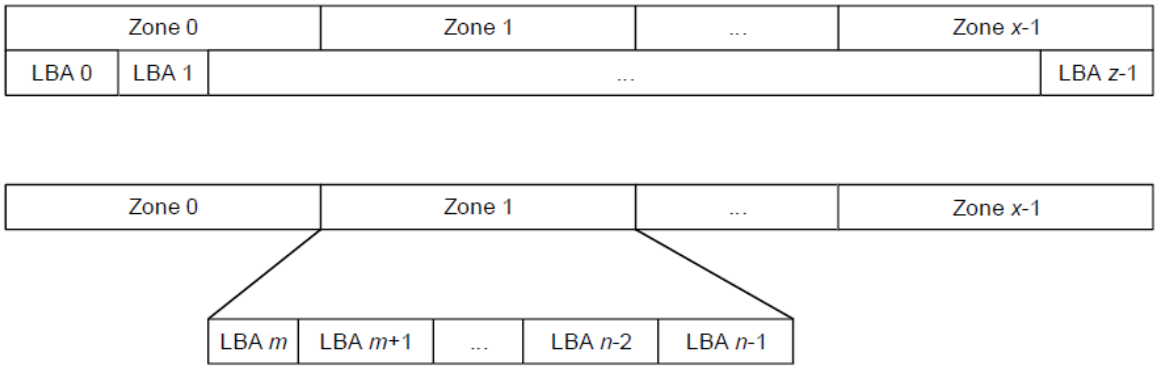
	the first division operation a third address portion, and use a second division operation on the third address portion to identify an address value to select one of the erase units from the subset which corresponds to the addressed segment and to identify a storage location within the selected one of the erase units;	corresponds to the addressed segment and to identify a storage location within the selected one of the erase units. Samsung's Accused Instrumentalities are able to identify from the first division operation a third address portion, which is the calculated remainder of said division operation. Samsung's Accused Instrumentalities are able to use such remainder to identify an address value to select one of the erase units corresponding to the addressed zone (segment), and to identify a logical block and/or page (storage location) within such erase unit, using a second division operation. Samsung's Accused Instrumentalities contain multiple equally sized erase units in each zone such that a second division operation is used to identify a particular erase unit within a zone. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

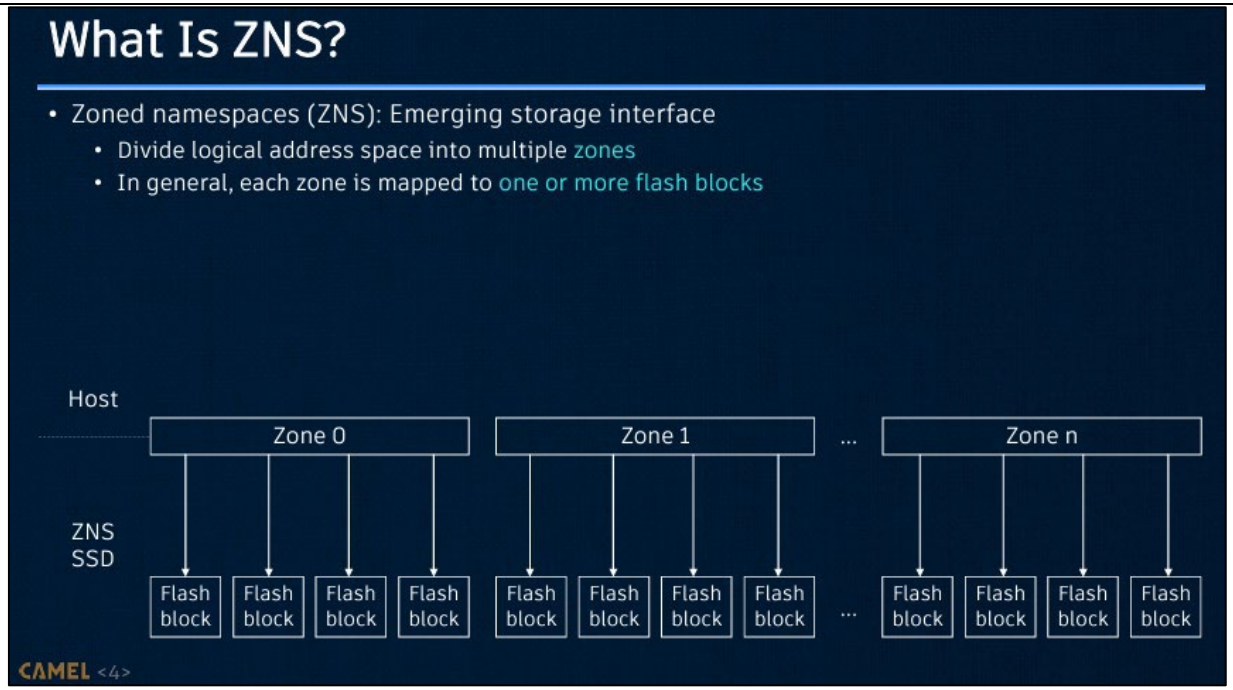
Figure 2: Zones in a Zoned Namespace



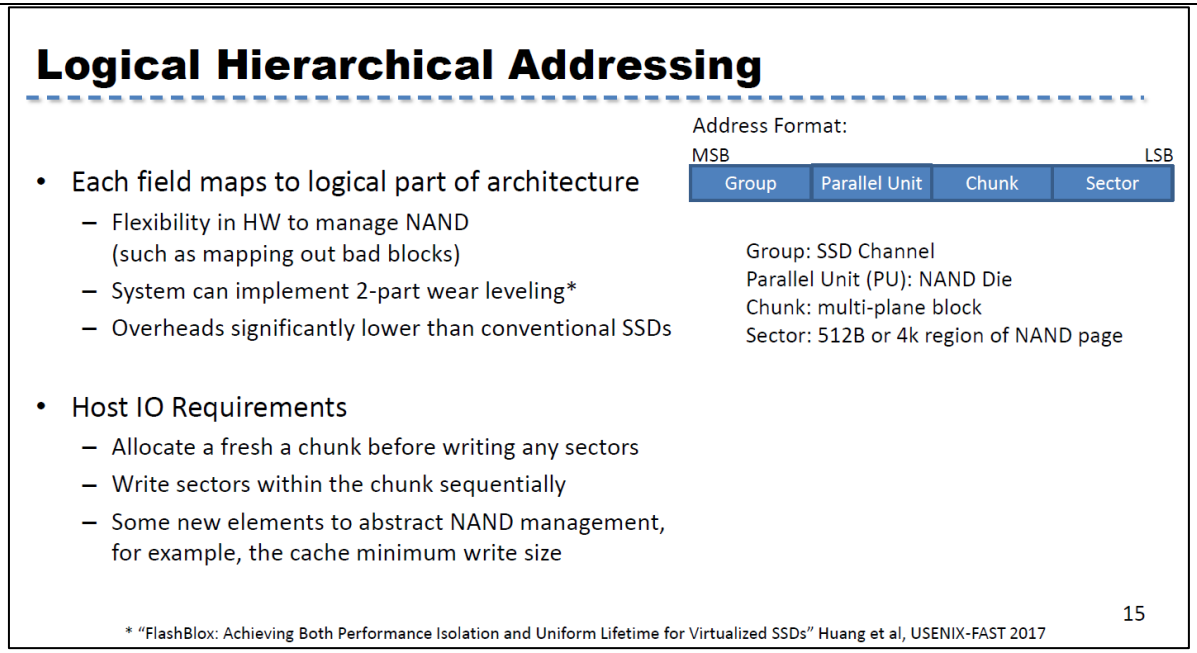
The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

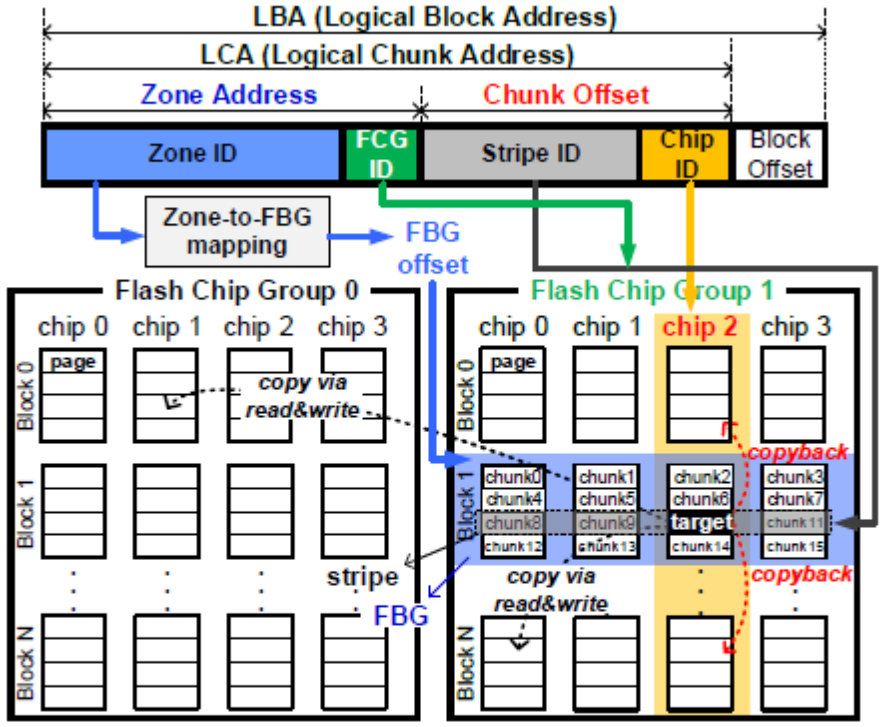
Source: ZNS Specification, at 8.



Source: Bae, at 4.



Source: Microsoft Denali Presentation, at 15.

		 Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted. Source: Han, at 149.
1h	wherein said circuitry is to execute each one of the data	Samsung's Accused Instrumentalities contain said circuitry to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the data write requests.

	write requests by programming accompanying data into the storage location that was identified for that one of the data write requests; and	Samsung's Accused Instrumentalities are able to execute data write requests by writing (programming) data into the identified logical block and/or page (storage location). See: **3.3.7 Write command** The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2). **3.3.7.1 Command Completion** Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19. **Figure 19: Write – Command Specific Status Values** **Value** / **Description** 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. BCh — **Zone Invalid Write:** The write to a zone was not at the write pointer. 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. Source: ZNS Specification, at 20.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107. **Figure 69: Write – Command Dword 10 and Command Dword 11** **Bits** / **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. Source: NVM Command Set Specification, at 54.
1i	wherein said circuitry is also to execute each one of the incoming erase requests by controlling erasure in the flash memory	Samsung's Accused Instrumentalities contain said circuitry to execute each one of the incoming erase requests by controlling erasure in the flash memory of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests. Samsung's Accused Instrumentalities are able to execute the Zone Management Send command specifying a Reset Zone action (erase request) by controlling erasure of one or more of the erase units corresponding to the addressed zone (segment). See:

	of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests.	The addressed zone was identified from the incoming address information accompanying the one of the incoming erase requests. See claim element [1f]. **Figure 39: Zone Management Send – Command Dword 13** **Bits** / **Description** 31:09 — Reserved 08 — **Select All:** If this bit is set to '1', then the SLBA field is 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. **Value** / **Description** / **Refer to section** 00h — Reserved 01h — **Close Zone:** Close one or more zones. — 3.4.3.1.1 02h — **Finish Zone:** Finish one or more zones. — 3.4.3.1.2 03h — **Open Zone:** Open one or more zones. — 3.4.3.1.3 04h — **Reset Zone:** Reset one or more zones. — 3.4.3.1.4 05h — **Offline Zone:** Offline one or more zones. — 3.4.3.1.5 06h to 0Fh — Reserved 10h — **Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone. — 3.4.3.1.6 11h to FFh — Reserved Source: ZNS Specification, at 29. **3.4.3.1.4 Reset Zone** If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the: a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

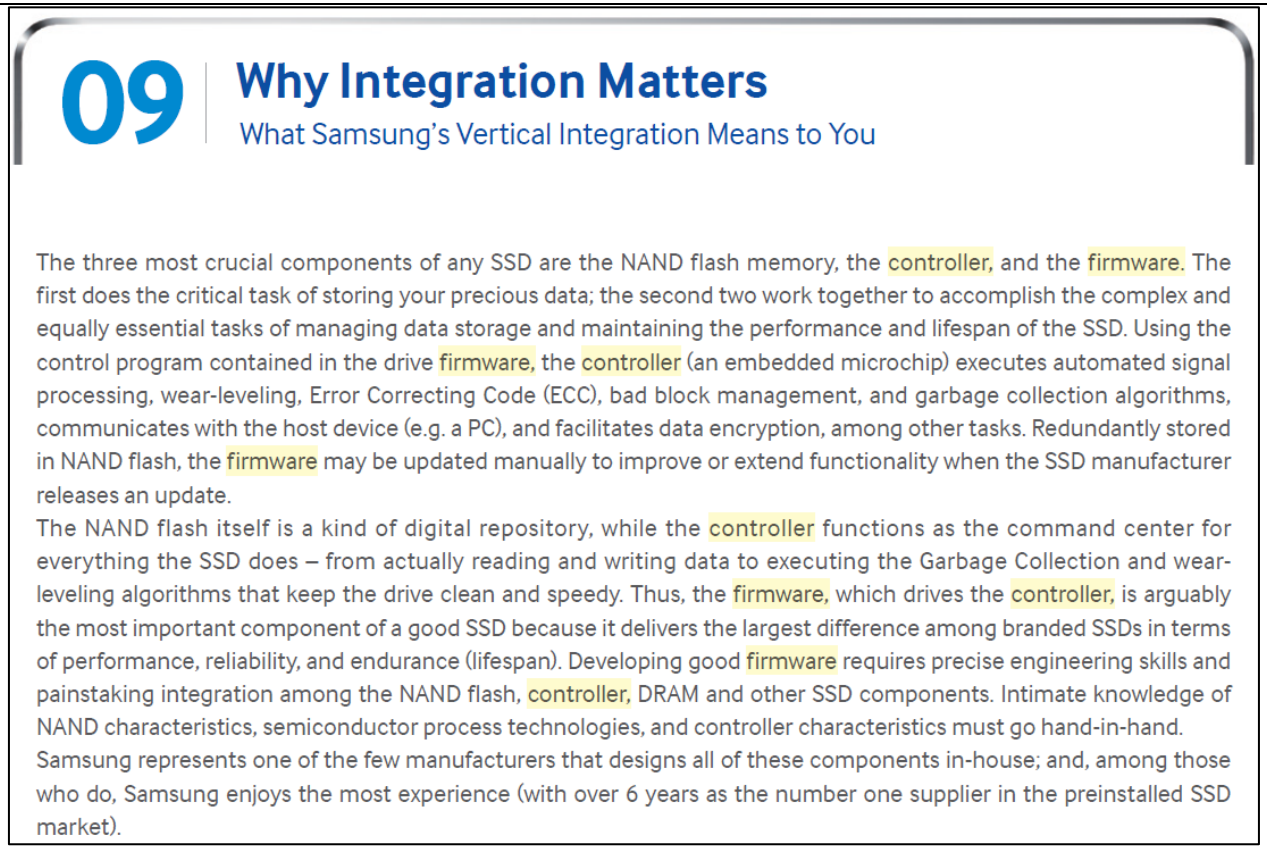
MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17. Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP).

		Source: Samsung Over-Provisioning White Paper, at 2 (highlighted). – *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block. Source: Boukhobza, at 22 (highlighted). **2.2.1. *Erase-before-write constraint*** Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: Boukhobza at 24 (highlighted).
2	The storage device of claim 1 wherein: said circuitry comprises at least one processor; and the storage device further comprises instructions stored on at	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1. Samsung's Accused Instrumentalities comprise said circuitry comprising at least one processor and instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor. See:

	least one physical storage medium, said instructions when executed to control the at least one processor.	 Source: Samsung PM1733 Product Brief.

		 09 Why Integration Matters What Samsung's Vertical Integration Means to You The three most crucial components of any SSD are the NAND flash memory, the controller, and the firmware. The first does the critical task of storing your precious data; the second two work together to accomplish the complex and equally essential tasks of managing data storage and maintaining the performance and lifespan of the SSD. Using the control program contained in the drive firmware, the controller (an embedded microchip) executes automated signal processing, wear-leveling, Error Correcting Code (ECC), bad block management, and garbage collection algorithms, communicates with the host device (e.g. a PC), and facilitates data encryption, among other tasks. Redundantly stored in NAND flash, the firmware may be updated manually to improve or extend functionality when the SSD manufacturer releases an update. The NAND flash itself is a kind of digital repository, while the controller functions as the command center for everything the SSD does – from actually reading and writing data to executing the Garbage Collection and wear-leveling algorithms that keep the drive clean and speedy. Thus, the firmware, which drives the controller, is arguably the most important component of a good SSD because it delivers the largest difference among branded SSDs in terms of performance, reliability, and endurance (lifespan). Developing good firmware requires precise engineering skills and painstaking integration among the NAND flash, controller, DRAM and other SSD components. Intimate knowledge of NAND characteristics, semiconductor process technologies, and controller characteristics must go hand-in-hand. Samsung represents one of the few manufacturers that designs all of these components in-house; and, among those who do, Samsung enjoys the most experience (with over 6 years as the number one supplier in the preinstalled SSD market). Source: Source: Samsung SSD White Paper, at 30.
3pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
3a	said circuitry can receive a request from a host to copy host-selected	Samsung's Accused Instrumentalities comprise said circuitry able to receive a request from a host to copy host-selected data from one storage location in flash memory to another storage location in the flash memory, and is to issue commands to the flash memory, in response to the request from the host to copy, to read the host-selected data from a first physical storage location in the flash memory and write the

	data from one storage location in flash memory to another storage location in the flash memory, and is to issue commands to the flash memory, in response to the request from the host to copy, to read the host-selected data from a first physical storage location in the flash memory and write the host-selected data to a second physical storage location in the flash memory without sending the data being copied to the host;	host-selected data to a second physical storage location in the flash memory without sending the data being copied to the host. Samsung's Accused Instrumentalities are able to receive and execute a Copy command from a host. A Copy command is a request to copy host-selected data from one storage location in flash memory to another storage location in the flash memory. Samsung's Accused Instrumentalities are able to issue commands to the flash memory, in response to the request from the host to copy, to read the host-selected data from a first physical storage location in the flash memory and write the host-selected data to a second physical storage location in the flash memory without sending the data being copied to the host. See: **3.3.2 Copy command** The Copy command is used by the host to copy user data from one or more source ranges in one or more source namespaces to a single consecutive destination logical block range in a destination namespace (i.e., the namespace specified by the NSID field). Each source range may be in the same namespace or a different namespace with respect to any other source range and with respect to the destination logical block range. Source: NVM Command Set Specification at 29.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

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

The data that the Copy command provides is a list of Source Range entries that describe the data to be copied to the destination range starting at the LBA specified by the SDLBA field. The Copy Descriptor Format type of the Source Range entries is specified in the Descriptor Format field in Command Dword 12. The Copy Descriptor Format types are distinguished by the supported protection information formats (refer to section 5.3.1) and whether the Copy Descriptor Format contains a Source Namespace Identifier (SNSID) field that supports a copy source in a different namespace than the copy destination, as described in Figure 38. For a Copy Descriptor Format that does not contain an SNSID field, the source namespace is the same as the destination namespace which is specified by the NSID field in the command.

Figure 39: Copy – Source Range Entries Copy Descriptor Format 0h and Format 2h

<table>
<tr><th>Range</th><th>Bytes</th><th>Description</th></tr>
<tr><td rowspan="3">Source Range 0</td><td>03:00</td><td>Source Parameters (SPARS): This field specifies attributes as follows:
<table>
<tr><th>Format 0h bytes</th><th>Format 2h bytes</th><th>Description</th></tr>
<tr><td>03:00</td><td>n/a</td><td>Reserved</td></tr>
<tr><td>n/a</td><td>03:00</td><td>Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.</td></tr>
</table></td></tr>
<tr><td>07:04</td><td>Reserved</td></tr>
<tr><td>15:08</td><td>Starting LBA (SLBA)</td></tr>
</table>

Range	Bytes	Description
	19:16	**Read Parameters (RPARS):** This field specifies attributes as follows. **Bits** / **Description** 31:20 – Reserved 19:16 – **Command Extension Type (CETYPE):** Specifies the Command Extension Type that applies to the command (refer to the Key Per I/O section in the NVM Express Base Specification). This field is used for the read portion of the copy operation for the LBAs specified in this Source Range entry. 15:00 – **Number of Logical Blocks (NLB):** This field indicates the number of logical blocks to be copied. This is a 0's based value.

Figure 40: Copy – Source Range Entries Copy Descriptor Format 1h and Format 3h

Range	Bytes	Description
Source Range 0	03:00	**Source Parameters (SPARS)**: This field specifies attributes as follows: **Format 1h bytes** / **Format 3h bytes** / **Description** 03:00 / n/a / Reserved n/a / 03:00 / **Source Namespace Identifier (SNSID):** Specifies the source namespace for this Source Range entry.
	07:04	Reserved
	15:08	**Starting LBA (SLBA)**
	19:16	**Read Parameters (RPARS):** This field specifies attributes as follows: **Bits** / **Description** 31:20 – Reserved 19:16 – **Command Extension Type (CETYPE):** Specifies the Command Extension Type that applies to the command (refer to the Key Per I/O section in the NVM Express Base Specification). This field is used for the read portion of the copy operation for the LBAs specified in this Source Range entry. 15:00 – **Number of Logical Blocks (NLB):** This field indicates the number of logical blocks to be copied. This is a 0's based value.

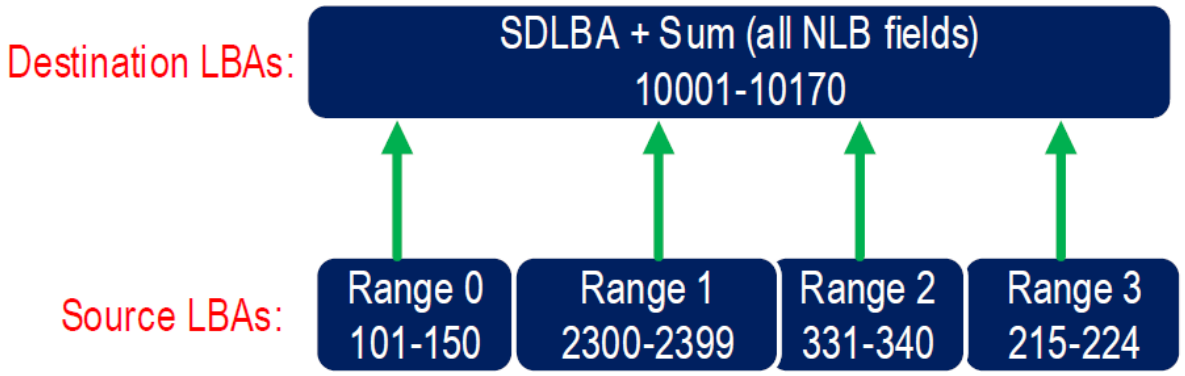
		Figure 41: Source LBA and Destination LBA Relationship Example  Source: NVM Command Set Specification at 29–37.
3b	the request from the host to copy is to designate destination address information corresponding to the second physical storage location; and	Samsung's Accused Instrumentalities comprise circuitry able to receive the request from the host to copy which is to designate destination address information corresponding to the second physical storage location. Samsung's Accused Instrumentalities are able to receive and execute the Copy command which designates destination address information corresponding to the second physical storage location. See: **3.3.2 Copy command** The Copy command is used by the host to copy user data from one or more source ranges in one or more source namespaces to a single consecutive destination logical block range in a destination namespace (i.e., the namespace specified by the NSID field). Each source range may be in the same namespace or a different namespace with respect to any other source range and with respect to the destination logical block range. Source: NVM Command Set Specification at 29.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

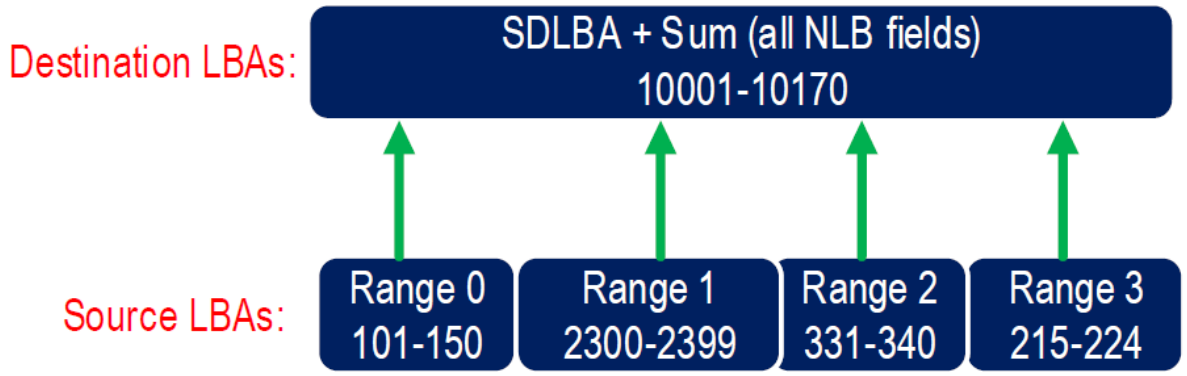
Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

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

The data that the Copy command provides is a list of Source Range entries that describe the data to be copied to the destination range starting at the LBA specified by the SDLBA field. The Copy Descriptor Format type of the Source Range entries is specified in the Descriptor Format field in Command Dword 12. The Copy Descriptor Format types are distinguished by the supported protection information formats (refer to section 5.3.1) and whether the Copy Descriptor Format contains a Source Namespace Identifier (SNSID) field that supports a copy source in a different namespace than the copy destination, as described in Figure 38. For a Copy Descriptor Format that does not contain an SNSID field, the source namespace is the same as the destination namespace which is specified by the NSID field in the command.

Figure 41: Source LBA and Destination LBA Relationship Example



Source: NVM Command Set Specification at 29–37.

3c	the circuitry is to use the first division operation on the designated destination address information to identify a destination segment within a destination block device.	Samsung's Accused Instrumentalities comprise circuitry to use the first division operation on the designated destination address information to identify a destination segment within a destination block device. Samsung's Accused Instrumentalities are able to the first division operation on the designated destination address information to identify a destination zone (segment) within a destination namespace (block device). See: **3.3.2 Copy command** The Copy command is used by the host to copy user data from one or more source ranges in one or more source namespaces to a single consecutive destination logical block range in a destination namespace (i.e., the namespace specified by the NSID field). Each source range may be in the same namespace or a different namespace with respect to any other source range and with respect to the destination logical block range. Source: NVM Command Set Specification at 29.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

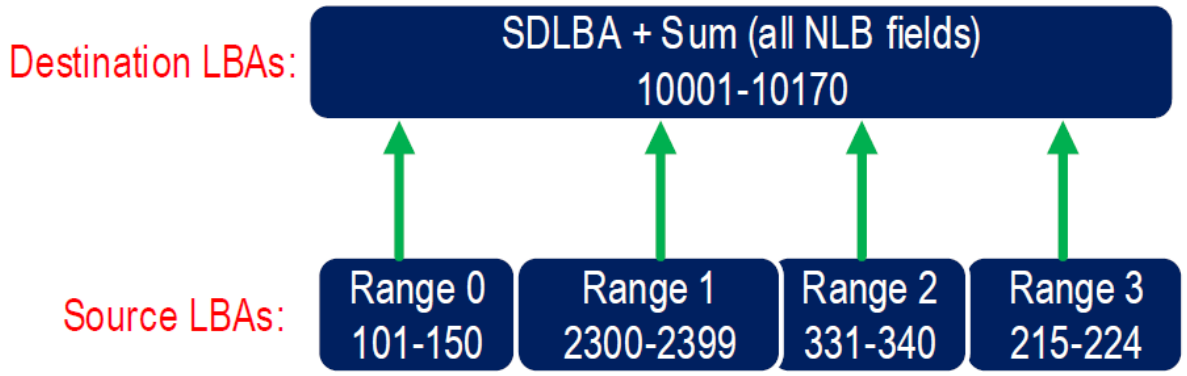
Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

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

The data that the Copy command provides is a list of Source Range entries that describe the data to be copied to the destination range starting at the LBA specified by the SDLBA field. The Copy Descriptor Format type of the Source Range entries is specified in the Descriptor Format field in Command Dword 12. The Copy Descriptor Format types are distinguished by the supported protection information formats (refer to section 5.3.1) and whether the Copy Descriptor Format contains a Source Namespace Identifier (SNSID) field that supports a copy source in a different namespace than the copy destination, as described in Figure 38. For a Copy Descriptor Format that does not contain an SNSID field, the source namespace is the same as the destination namespace which is specified by the NSID field in the command.

Figure 41: Source LBA and Destination LBA Relationship Example



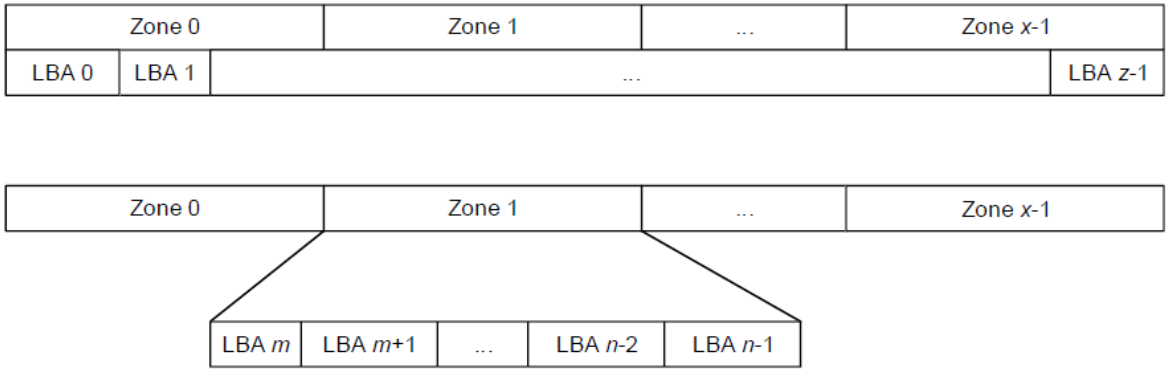
Source: NVM Command Set Specification at 29–37.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

3.3.2 Copy command

The Copy command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

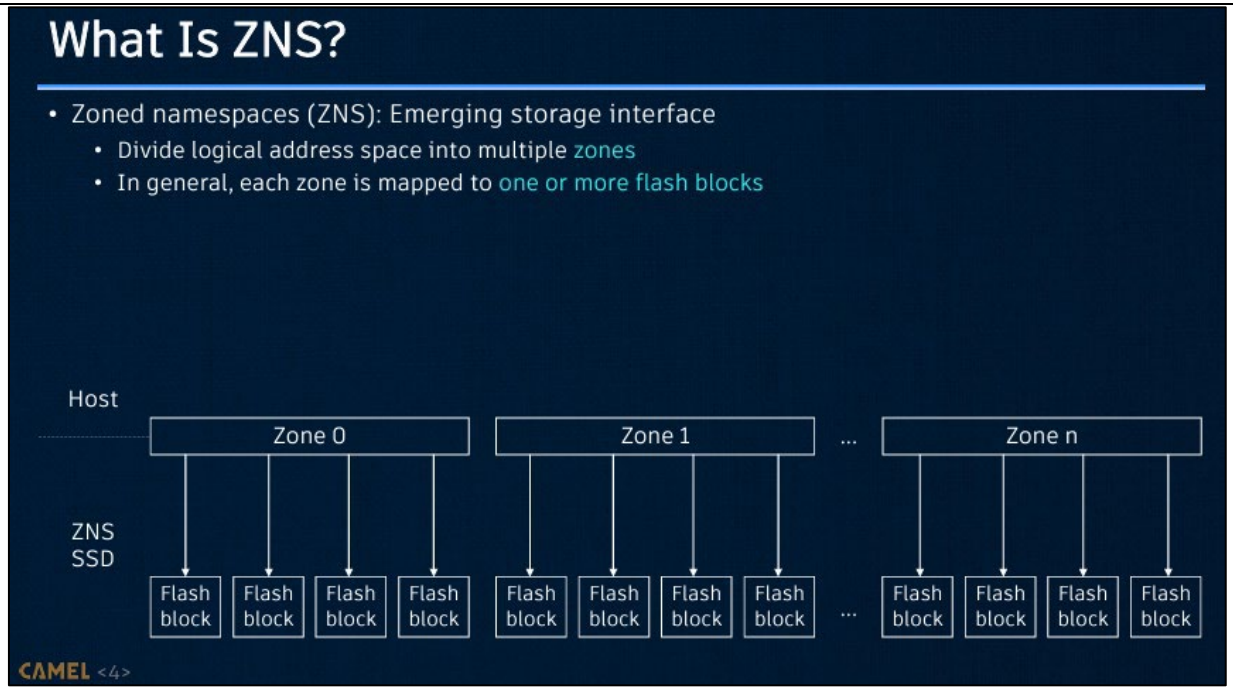
3.3.2.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 14.

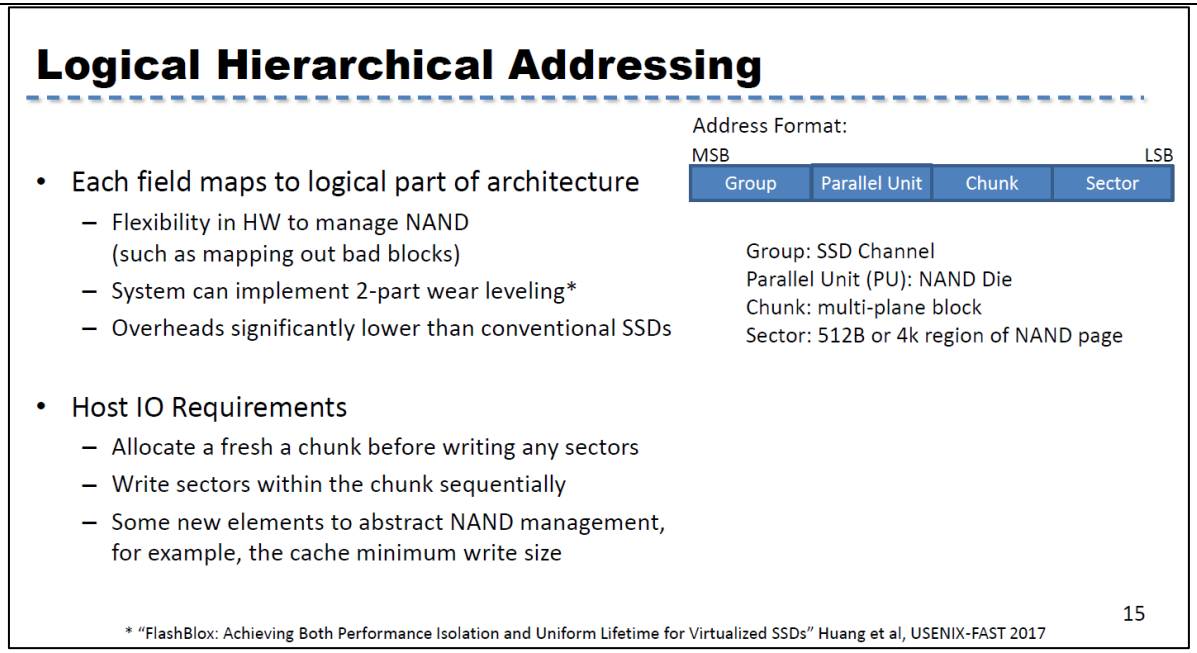
Figure 14: Copy – Command Specific Status Values

Value	Description
B8h	**Zone Boundary Error:** The command specifies a Source Range Entry that contains logical blocks in more than one zone or a destination LBA range that contains logical blocks in more than one zone.
B9h	**Zone Is Full:** The zone specified by the destination LBA range is in the ZSF:Full state.
BAh	**Zone Is Read Only:** The zone specified by the destination LBA range is in the ZSRO:Read Only state.
BBh	**Zone Is Offline:** A zone specified by a Source Range Entry or the zone specified by the destination LBA range is in the ZSO:Offline state.
BCh	**Zone Invalid Write:** The write to the zone was not at the write pointer.
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.

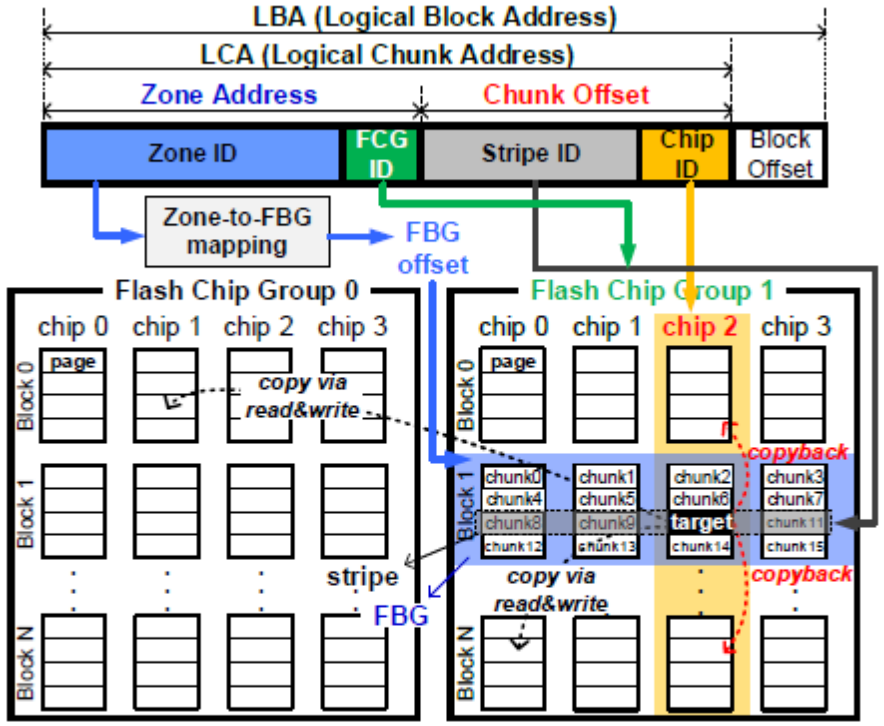
Source: ZNS Specification, at 19.

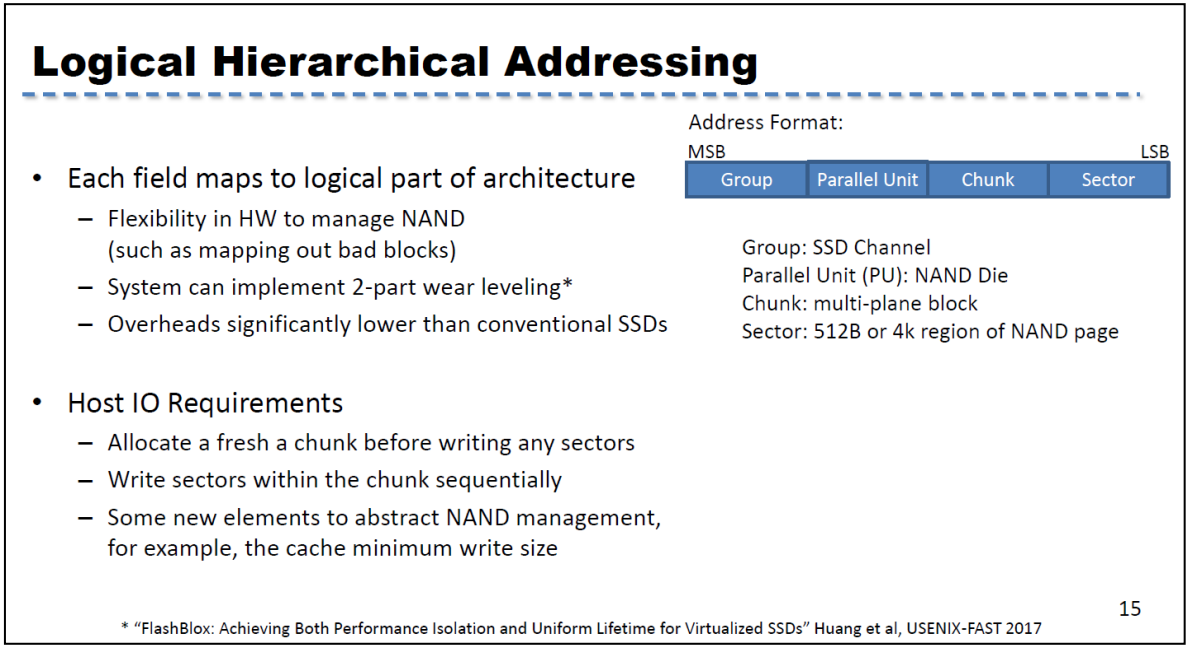


Source: Bae, at 4.



Source: Microsoft Denali Presentation, at 15.

		 Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted. Source: Han, at 149.
4	The storage device of claim 3 wherein the circuitry is to	Samsung's Accused Instrumentalities comprise the storage device of claim 3. See claim 3.

	identify from the destination address information a destination logical erase unit and is to select the second physical storage location to be within a physical erase unit that is already mapped to the destination logical erase unit.	Samsung's Accused Instrumentalities comprise the circuitry to identify from the destination address information a destination logical erase unit and to select the second physical storage location to be within a physical erase unit that is already mapped to the destination logical erase unit. Samsung's Accused Instrumentalities are able to identify a destination logical erase unit from the destination address information and to select the second storage location to be within a physical block (erase unit) mapped to the destination logical erase unit. For example, Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones, and they maintain a mapping of logical erase units to physical erase units to enable them to do this. See:  Source: Microsoft Denali Presentation, at 15.

		Bad Block Management & Error Correcting Code (ECC) In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance. Source: Samsung SSD White Paper, at 19. **2. Why is OP important?** OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.

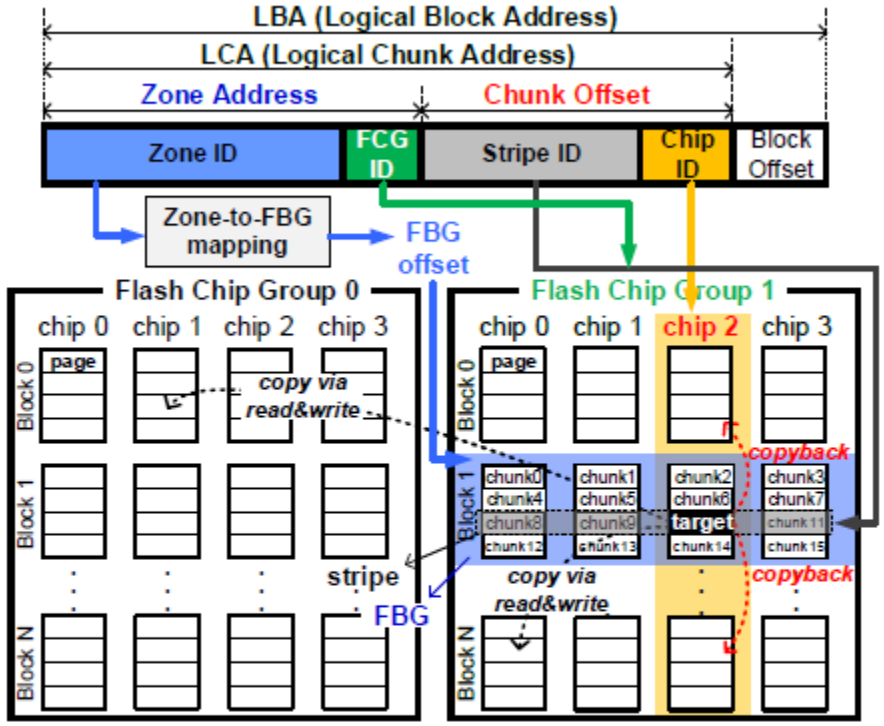


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein because the accused functionality would be insubstantially different from or function in the same way to provide the same result as the relevant claim element.

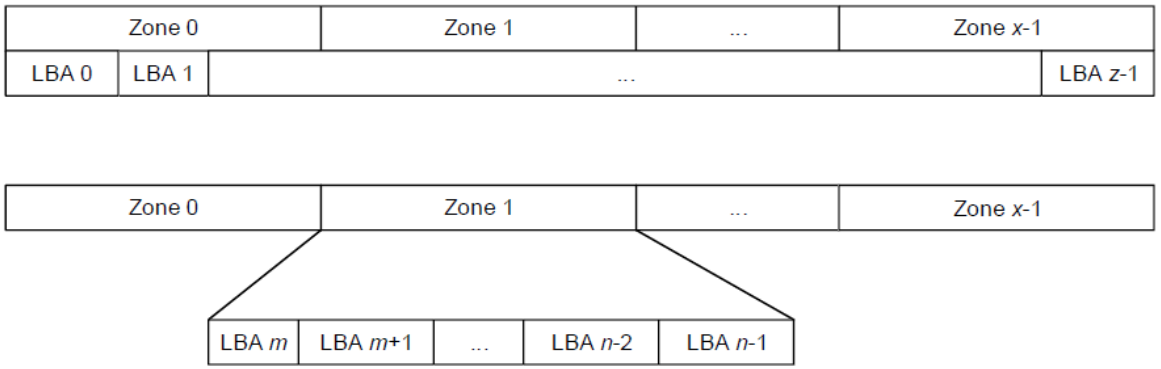
		As an example, there is no substantial difference between (1) identifying a destination logical erase unit and selecting a physical storage location within a physical erase unit that is mapped to the destination logical erase unit and (2) identifying a destination logical erase unit group as well as an offset within the erase unit group and selecting a physical storage location within a physical erase unit that corresponds to the destination logical erase unit group as well as the offset within the erase unit group. The two ways of identification and selection perform substantially the same function in substantially the same way to achieve substantially the same result.
8	The storage device of claim 1 wherein the third address portion specifies a location within a sequential range of locations that correspond to the erase units in the subset.	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1. Samsung's Accused Instrumentalities comprise the storage device of claim 1 wherein the third address portion specifies a location within a sequential range of locations that correspond to the erase units in the subset. The calculated remainder of the first division operation (third address portion) specifies a location of a logical block, page, and/or erase unit within a sequential range of locations (such as a sequential range of logical and/or physical block addresses) that correspond to the erase units of a zone. For example, the third address portion can specify a logical block address (LBA) within a sequential range of LBAs of a zone. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

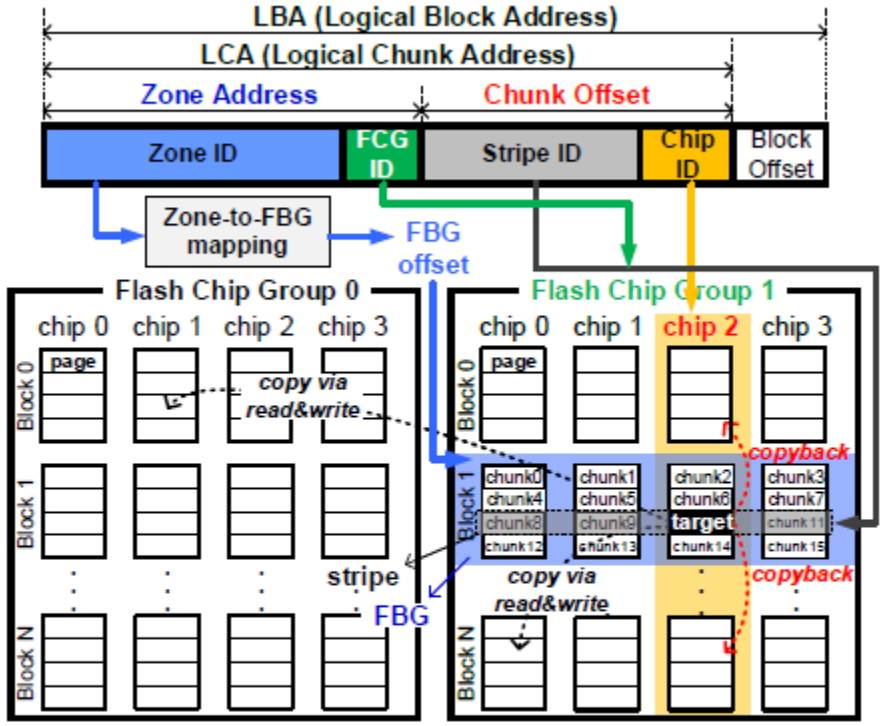
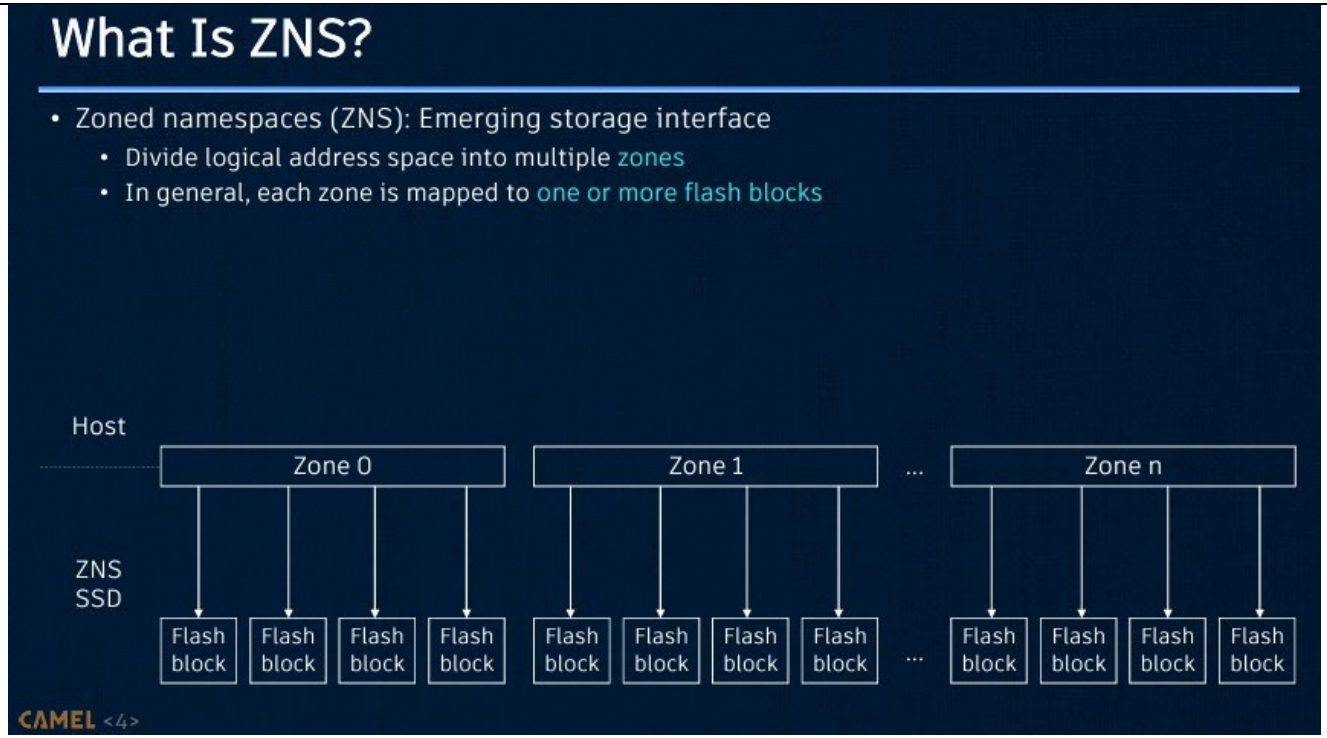
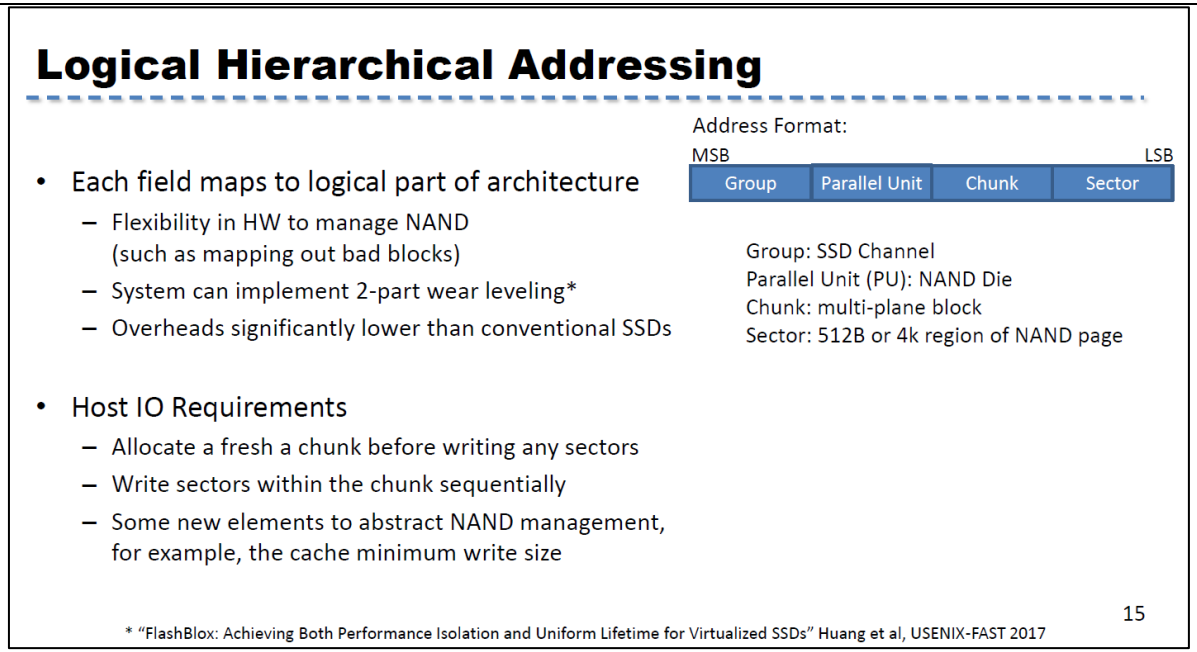


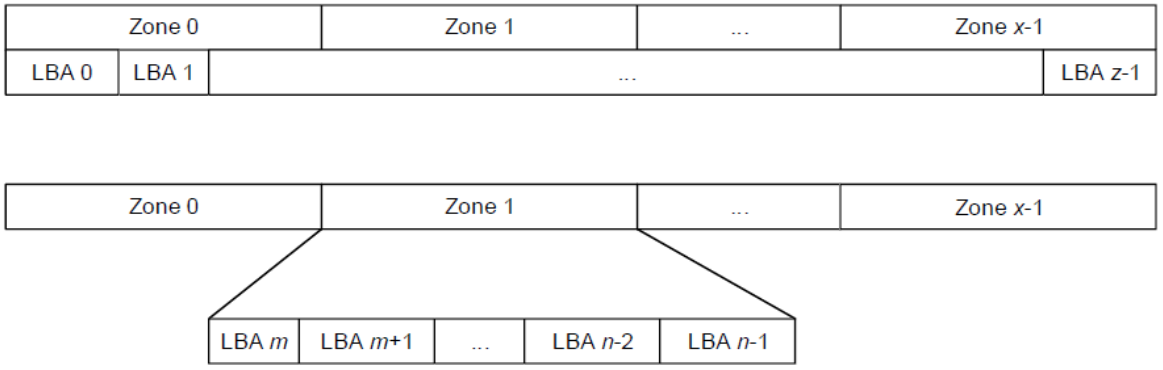
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

		 Source: Microsoft Denali Presentation, at 15.
9	The storage device of claim 1 wherein the first division operation and the second division operation each subdivide a value obtained from the incoming address	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1. Samsung's Accused Instrumentalities comprise the storage device of claim 1 wherein the first division operation and the second division operation each subdivide a value obtained from the incoming address information accompanying each of the write requests into respective bitfields. Samsung's Accused Instrumentalities are able to use the first division operation to subdivide the starting LBA into respective bitfields and to use the second division operation to subdivide the calculated remainder of the first division operation into respective bitfields. See:

	information accompanying each of the write requests into respective bitfields.	**2.1.1 Namespaces** A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification. A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA. **Figure 2: Zones in a Zoned Namespace**  The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification). Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors. Source: ZNS Specification, at 8.

3.3.7 Write command

The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

3.3.7.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19.

Figure 19: Write – Command Specific Status Values

Value	Description
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.
BCh	**Zone Invalid Write:** The write to a zone was not at the write pointer.
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.

Source: *Id.* at 20.

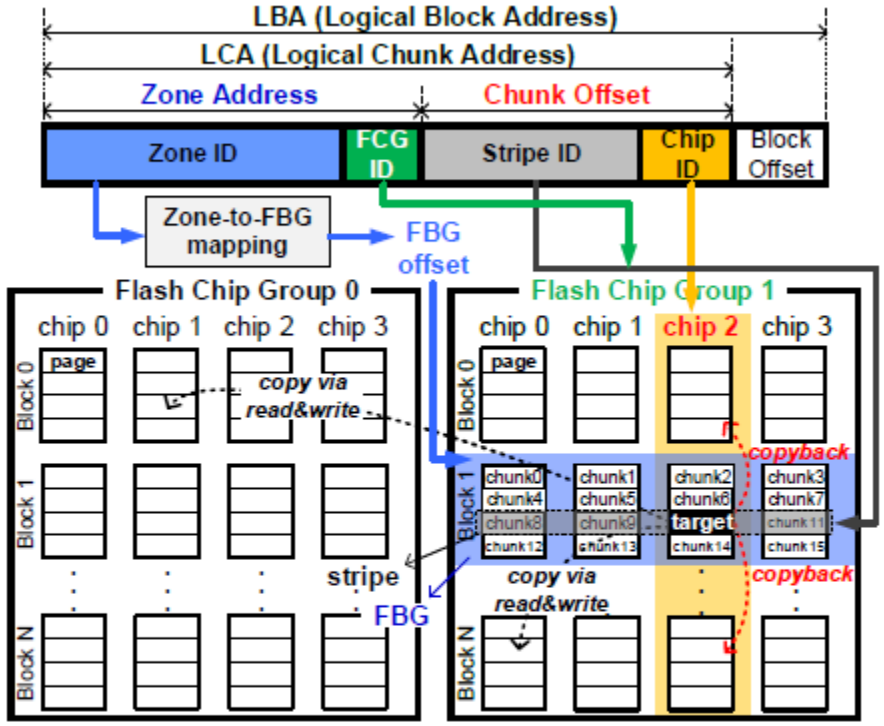
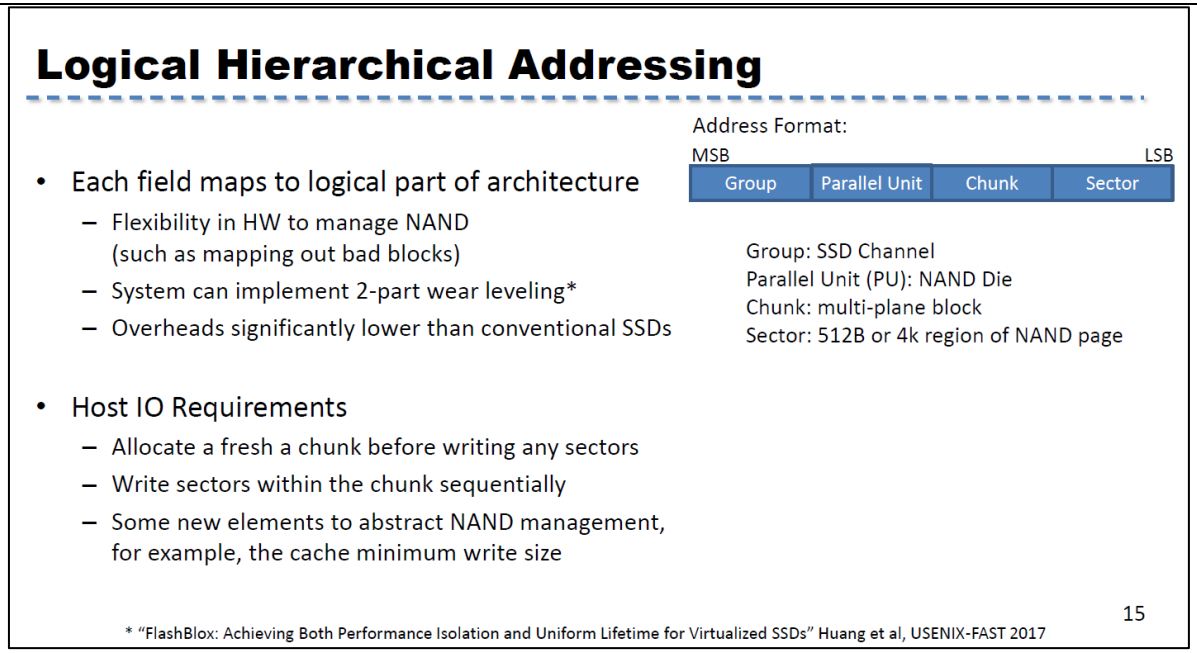
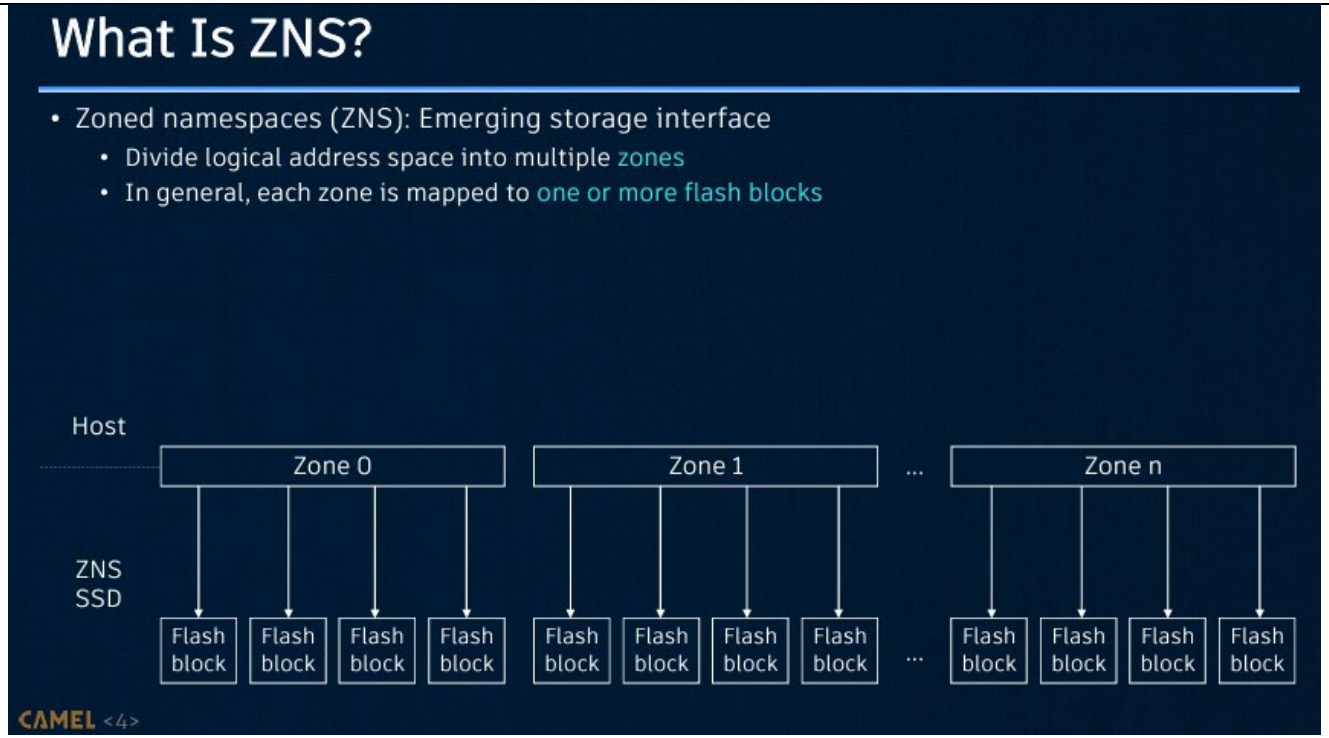


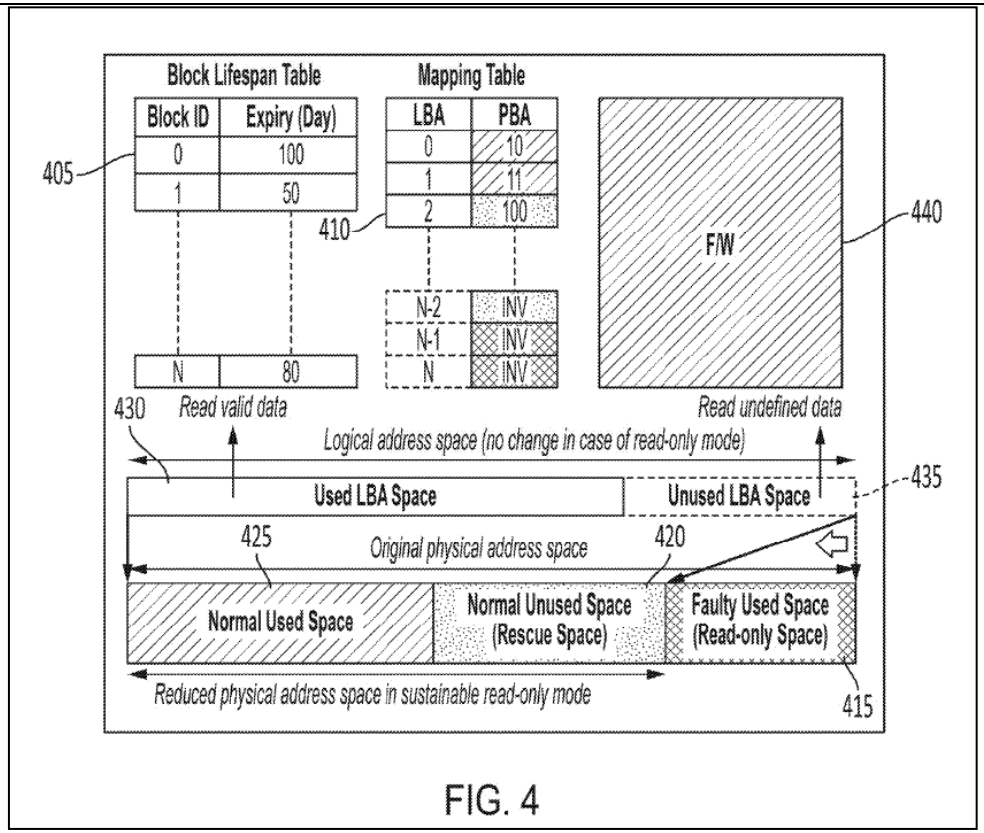
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Microsoft Denali Presentation, at 15.

		 Source: Bae, at 4.
10pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
10a	the storage device is to store, for each segment having an erase unit which stores valid data, a	Samsung's Accused Instrumentalities comprise the storage device to store, for each segment having an erase unit which stores valid data, a value representing a time since data was programmed. Samsung's Accused Instrumentalities are able to store metadata for each zone (segment) having an erase unit which stores valid data. This metadata includes values indicating a time since data was programmed, including metadata such as age of stored data or a data retention period. See:

	value representing a time since data was programmed;	 FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

		[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.	

		[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].  Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

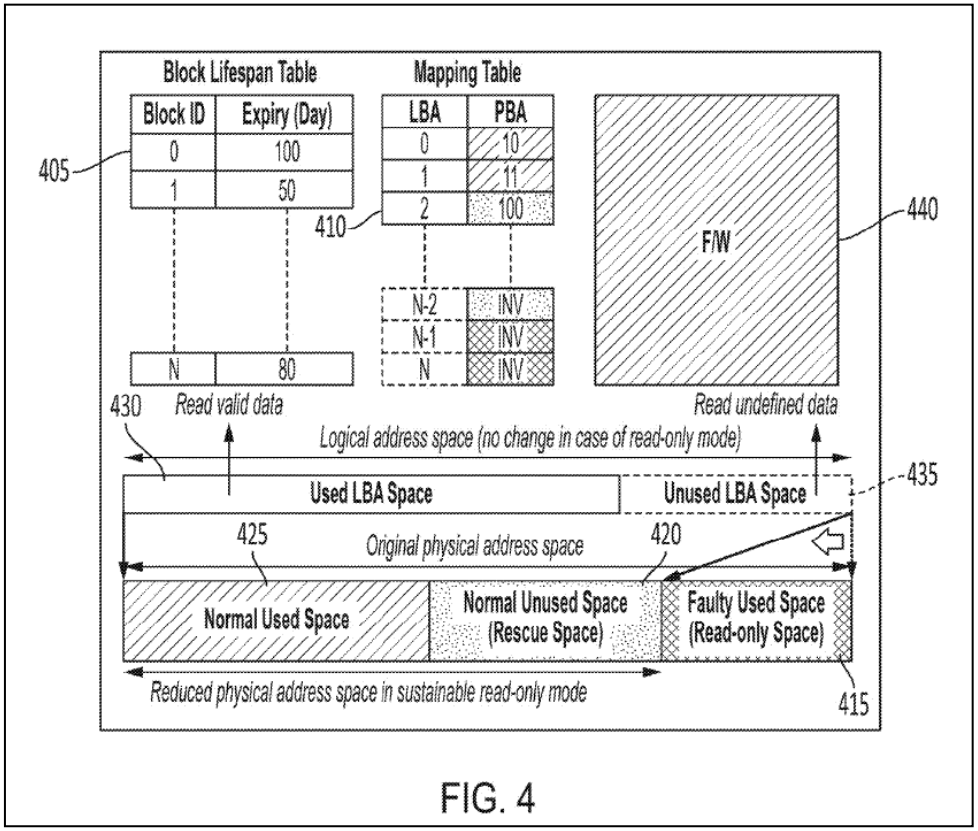
Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

		2.2.3. *Reliability limitation* A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09]. Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09]. Source: Boukhobza, at 26.
10b	the storage device is to compare the values	Samsung's Accused Instrumentalities comprise a storage device to compare the values representing time since data was programmed to at least one threshold.

	representing time since data was programmed to at least one threshold;	Samsung's Accused Instrumentalities are able to compare the metadata to at least one threshold. This metadata includes metadata representing a time since the data was programmed in the respective block, including metadata such as age of stored data or a data retention period. Samsung's Accused Instrumentalities are capable of determining when the maintained metadata exceeds a threshold (e.g., the age of data meets or exceeds a data retention period threshold). See:  FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

		[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.	

		[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].  Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

		2.2.3. *Reliability limitation* A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09]. Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09]. Source: Boukhobza, at 26.
10c	upon detection that a specific one of the values has	Samsung's Accused Instrumentalities comprise the circuitry to, upon detection that a specific one of the values has satisfied the one or more thresholds, copy data stored at one or more physical storage locations

	satisfied the one or more thresholds, the circuitry is to copy data stored at one or more physical storage locations associated with the specific one of the values from a first physical erase unit to a second physical erase unit; and	associated with the specific one of the values from a first physical erase unit to a second physical erase unit. Samsung's Accused Instrumentalities are able to copy data from one or more physical storage locations at one physical block (erase unit) to another physical block (erase unit) using internal operations, including for those blocks where maintained metadata representing time since data was programmed in a block meets or exceeds a threshold. See: **5.4 Reset Zone Recommended** A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the ZSF:Full state may notify host software to initiate a zone reset operation (refer to section 3.4.3.1.4). If a controller schedules such an internal operation on a zone, the controller may notify the host by: a) setting the Reset Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 37); b) setting the Reset Zone Recommended Time Limit zone attribute information to indicate the number of seconds before the controller intends to perform an internal operation on the specified zone; and c) generating a Zone Descriptor Changed event for the specific zone. As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed. If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by: a) clearing the Reset Zone Recommended zone attribute of the specific zone to '0'; and b) generating a Zone Descriptor Changed event for the specific zone. If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Reset Zone Recommended attribute shall be cleared to '0'. Source: ZNS Specification, at 41–42.

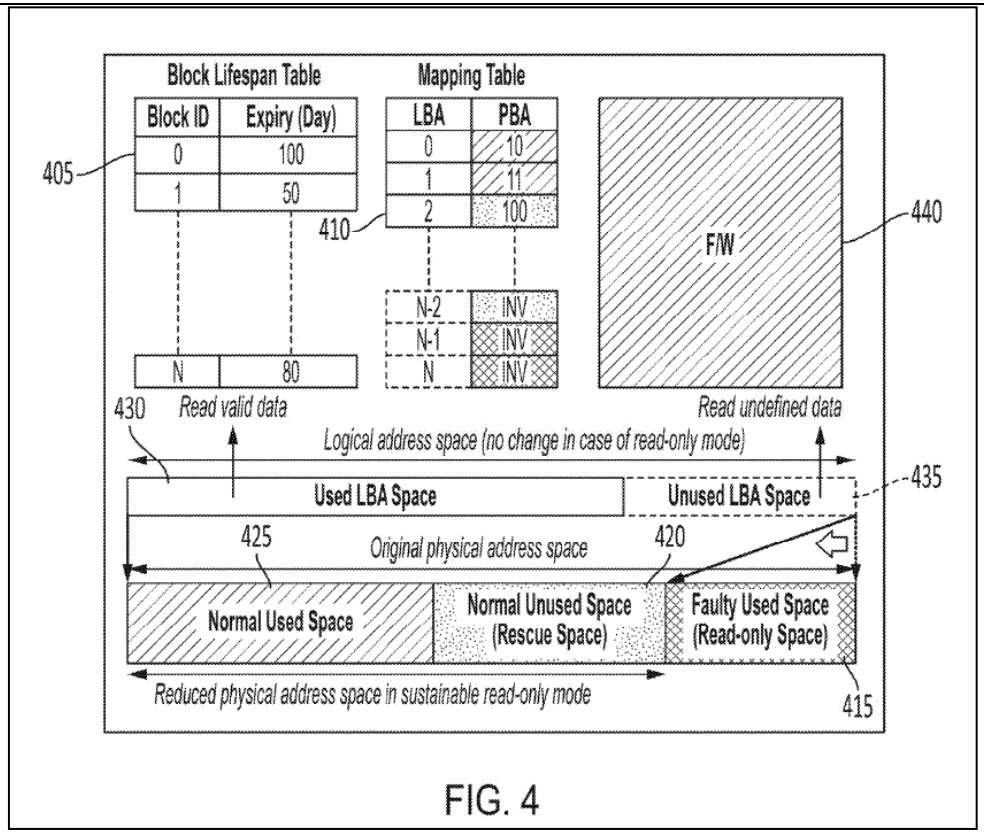

Block Lifespan Table
Block ID
Expiry (Day)
0
100
1
50
N
80
405
Mapping Table
LBA
PBA
0
10
1
11
2
100
410
N-2
INV
N-1
INV
N
INV
F/W
440
430
Read valid data
Read undefined data
Logical address space (no change in case of read-only mode)
Used LBA Space
Unused LBA Space
435
425
Original physical address space
420
Normal Used Space
Normal Unused Space
(Rescue Space)
Faulty Used Space
(Read-only Space)
Reduced physical address space in sustainable read-only mode
415

FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

		[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.	

		[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].  Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

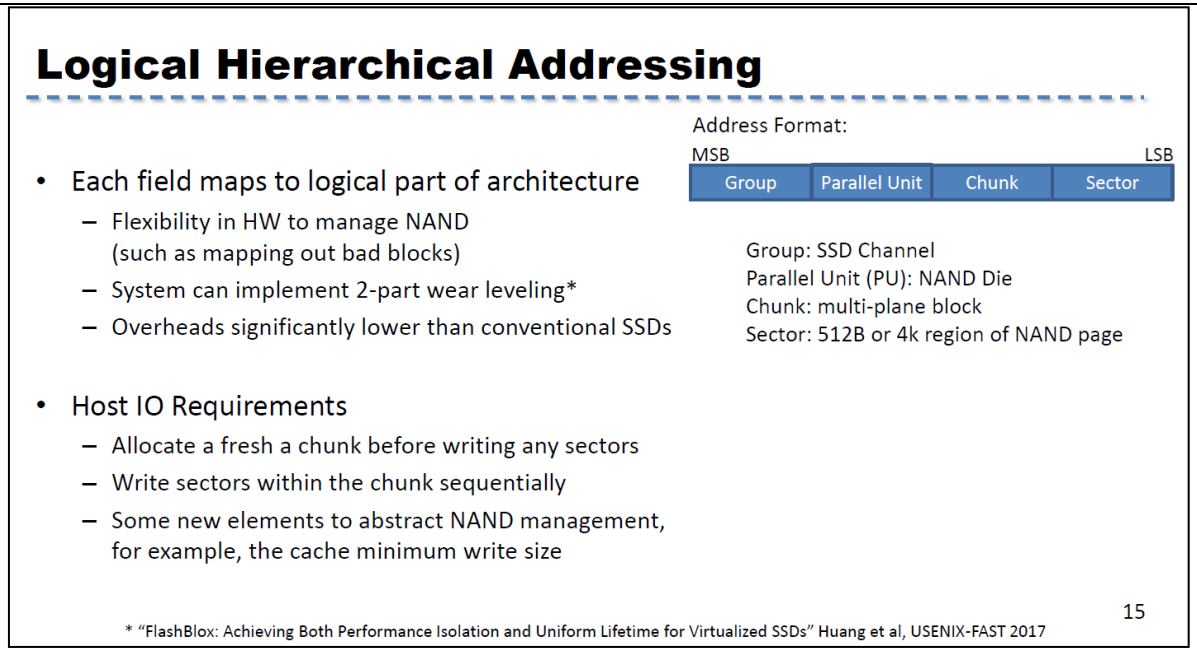
Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

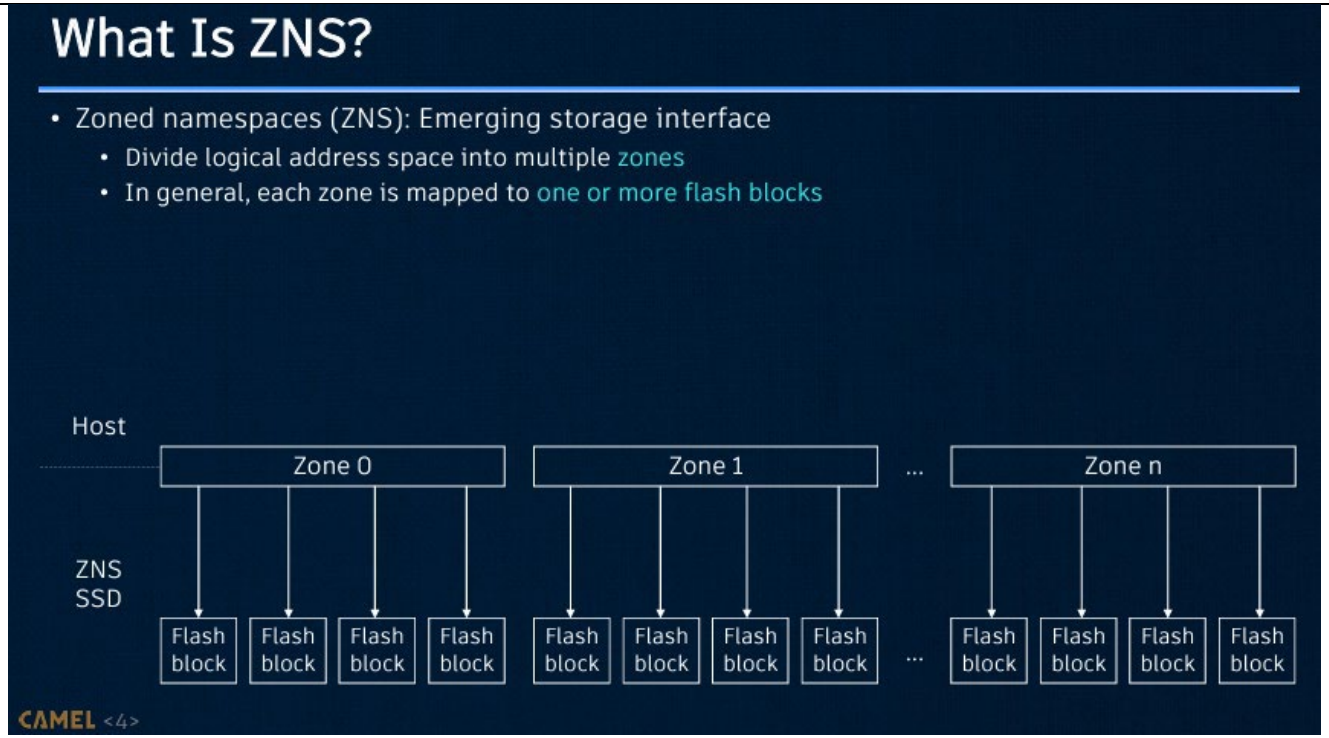
		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

		2.2.3. *Reliability limitation* A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09]. Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09]. Source: Boukhobza, at 26.
10d	the storage device is to remap a logical address for the	Samsung's Accused Instrumentalities comprise the storage device to remap a logical address for the data being copied from the first physical erase unit to the second physical erase unit.

	data being copied from the first physical erase unit to the second physical erase unit.	Samsung's Accused Instrumentalities are able to remap the correspondence of the logical address so that it instead corresponds to the second physical block (erase unit) after the data is copied from the first physical block to the second physical block. Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones, and they maintain a mapping of logical erase units to physical erase units to enable them to do this. See: **5.4 Reset Zone Recommended** A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the ZSF:Full state may notify host software to initiate a zone reset operation (refer to section 3.4.3.1.4). If a controller schedules such an internal operation on a zone, the controller may notify the host by: a) setting the Reset Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 37); b) setting the Reset Zone Recommended Time Limit zone attribute information to indicate the number of seconds before the controller intends to perform an internal operation on the specified zone; and c) generating a Zone Descriptor Changed event for the specific zone. As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed. If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by: a) clearing the Reset Zone Recommended zone attribute of the specific zone to '0'; and b) generating a Zone Descriptor Changed event for the specific zone. If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Reset Zone Recommended attribute shall be cleared to '0'. Source: ZNS Specification, at 41–42.



Source: Microsoft Denali Presentation, at 15.



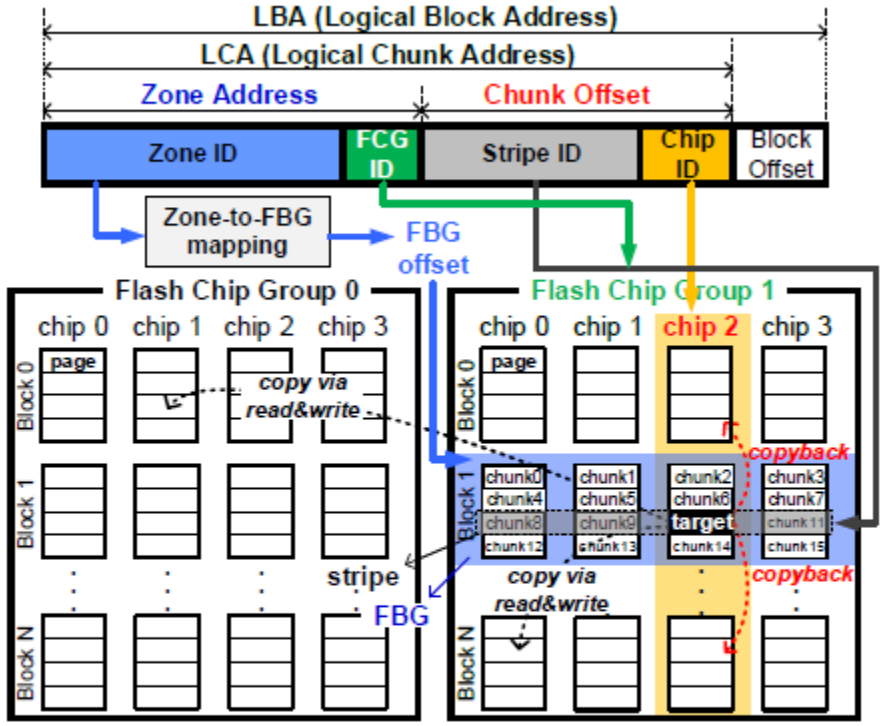
Source: Bae, at 4.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

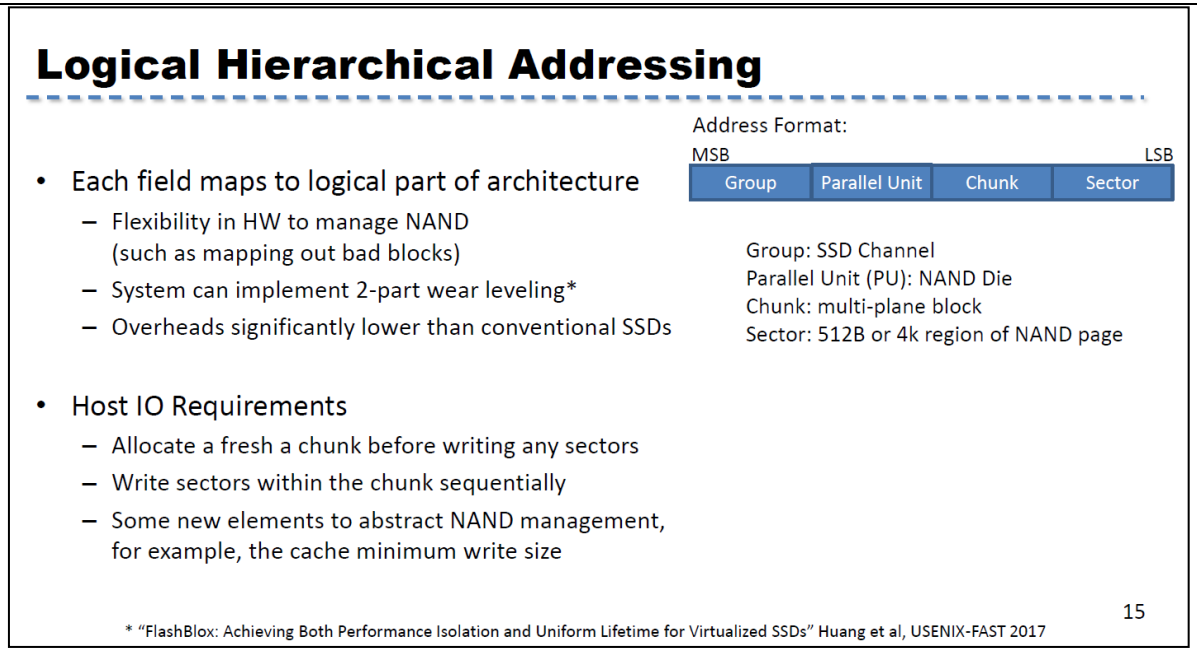
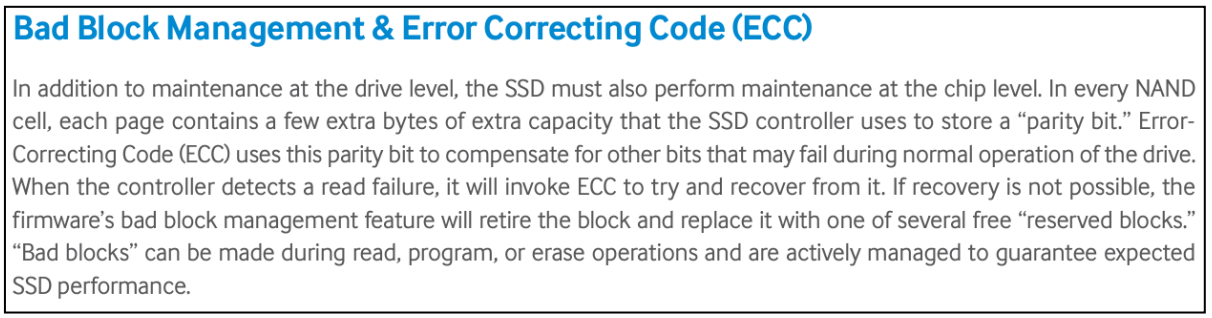
Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.

		 Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted. Source: Han, at 149.
11pre	The storage device of claim 10 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 10. See claim 10.

11a	the flash memory is to have block devices, one of the block devices being the addressed block device; and	Samsung's Accused Instrumentalities comprise the flash memory to have block devices, one of the block devices being the addressed block device. Samsung's Accused Instrumentalities are able to include multiple namespaces (block devices), one of which being the addressed namespace. See: **1.5.37 namespace** A formatted quantity of non-volatile memory that may be directly accessed by a host. **1.5.38 namespace ID (NSID)** An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID. Source: NVMe Base Specification, at 22. A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace. Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
11b	the storage device is to identify, for the data being copied, an address of a logical erase unit, the logical erase unit being associated with the addressed block device to the exclusion of each other one	Samsung's Accused Instrumentalities comprise the storage device to identify, for the data being copied, an address of a logical erase unit, the logical erase unit being associated with the addressed block device to the exclusion of each other one of the block devices. Samsung's Accused Instrumentalities are able to identify a logical erase unit and its address for the data being copied. Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones, and they maintain a mapping of logical erase units to physical erase units to enable them to do this. See:

	of the block devices; and	 Source: Microsoft Denali Presentation, at 15.  Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.

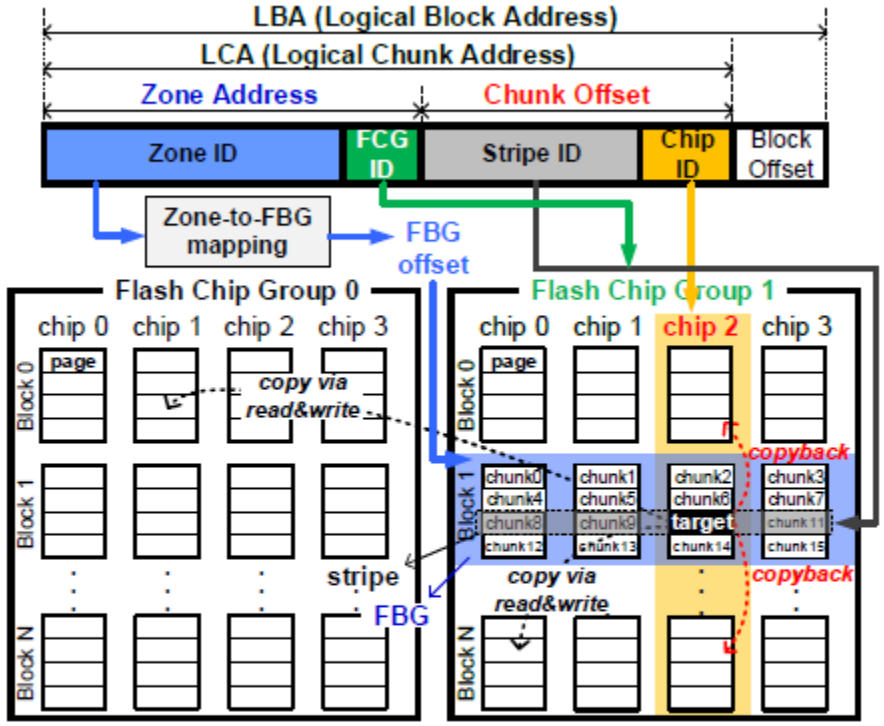


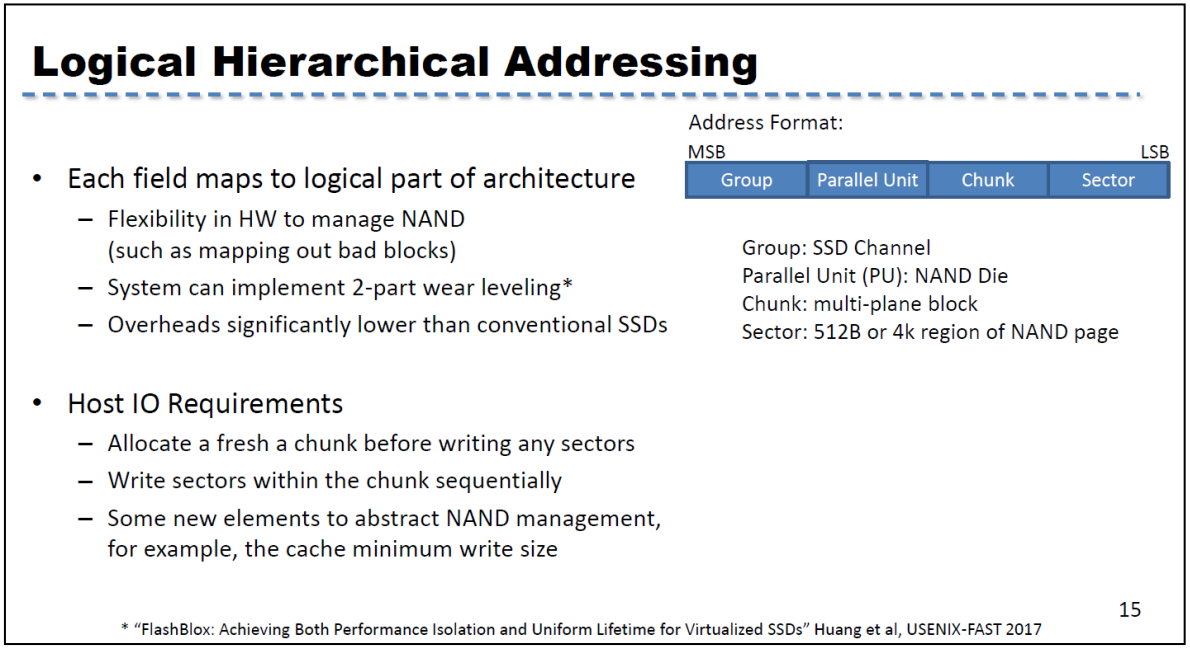
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

The logical erase unit is associated with only the addressed namespace (block device). See:

		1.5.37 namespace A formatted quantity of non-volatile memory that may be directly accessed by a host. **1.5.38 namespace ID (NSID)** An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID. Source: NVMe Base Specification, at 22. A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace. Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** | **Description** 07:04 | **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107. To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein because the accused functionality would be insubstantially different from or function in the same way to provide the same result as the relevant claim element. As an example, there is no substantial difference between (1) identifying a logical erase unit and (2) identifying a logical erase unit group and separately identifying an offset within the erase unit group. The two ways of identification perform substantially the same function in substantially the same way to achieve substantially the same result.
11c	the storage device is to remap a correspondence	Samsung's Accused Instrumentalities comprise the storage device to remap a correspondence between the address of the logical erase unit and the first physical erase unit to instead be between the address of the logical erase unit and the second physical erase unit.

	between the address of the logical erase unit and the first physical erase unit to instead be between the address of the logical erase unit and the second physical erase unit.	Samsung's Accused Instrumentalities are able to remap the correspondence between a logical erase unit and the first physical block (erase unit) in response to an internal operation to copy data from a first physical block to a second physical block to instead be between the logical erase unit and the second physical block. Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones, and they maintain a mapping of logical erase units to physical erase units to enable them to do this. See:  Source: Microsoft Denali Presentation, at 15.

		Bad Block Management & Error Correcting Code (ECC) In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance. Source: Samsung SSD White Paper, at 19. **2. Why is OP important?** OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.

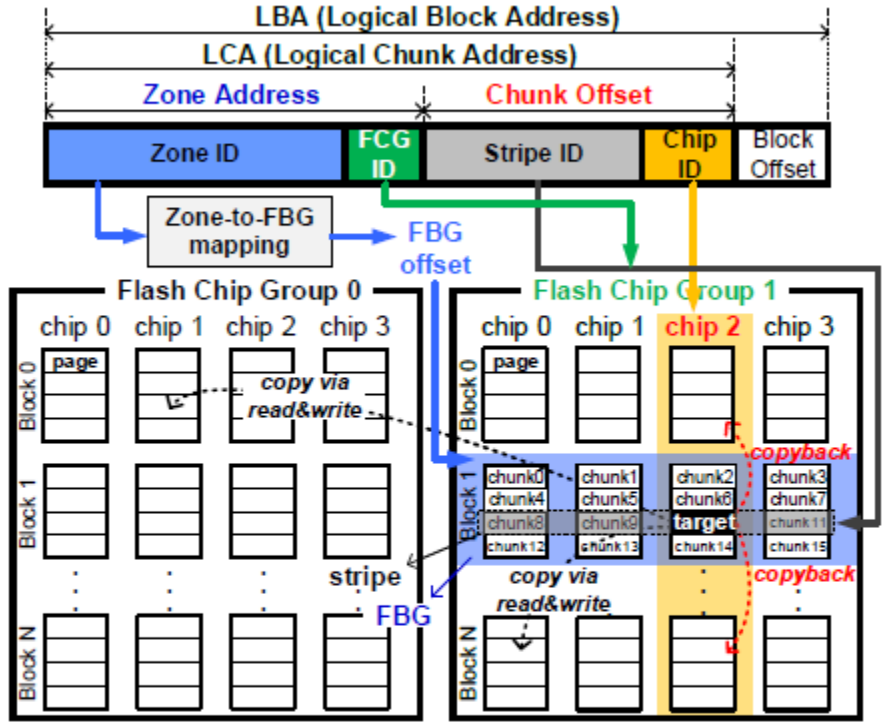
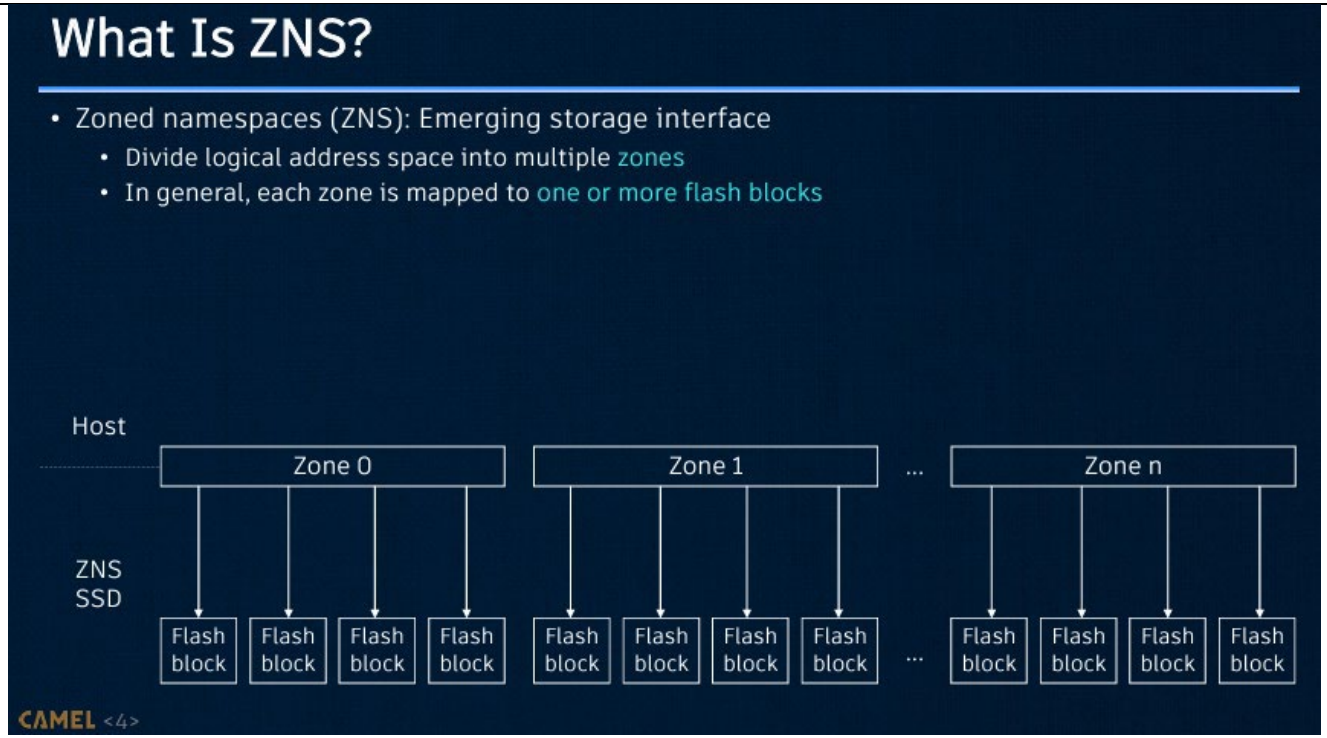


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein because the accused functionality would be insubstantially different from or function in the same way to provide the same result as the relevant claim element.

As an example, there is no substantial difference between (1) remapping the correspondence between a logical erase unit and a physical erase unit and (2) remapping the correspondence between a logical erase unit group and a physical erase unit group while maintaining an offset within the erase unit group. The two ways of remapping perform substantially the same function in substantially the same way to achieve substantially the same result.

12pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
12a	the flash memory is to have block devices, including the addressed block device; and	Samsung's Accused Instrumentalities comprise the flash memory which is to have block devices, including the addressed block device. Samsung's Accused Instrumentalities comprise the flash memory which is to have namespaces (block devices) including the addressed namespace. See: **1.5.37 namespace** A formatted quantity of non-volatile memory that may be directly accessed by a host. **1.5.38 namespace ID (NSID)** An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID. Source: NVMe Base Specification, at 22. A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace. Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
12b	the storage device is to store metadata on an independent basis for portions of flash memory associated with each one of the block devices, wherein the stored metadata is dependent on	Samsung's Accused Instrumentalities comprise the storage device to store metadata on an independent basis for portions of flash memory associated with each one of the block devices, wherein the stored metadata is dependent on an erase count of associated physical memory space. Samsung's Accused Instrumentalities are able to store metadata for each namespace. This metadata includes metadata dependent on a number of times the associated physical memory space has been erased, including metadata such as wear-leveling information or an erase count. See:

	an erase count of associated physical memory space;	After a sub-block erase operation, the erase manage module **43***a* may check an erase status of an erased sub-block and/or a sub-block adjacent to the erased sub-block. For example, the erase manage module **43***a* may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module **43***a* may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module **43***a* may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module **43***a* may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module **43***a* may read data of the erased sub-block to monitor a variation in threshold voltages of selected memory cells and/or a variation in the bit error rate (BER). The erase manage module **43***a* may also read data of an unselected sub-block to detect a variation in a threshold voltage. The memory controller **40** may perform various procedures for compensating for insufficient erasing of a selected sub-block based on erase status information detected by the erase manage module **43***a*. Source: Samsung's '250 Patent at 5:16–38.



Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.
12c	the storage device is to compare the metadata to at least one threshold;	Samsung's Accused Instrumentalities comprise the storage device to compare the metadata to at least one threshold. Samsung's Accused Instrumentalities are able to compare the metadata to at least one threshold. This metadata includes metadata dependent on a number of times the associated memory space has been erased, including metadata such as wear-leveling information or an erase count. Samsung's Accused Instrumentalities are capable of determining when the maintained metadata exceeds a threshold (e.g., the erase count meets or exceeds a erase count threshold). See:

		After a sub-block erase operation, the erase manage module **43***a* may check an erase status of an erased sub-block and/or a sub-block adjacent to the erased sub-block. For example, the erase manage module **43***a* may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module **43***a* may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module **43***a* may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module **43***a* may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module **43***a* may read data of the erased sub-block to monitor a variation in threshold voltages of selected memory cells and/or a variation in the bit error rate (BER). The erase manage module **43***a* may also read data of an unselected sub-block to detect a variation in a threshold voltage. The memory controller **40** may perform various procedures for compensating for insufficient erasing of a selected sub-block based on erase status information detected by the erase manage module **43***a*. Source: Samsung's '250 Patent at 5:16–38.	



Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.
12d	upon detection that specific metadata has satisfied the one or more thresholds, the circuitry is to copy data stored at one or more physical storage locations associated with the specific metadata from a first physical erase unit to a second physical erase unit; and	Samsung's Accused Instrumentalities comprise circuitry to, upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first physical erase unit to a second the physical erase unit. Samsung's Accused Instrumentalities are able to copy data from one physical storage location at one physical erase unit to another physical erase unit using internal operations. Samsung's Accused Instrumentalities are able to perform such internal operations upon detection that the erase count has satisfied the one or more thresholds. See: **5.4 Reset Zone Recommended** A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the ZSF:Full state may notify host software to initiate a zone reset operation (refer

to section 3.4.3.1.4). If a controller schedules such an internal operation on a zone, the controller may notify the host by:

a) setting the Reset Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 37);
b) setting the Reset Zone Recommended Time Limit zone attribute information to indicate the number of seconds before the controller intends to perform an internal operation on the specified zone; and
c) generating a Zone Descriptor Changed event for the specific zone.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:

a) clearing the Reset Zone Recommended zone attribute of the specific zone to '0'; and
b) generating a Zone Descriptor Changed event for the specific zone.

If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Reset Zone Recommended attribute shall be cleared to '0'.

Source: ZNS Specification, at 41–42.

		After a sub-block erase operation, the erase manage module **43***a* may check an erase status of an erased sub-block and/or a sub-block adjacent to the erased sub-block. For example, the erase manage module **43***a* may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module **43***a* may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module **43***a* may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module **43***a* may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module **43***a* may read data of the erased sub-block to monitor a variation in threshold voltages of selected memory cells and/or a variation in the bit error rate (BER). The erase manage module **43***a* may also read data of an unselected sub-block to detect a variation in a threshold voltage. The memory controller **40** may perform various procedures for compensating for insufficient erasing of a selected sub-block based on erase status information detected by the erase manage module **43***a*. Source: Samsung's '250 Patent at 5:16–38.



Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.
12e	the storage device is to remap a logical erase unit value associated with the third address portion from the first physical erase unit to the second physical erase unit.	Samsung's Accused Instrumentalities comprise the storage device to remap the logical erase unit value associated with the third address portion from the first physical erase unit to the second physical erase unit. Samsung's Accused Instrumentalities are able to remap the correspondence of the logical erase unit value so that it instead corresponds to the second physical block (erase unit) after the data is copied from the first physical block to the second physical block. Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones, and they maintain a mapping of logical erase units to physical erase units to enable them to do this. The logical erase unit value is associated with the third address portion. See: **5.4 Reset Zone Recommended** A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the ZSF:Full state may notify host software to initiate a zone reset operation (refer

to section 3.4.3.1.4). If a controller schedules such an internal operation on a zone, the controller may notify the host by:

a) setting the Reset Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 37);
b) setting the Reset Zone Recommended Time Limit zone attribute information to indicate the number of seconds before the controller intends to perform an internal operation on the specified zone; and
c) generating a Zone Descriptor Changed event for the specific zone.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:

a) clearing the Reset Zone Recommended zone attribute of the specific zone to '0'; and
b) generating a Zone Descriptor Changed event for the specific zone.

If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Reset Zone Recommended attribute shall be cleared to '0'.

Source: ZNS Specification, at 41–42.

Logical Hierarchical Addressing

- Each field maps to logical part of architecture
  - Flexibility in HW to manage NAND (such as mapping out bad blocks)
  - System can implement 2-part wear leveling*
  - Overheads significantly lower than conventional SSDs
- Host IO Requirements
  - Allocate a fresh a chunk before writing any sectors
  - Write sectors within the chunk sequentially
  - Some new elements to abstract NAND management, for example, the cache minimum write size

Address Format:



Group: SSD Channel
Parallel Unit (PU): NAND Die
Chunk: multi-plane block
Sector: 512B or 4k region of NAND page

15

* "FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs" Huang et al, USENIX-FAST 2017

Source: Microsoft Denali Presentation, at 15.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.

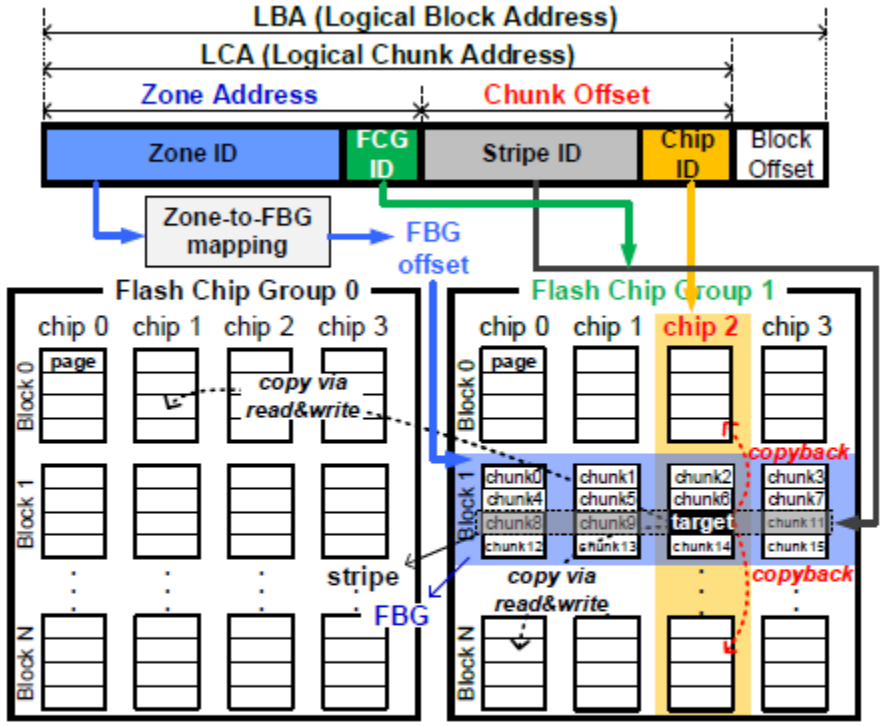


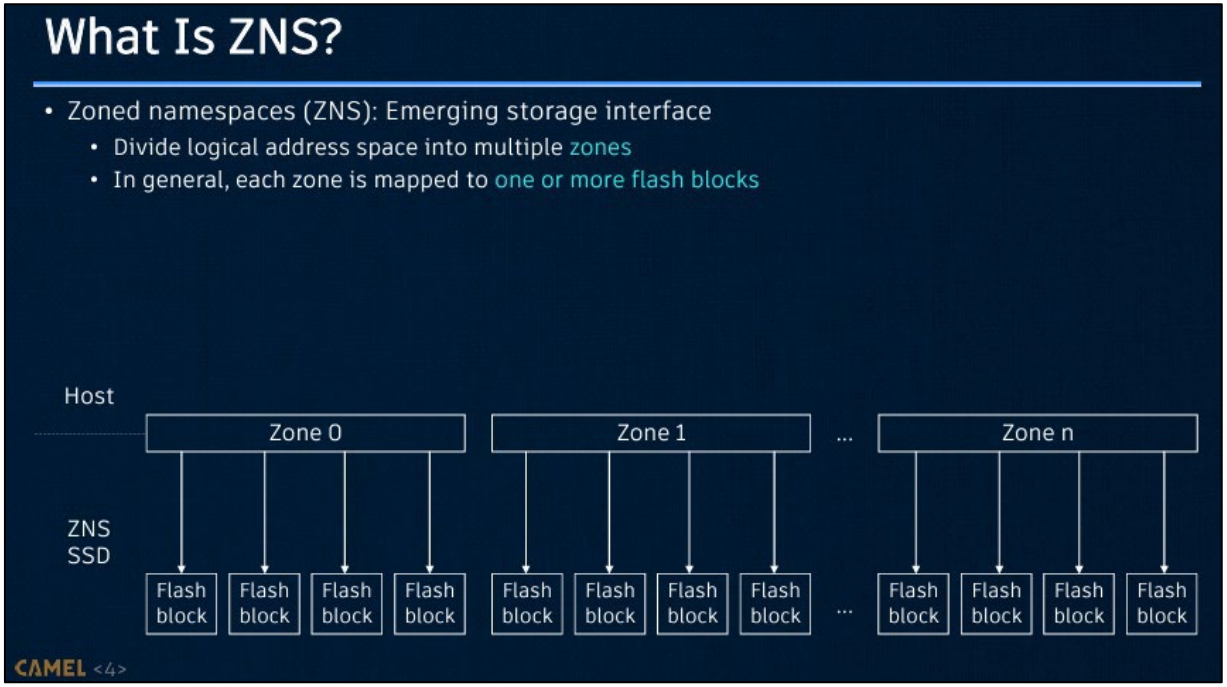
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

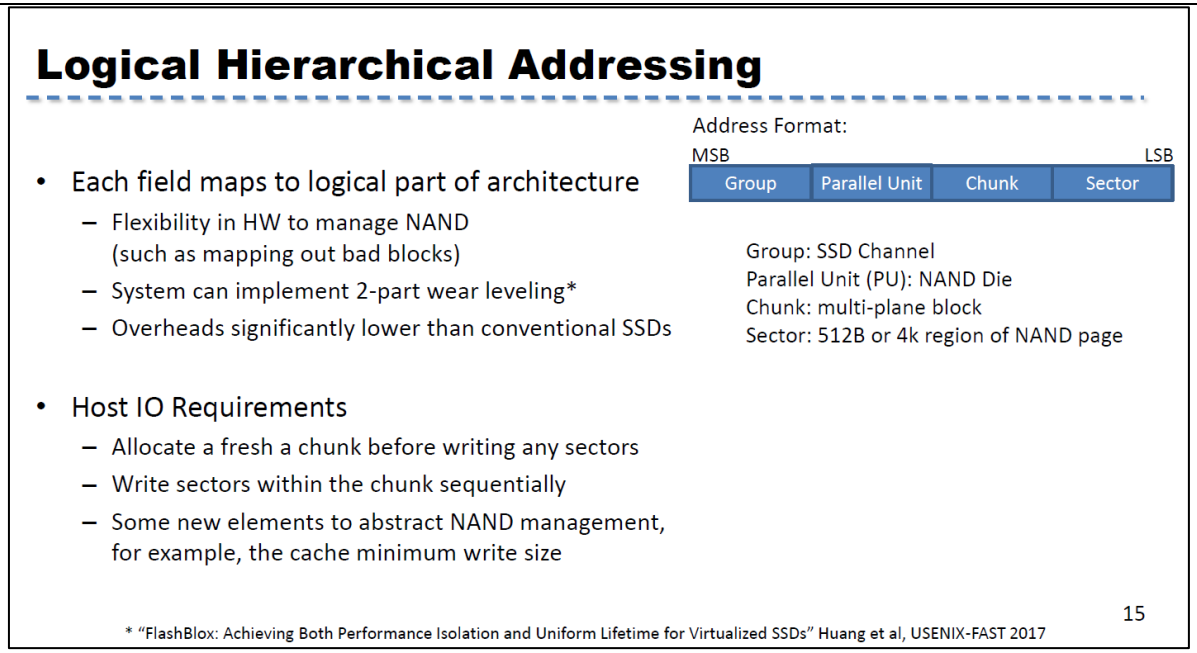
Source: Han, at 149.

To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein because the accused functionality would be insubstantially different from or function in the same way to provide the same result as the relevant claim element.

		As an example, there is no substantial difference between (1) remapping a logical erase unit corresponding to a physical erase unit and (2) remapping a logical erase unit group corresponding to a physical erase unit group while maintaining an offset within the erase unit group. The two ways of remapping perform substantially the same function in substantially the same way to achieve substantially the same result.
13pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
13a	the erase units of the flash memory are physical erase units;	Samsung's Accused Instrumentalities comprise erase units of the flash memory that are physical erase units. Samsung's Accused Instrumentalities comprise NAND flash memory that comprises physical blocks (erase units). See: **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17.

		Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP). Source: Samsung Over-Provisioning White Paper, at 2 (highlighted). – *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block. Source: Boukhobza, at 22 (highlighted). **2.2.1. *Erase-before-write constraint*** Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: *Id.* at 24.
13b	the circuitry is to use the second division operation to	Samsung's Accused Instrumentalities comprise the circuitry to use the second division operation to identify a logical erase unit, wherein each logical erase unit is mapped to one of the physical erase units. Samsung's Accused Instrumentalities are able to use the second division operation to identify a logical erase unit which is mapped to one of the physical blocks (erase units). Samsung's Accused

	identify a logical erase unit, wherein each logical erase unit is mapped to one of the physical erase units; and	Instrumentalities contain multiple equally sized erase units in each zone such that a division operation is used to identify a particular erase unit within a zone. See:  Source: Bae, at 4.



Source: Microsoft Denali Presentation, at 15.

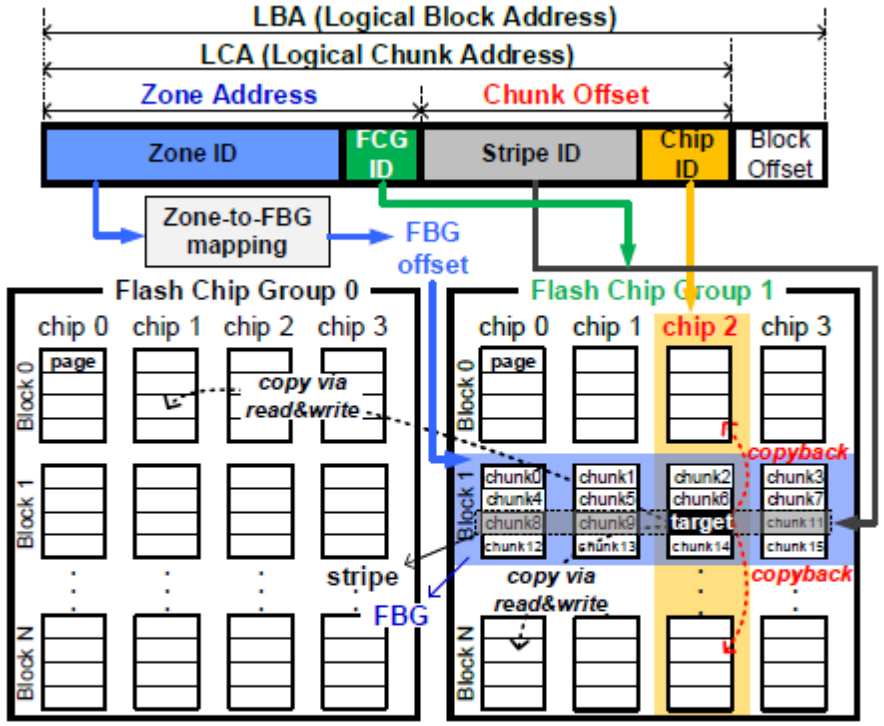


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

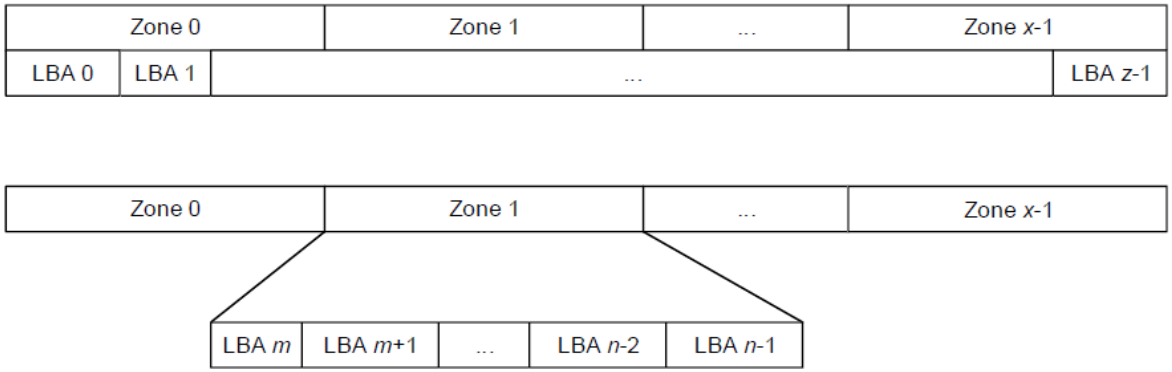
Source: Han, at 149.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



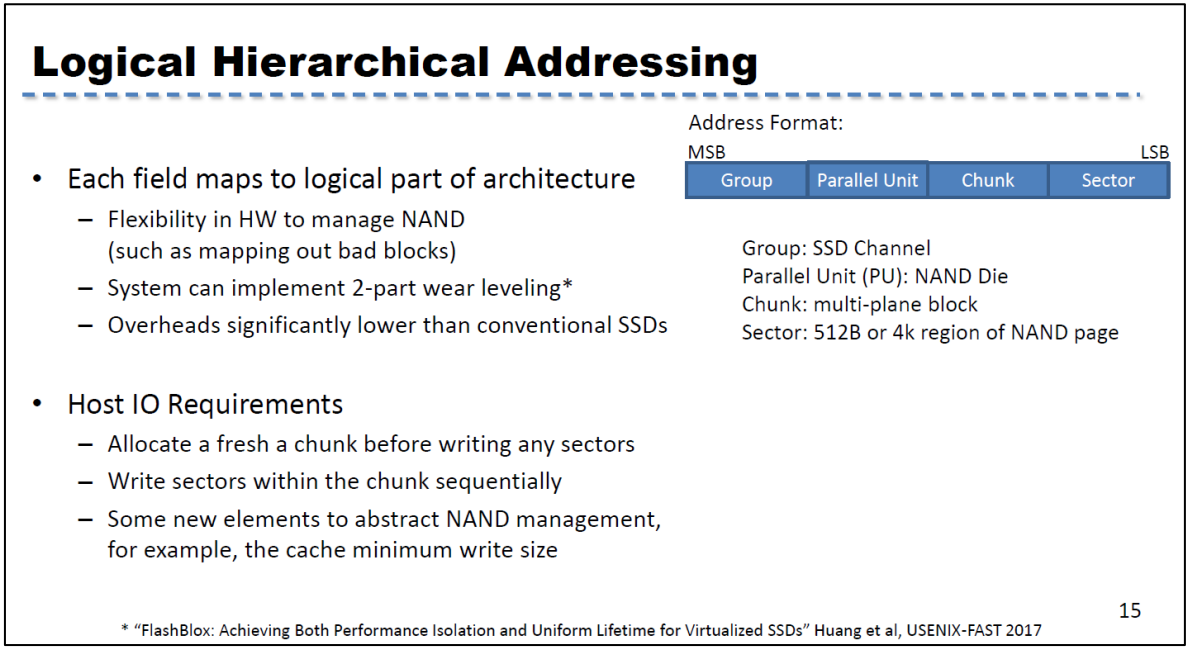
The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones, and they maintain a mapping of logical erase units to physical erase units to enable them to do this. See:

		Bad Block Management & Error Correcting Code (ECC) In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance. Source: Samsung SSD White Paper, at 19. **2. Why is OP important?** OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein because the accused functionality would be insubstantially different from or function in the same way to provide the same result as the relevant claim element. As an example, there is no substantial difference between (1) identifying a logical erase unit mapped to a physical erase unit and (2) identifying a logical erase unit group mapped to a physical erase unit group and separately identifying an offset within the erase unit group. The two ways of identification perform substantially the same function in substantially the same way to achieve substantially the same result.
13c	the storage device is to detect a failure	Samsung's Accused Instrumentalities comprise the storage device to detect a failure condition of the physical erase units and to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units

	condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests.	for which the failure condition has not been detected and to utilize the remapped correspondence to service one or more of the incoming data access requests. Samsung's Accused Instrumentalities are able to detect a failure condition of the physical blocks, retire the bad blocks, and substitute different blocks for the bad ones. Samsung's Accused Instrumentalities are able to remap the logical erase unit to physical erase unit correspondence accordingly and to utilize the remapped correspondence to service one or more of the incoming data access requests. See:  Source: Microsoft Denali Presentation, at 15.

		Bad Block Management & Error Correcting Code (ECC) In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance. Source: Samsung SSD White Paper, at 19. **2. Why is OP important?** OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2.

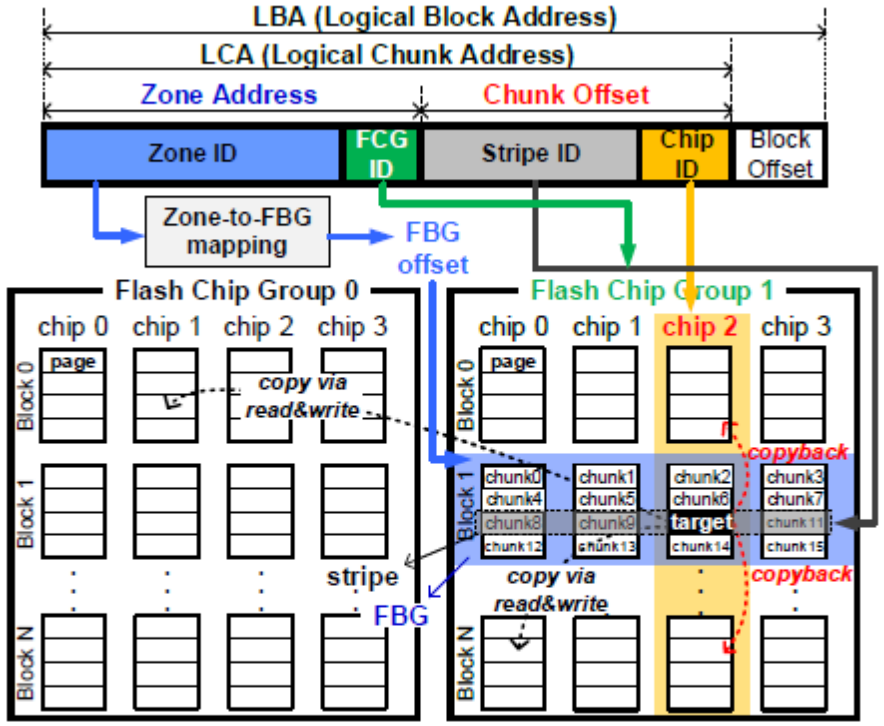
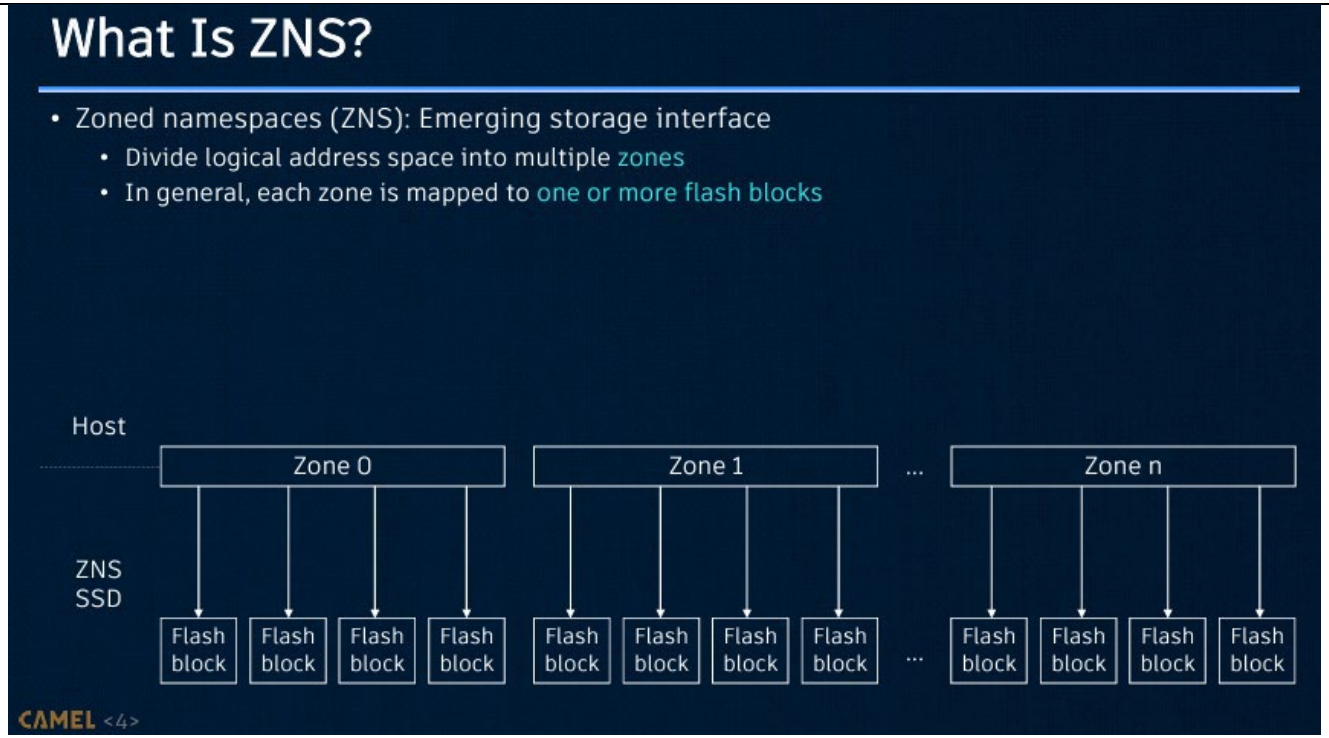


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein because the accused functionality would be insubstantially different from or function in the same way to provide the same result as the relevant claim element.

As an example, there is no substantial difference between (1) remapping a logical erase unit corresponding to a physical erase unit and (2) remapping a logical erase unit group corresponding to a physical erase unit group and maintaining an offset within the erase unit group. The two ways of

		identification perform substantially the same function in substantially the same way to achieve substantially the same result.
14pre	The storage device of claim 1 wherein the storage device is to:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
14a	store, for each segment having an erase unit which stores valid data, a value representing a time since data was programmed;	Samsung's Accused Instrumentalities comprise the storage device to store, for each segment having an erase unit which stores valid data, a value representing a time since data was programmed. Samsung's Accused Instrumentalities are able to store metadata for each zone having an erase unit which stores valid data. This metadata includes values indicating a time since data was programmed, including metadata such as age of stored data or a data retention period. See:

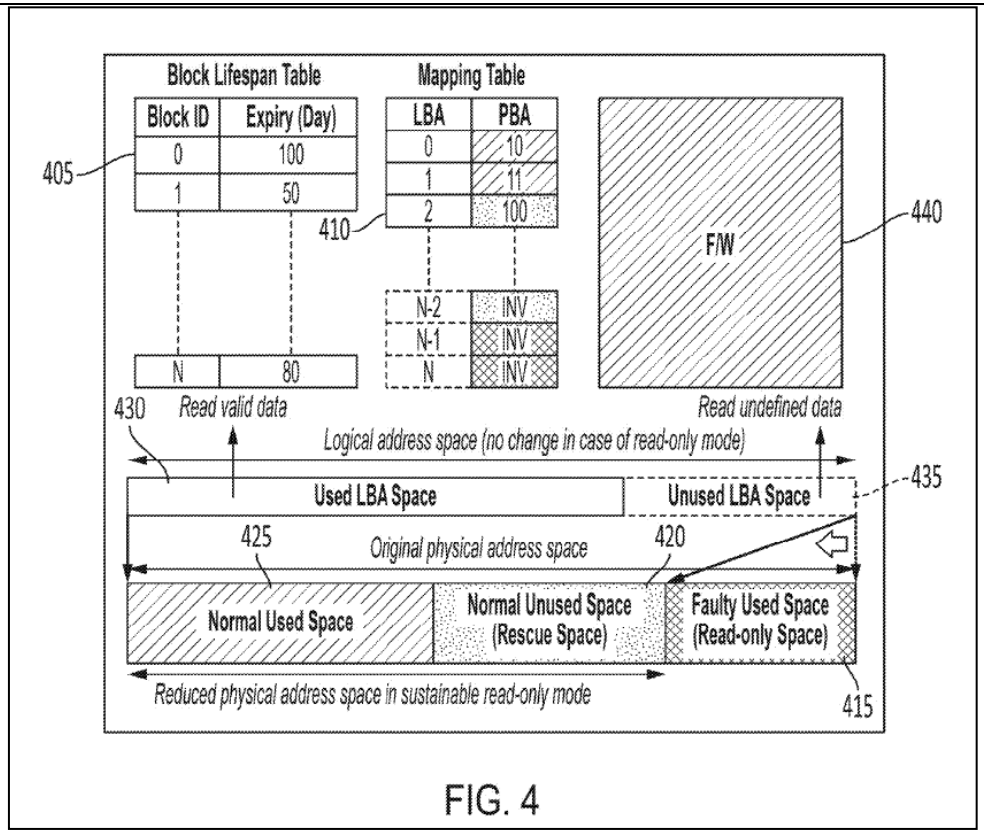

Block Lifespan Table
Block ID
Expiry (Day)
0
100
1
50
N
80
405
Mapping Table
LBA
PBA
0
10
1
11
2
100
410
N-2
INV
N-1
INV
N
INV
F/W
440
430
Read valid data
Read undefined data
Logical address space (no change in case of read-only mode)
435
Used LBA Space
Unused LBA Space
425
Original physical address space
420
Normal Used Space
Normal Unused Space
(Rescue Space)
Faulty Used Space
(Read-only Space)
Reduced physical address space in sustainable read-only mode
415

FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

		[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.

		[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].  Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

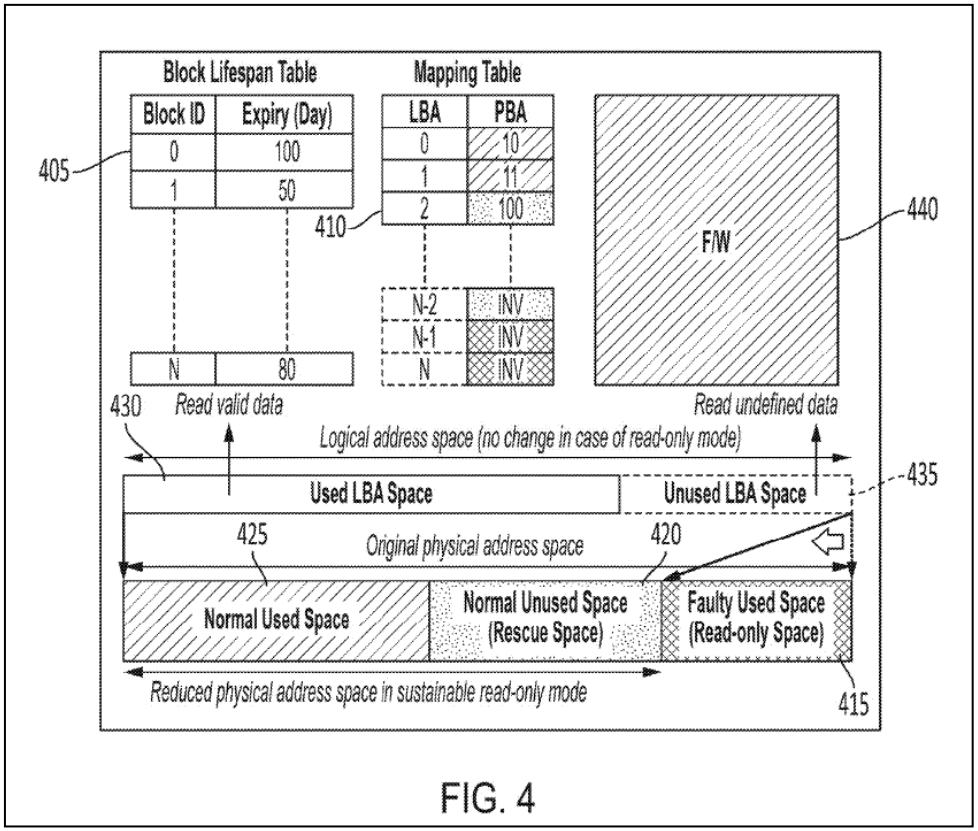
Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

		2.2.3. *Reliability limitation* A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09]. Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09]. Source: Boukhobza, at 26.
14b	compare the values representing time since data	Samsung's Accused Instrumentalities comprise the storage device to compare the values representing time since data was programmed to at least one threshold.

	was programmed to at least one threshold; and	Samsung's Accused Instrumentalities are able to maintain metadata representing a time since the data was programmed, including metadata such as age of stored data or a data retention period. Samsung's Accused Instrumentalities are capable of comparing the maintained metadata to at least one threshold (e.g., the age of data meets or exceeds a data retention period threshold). See:  FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.

		[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].  Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

		2.2.3. *Reliability limitation* A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09]. Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09]. Source: Boukhobza, at 26.
14c	responsive to detection of a condition that a specific one of	Samsung's Accused Instrumentalities comprise the storage device to, responsive to detection of a condition that a specific one of the values representing time since data was programmed has satisfied the one or more thresholds, transmit information to a host wherein such transmitted information conveys

	the values representing time since data was programmed has satisfied the one or more thresholds, transmit information to a host wherein such transmitted information conveys address information for data stored at one or more physical storage locations associated with the specific one of the values.	address information for data stored at one or more physical storage locations associated with the specific one of the values. Samsung's Accused Instrumentalities are able to, responsive to detection that the tracked metadata has satisfied the one or more thresholds, transmit to the host information conveying address information for the corresponding data. Samsung's Accused Instrumentalities are able to compare the metadata to at least one threshold and, responsive to detection that specific metadata has satisfied the at least one threshold, notifying the host by recommending that the host initiate a reset of that zone. The controller can schedule an internal operation related to background operations on the NAND flash, after detecting that metadata has met or exceeded a threshold, and notify the host by using Reset Zone Recommended for the zone, which indicates to the host that the controller recommends the host initiate a reset of that zone. ZNS Specification defines a "Reset Zone Recommended" field, which is information accessible to the host which indicates that certain condition was detected for a zone such that the memory device recommends the reset of that zone. See: **5.4 Reset Zone Recommended** A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the ZSF:Full state may notify host software to initiate a zone reset operation (refer

to section 3.4.3.1.4). If a controller schedules such an internal operation on a zone, the controller may notify the host by:

a) setting the Reset Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 37);
b) setting the Reset Zone Recommended Time Limit zone attribute information to indicate the number of seconds before the controller intends to perform an internal operation on the specified zone; and
c) generating a Zone Descriptor Changed event for the specific zone.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:

a) clearing the Reset Zone Recommended zone attribute of the specific zone to '0'; and
b) generating a Zone Descriptor Changed event for the specific zone.

If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Reset Zone Recommended attribute shall be cleared to '0'.

Source: ZNS Specification, at 41–42.

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.
6:3	Reserved
2	**Reset Zone Recommended (RZR):** If this bit is set to '1', then the controller recommends that this zone be reset. Refer to section 5.4.
1	**Finish Zone Recommended (FZR):** If this bit is set to '1', then the controller recommends that this zone be finished. Refer to section 5.5.
0	**Zone Finished by Controller (ZFC):** If this bit is set to '1', then the controller finished this zone due to a Zone Active Excursion. Refer to section 5.6.

Source: *Id.* at 28.

Samsung's Accused Instrumentalities are able to transmit to the host the ZSLBA (address information) corresponding to the zone with a Zone Descriptor, or, alternatively, with a Changed Zone List Log Page. See:

Figure 42: Asynchronous Event Information – Notice, Zoned Namespace Command Set

Value	Description
EFh	**Zone Descriptor Changed:** The Zone Descriptor data structure for a zone in a specific zoned namespace (refer to Figure 43) has changed. The Zone Descriptor of the zone that changed is indicated in the Changed Zone List log page. To clear this event, host software reads the Zone Changed List log page using the Get Log Page command with the Retain Asynchronous Event bit cleared to '0'. For a specific zone, a Zone Descriptor data structure change that occurred, for any of the following reasons, shall not generate a Zone Descriptor Changed event and shall not cause modifications to the Changed Zones List log page: a) a Zone Management Send command that specified that zone; b) a Zone Management Send command that specified all zones; c) a write operation that transitioned that zone: i. from the ZSE:Empty state to the ZSIO:Implicitly Opened state; ii. from the ZSIO:Implicitly Opened state to the ZSF:Full state; iii. from the ZSEO:Explicitly Opened state to the ZSF:Full state; or iv. from the ZSC:Closed state to the ZSIO:Implicitly Opened state; d) the controller transitioning that zone to the ZSF:Full state due to an NVM Subsystem Reset; and e) the controller transitioning that zone to the ZSC:Closed state (refer to section 2.1.1.4.1).

Source: *Id.* at 33.

3.4.2.2.3 Zone Descriptor Data Structure

Figure 37 defines the Zone Descriptor data structure.

Figure 37: Zone Descriptor Data Structure

23:16	**Zone Start Logical Block Address (ZSLBA):** This field contains the 64-bit address of the lowest logical block for the zone.

Source: *Id.* at 27–28.

Figure 46: Changed Zone List Log Page

Bytes	Description
	Header
01:00	**Number of Zone Identifiers:** This field indicates the number of Zone Identifiers entries in the list.
07:02	Reserved
	Zone Identifier List
15:08	**Zone Identifier 0:** This field contains the ZSLBA of the first zone in the list, if present.
23:16	**Zone Identifier 1:** This field contains the ZSLBA of the second zone in the list, if present.
…	…
(N*8+15):(N*8+8)	**Zone Identifier N:** This field contains the ZSLBA of the N+1 zone in the list, if present.

Source: *Id.* at 35.

This metadata includes values representing time since data was programmed, including metadata such as age of stored data or a data retention period. See:

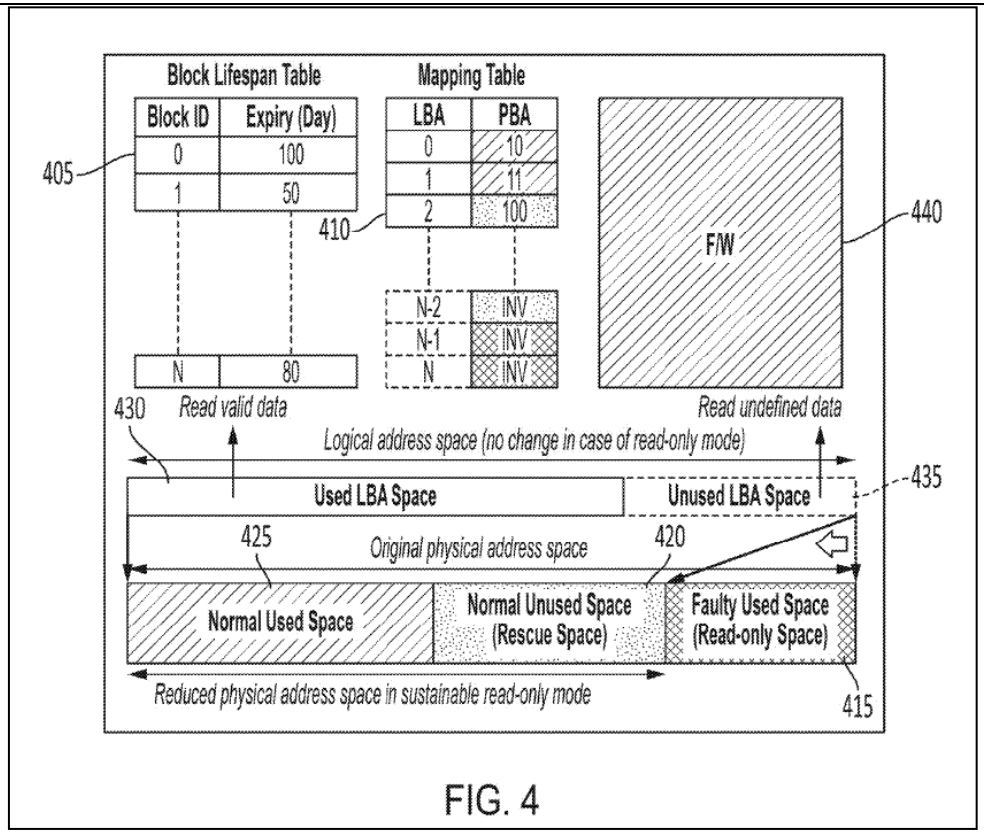

Block Lifespan Table
Block ID
Expiry (Day)
0
100
1
50
N
80
405
Mapping Table
LBA
PBA
0
10
1
11
2
100
410
N-2
INV
N-1
INV
N
INV
F/W
440
430
Read valid data
Read undefined data
Logical address space (no change in case of read-only mode)
Used LBA Space
Unused LBA Space
435
425
Original physical address space
420
Normal Used Space
Normal Unused Space
(Rescue Space)
Faulty Used Space
(Read-only Space)
Reduced physical address space in sustainable read-only mode
415

FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.

		[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].  Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

		2.2.3. *Reliability limitation* A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09]. Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09]. Source: Boukhobza, at 26.
15pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.

15a	the flash memory is to have block devices, including the addressed block device; and	Samsung's Accused Instrumentalities comprise the flash memory which is to have block devices, including the addressed block device. Samsung's Accused Instrumentalities are able to have namespaces (block devices) including the addressed namespace. See: **1.5.37 namespace** A formatted quantity of non-volatile memory that may be directly accessed by a host. **1.5.38 namespace ID (NSID)** An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID. Source: NVMe Base Specification, at 22. A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace. Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
15b	the storage device is further to:	Samsung's Accused Instrumentalities satisfy this claim element. Samsung's Accused Instrumentalities comprise the storage device which are able to perform what is described in claim elements [15c] to [15e]. See claim elements [15c] to [15e].
15c	store metadata on an independent basis for portions of flash memory associated with each one of the block devices,	Samsung's Accused Instrumentalities comprise the storage device which is to store metadata on an independent basis for portions of flash memory associated with each one of the block devices. Samsung's Accused Instrumentalities are able to store metadata on an independent basis for portions of flash memory associated with each namespace (block device). See:

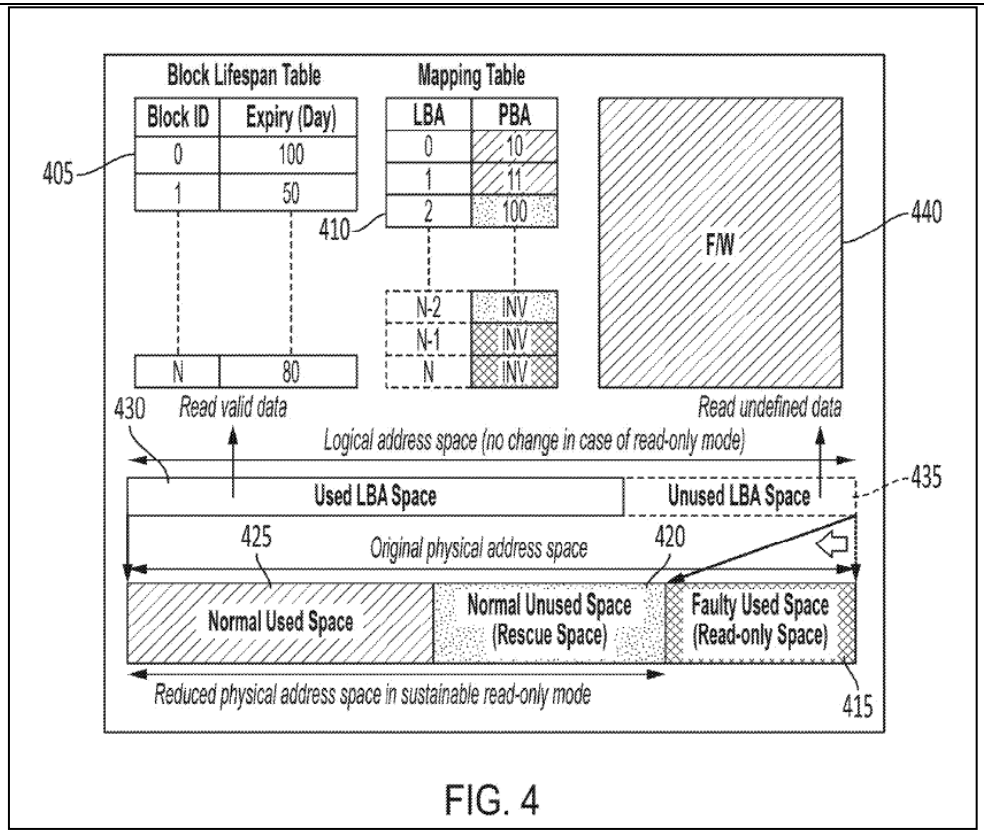

Block Lifespan Table
Block ID
Expiry (Day)
0
100
1
50
N
80
405
Mapping Table
LBA
PBA
0
10
1
11
2
100
410
N-2
INV
N-1
INV
N
INV
F/W
440
430
Read valid data
Read undefined data
Logical address space (no change in case of read-only mode)
435
Used LBA Space
Unused LBA Space
425
Original physical address space
420
Normal Used Space
Normal Unused Space
(Rescue Space)
Faulty Used Space
(Read-only Space)
Reduced physical address space in sustainable read-only mode
415

FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

		[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.	

[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of

Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].



Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

2.2.3. *Reliability limitation*

A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09].

Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09].

Source: Boukhobza, at 26.

		After a sub-block erase operation, the erase manage module **43***a* may check an erase status of an erased sub-block and/or a sub-block adjacent to the erased sub-block. For example, the erase manage module **43***a* may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module **43***a* may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module **43***a* may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module **43***a* may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module **43***a* may read data of the erased sub-block to monitor a variation in threshold voltages of selected memory cells and/or a variation in the bit error rate (BER). The erase manage module **43***a* may also read data of an unselected sub-block to detect a variation in a threshold voltage. The memory controller **40** may perform various procedures for compensating for insufficient erasing of a selected sub-block based on erase status information detected by the erase manage module **43***a*. Source: Samsung's '250 Patent at 5:16–38.
15d	receive a query, issued by a host, directed to a specific one of the block devices, and	Samsung's Accused Instrumentalities comprise the storage device to receive a query, issued by a host, directed to a specific one of the block devices. Samsung's Accused Instrumentalities are able to receive a query from a host that is directed to a specific namespace (block device). See:

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

1.5.37 namespace

A formatted quantity of non-volatile memory that may be directly accessed by a host.

1.5.38 namespace ID (NSID)

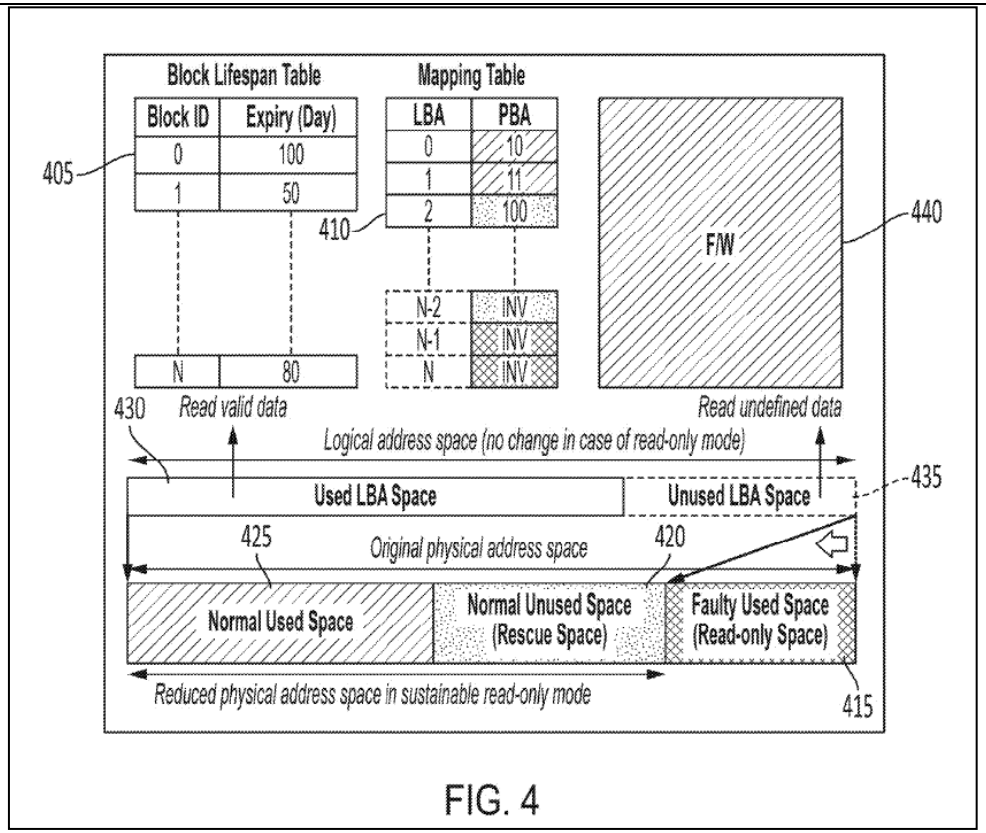
An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID.

Source: NVMe Base Specification, at 22.

A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace.

Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
15e	responsive to the query, compare the metadata to at least one threshold, to identify specific metadata, and transmit information to the host wherein such transmitted information	Samsung's Accused Instrumentalities comprise the storage device to, responsive to the query, compare the metadata to at least one threshold, to identify specific metadata, and transmit information to the host wherein such transmitted information conveys address information for data stored at one or more physical storage locations associated with the specific metadata. On information and belief, Samsung's Accused Instrumentalities are able to, in response to a query from the host, compare the metadata to a threshold and transmit address information to the host for data associated with identified specific metadata. See:

	conveys address information for data stored at one or more physical storage locations associated with the specific metadata.	 FIG. 4

		[0067] The storage media **125** may have a retention period (which may depend on the usage history of the storage media **125**, and, as such, may vary within the storage media **125**); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become unreliable and may be said to have expired. Data may be stored in the storage media **125** using an error correcting code, which may be a block code. When data is read from the storage media **125**, a quantity of raw data, referred to as a code block, may be read from the storage media **125**, and an attempt to decode it may be made. If the attempt fails, additional attempts (read retrials) may be made. With use, a portion, e.g., a block, of the storage media **125** may degrade to the point that the retention period becomes unacceptably short, and the block may be classified as a "bad block". To avoid allowing this circumstance to render the entire storage media **125** inoperable, reserve space, referred to as "bad block management reserve space" may be present (e.g., included in each flash memory die or in each flash memory plane), and the controller **120**, or another controller internal to the flash memory die or to the flash memory plane may begin to use a block in the reserve and cease to use the bad block.	

[0089] FIG. **4** is a schematic data layout diagram of a storage device **110** in accordance with example embodiments of the disclosure. A block lifespan table **405** lists the retention period of each block in the storage device **110**, and a mapping table **410** lists mappings from logical block address to physical block addresses. When the storage device **110** transitions to the sustainable read-only mode, the storage device **110** copies data from the faulty used space **415** (the physical space affected by the fault condition) to the rescue space **420**. Data in the normal used space **425** is handled as it would be if the storage device **110** were in normal operating mode (instead of being in the sustainable read-only mode). Because the fault condition results in a loss of physical space, there is a corresponding reduction in the size of the used LBA space **430**, with some of the previously available LBA space becoming unused LBA space **435**. Firmware **440**, executed by the controller **120**, may cause the data to be moved, as described above, and the tables to be updated.

[0090] FIG. **5**A is a flow chart for a method of transitioning to the sustainable read-only mode, in accordance with example embodiments of the disclosure. At **505**, the host **105** sends/receives data from storage device; at **510**, the host **105** determines if an error has occurred in the storage device; at **520**, the host **105** determines if an error happens with the storage device; at **525**, the storage device **110** performs an internal diagnosis and transitions to the sustainable read-only mode; at **530**, an FPGA may optionally monitor the nature of valid data on the storage device **110** (e.g., based on data type, or age); at **535**, the storage device **110** relocates valid data to the rescue space when the age of the data approaches the retention period; and at **540**, the storage device **110** may optionally transmit information about the status and contents of the rescue space to the host **105**, which flushes the data. FIG. **5**B is a flow chart showing details of

Source: Samsung's '781 Patent Application at Fig. 4, [0067], [0089]–[0090].



Source: Samsung Over-Provisioning White Paper, at 1.

ID	Attribute name	Status Flag	Threshold (%)
5	Reallocated Sector Count	110011	10
9	Power-on Hours	110010	-
12	Power-on Count	110010	-
177	Wear Leveling Count	010011	5
179	Used Reserved Block Count (total)	010011	10
180	Unused Reserved Block Count (total)	010011	10
181	Program Fail Count (total)	110010	10
182	Erase Fail Count (total)	110010	10
183	Runtime Bad Count (total)	010011	10
184	End to End Error data path Error count	110011	97
187	Uncorrectable Error Count	110010	-
190	Airflow Temperature	110010	-

194	Temperature	100010	-
195	ECC Error Rate	011010	-
197	Pending Sector Count	110010	-
199	CRC Error Count	111110	-
202	SSD Mode Status	110011	10
235	POR Recovery Count	010010	-
241	Total LBAs Written	110010	-
242	Total LBAs Read	110010	-
243	SATA Downshift Control	110010	-
244	Thermal Throttle Status	110010	-
245	Timed Workload Media Wear	110010	-
246	Timed Workload Host Read / Write Ratio	110010	-
247	Timed Workload Timer	110010	-
251	NAND Writes	110010	-

Source: *Id.* at 6.

Bad Block Management & Error Correcting Code (ECC)

In addition to maintenance at the drive level, the SSD must also perform maintenance at the chip level. In every NAND cell, each page contains a few extra bytes of extra capacity that the SSD controller uses to store a "parity bit." Error-Correcting Code (ECC) uses this parity bit to compensate for other bits that may fail during normal operation of the drive. When the controller detects a read failure, it will invoke ECC to try and recover from it. If recovery is not possible, the firmware's bad block management feature will retire the block and replace it with one of several free "reserved blocks." "Bad blocks" can be made during read, program, or erase operations and are actively managed to guarantee expected SSD performance.

Wear Leveling

NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) and on the program mechanism (e.g. SLC, MLC). In order to overcome this limitation, the SSD firmware employs a wear-leveling algorithm that guarantees that write operations are spread evenly among all NAND cells. Using this technique, no single cell should be unduly stressed and prematurely fail. If too many cells were to fail, the entire block would have to be retired as just discussed above. There are only a limited number of reserved blocks, however, so this event should be avoided to prolong overall drive life.

Wrap Up

Fortunately, all of the above procedures (with the exception of TRIM if you're using an older Windows OS) happen transparently and without action on behalf of the user. While specific implantation will vary, most modern SSDs include all of these features. In fact, without features like Wear Leveling and ECC (to extend drive life and protect data integrity) and TRIM and Garbage Collection (to maintain SSD performance), SSD quality and user experience would suffer.

Why Samsung?

Maintenance procedures like wear-leveling and Garbage collection, which are created to overcome the unique properties of NAND flash memory, work together to help ensure that your SSD performs well over extended use. Together, these algorithms actually increase write activities to the NAND, which reduces overall lifespan. Thus, the key in designing a great SSD is finding the optimum balance among lifespan, performance, and reliability. As the #1 player in the memory business for over 20 years and the largest global supplier of SSDs in the preinstalled storage business, Samsung has unrivaled knowledge of and experience with SSD technology. Samsung's unique, integrated approach to SSD manufacturing affords it full control of every component. You can trust that Samsung's expertise is safeguarding your precious data, and your productivity, when you purchase a Samsung SSD.

Source: Samsung SSD White Paper, at 19.

		2. Why is OP important? OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process that requires additional time as data is copied, erased and recopied. An added benefit is that OP makes all of the SSD maintenance procedures more efficient, reducing the Write Amplification Factor (WAF) by ensuring there's room to work, which improves the SSD's lifetime. Source: Samsung Application Note, at 2. **Reliability.** Endurance, in drive writes per day (DWPD), goes to 0.26 for the new device from 0.18 for the previous generation. Power-off data retention is increased from 1 month to 3 months. Source: Samsung Tech Blog.

2.2.3. *Reliability limitation*

A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (*bitflips*) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages applied to the cells during the write and erase operations [RIC 14b, CHE 07]. Ultimately, this lack of reliability means that without a specific management the data written in the flash memory and that read some time afterwards will not necessarily be read the same. Moreover, in order to reduce the errors due to reliability problems of a NAND flash, during a write operation on the flash it is usually advisable to program the pages within a same block in a sequential way. This is a strong constraint for MLC chips, and a (strong) suggestion for SLC chips [GRU 09].

Data retention is a metric that indicates the time during which a cell is capable of memorizing its content after it was programmed. The retention time of flash memory cells is reported to be between 5 and 10 years [CHE 07]. These numbers should be weighted carefully, as the retention capability of a flash memory cell depends on several parameters, in particular on the number of erase operations performed during the programming. The numbers above suppose a number of erase operations equal to zero. The retention can drop to 1 year for chips at the end of their life [GRU 09].

Source: Boukhobza, at 26.

		After a sub-block erase operation, the erase manage module **43***a* may check an erase status of an erased sub-block and/or a sub-block adjacent to the erased sub-block. For example, the erase manage module **43***a* may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module **43***a* may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module **43***a* may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module **43***a* may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module **43***a* may read data of the erased sub-block to monitor a variation in threshold voltages of selected memory cells and/or a variation in the bit error rate (BER). The erase manage module **43***a* may also read data of an unselected sub-block to detect a variation in a threshold voltage. The memory controller **40** may perform various procedures for compensating for insufficient erasing of a selected sub-block based on erase status information detected by the erase manage module **43***a*. Source: Samsung's '250 Patent at 5:16–38.
18pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
18a	the flash memory comprises	Samsung's Accused Instrumentalities comprise the flash memory comprising multiple planes. Samsung's Accused Instrumentalities comprise a NAND flash memory which comprises multiple planes. See:

	multiple planes; and	**Memory density and die floor plan** Figure 1 shows 16 48L V-NAND dice with two F-Chips in an MCP (multichip package). Die efficiency is higher in the 48L. The 32L V-NAND die area is 84.3 mm^2 while the 48L V-NAND die measures 99.8 mm^2, which is a 17.3% increase due to greater die length (Figure 2). Memory density per unit die area is increased to 2.57 Gb/mm^2. The memory density of leading-edge 2D planar NAND devices such as Toshiba 15 nm TLC NAND is 1.28 Gb/mm^2, (see more on our analysis on the Toshiba 15 nm NAND). The key feature differences in die floor plans are 1) planar (NAND memory array) area, 2) bitline switch and page buffer area, 3) LOGIC and peripheral area, and 4) the addition of F-chip. Each die has two planes. The NAND memory array area increased from 48.9 mm^2 to 68.7 mm^2 which is 40.3% larger. The area of the bitline switch circuit is the same as that of the 32L, while the page buffer area is 20% smaller. The logic and peripheral circuit area is 34.8% reduced. In other words, Samsung dramatically shrank the area of the page buffer and peripheral region so that they can further increase memory density and die efficiency. Die thickness is also reduced from 132 μm to 36 μm for the sixteen die stacked in the MCP. Source: EETimes V-NAND Analysis (highlighted).

Samsung TLC V- NAND Flash Memory		
	9th Gen V-NAND	8th Gen V-NAND
Layers	290?	236
Decks	2 (x145)	2 (x118)
Die Capacity	1 Tbit	1 Tbit
Die Size (mm^2)	?mm2	?mm2
Density ($Gbit/mm^2$)	?	?
I/O Speed	3.2 GT/s (Toggle 5.1)	2.4 GT/s (Toggle 5.0)
Planes	6?	4
CuA / PuC	Yes	Yes

Source: Anand V-NAND Analysis (annotated).

		To maximize the flash operation parallelism, the flash blocks from a set of flash chips accessible in parallel will compose the FBG of a zone, and the chunks of a zone need to be interleavingly placed on the parallel flash chips. The number of parallel flash chips for the chunk interleaving is referred to as the *zone interleaving degree* D_{zone}, and the set of logically consecutive chunks across the parallel flash chips is referred to as a **stripe**. For a coarse-grained zone-to-FBG mapping, an FBG has flash blocks at the same block offset in the parallel flash chips, and the chunks of a stripe are located at the same page offset in different flash blocks. Source: Han, at 149.
18b	the circuitry is to, for each one of a set of the data write requests having sequential addressing, identify a commonly addressed page and concurrently write data accompanying	Samsung's Accused Instrumentalities comprise the circuitry to, for each one of a set of the data write requests having sequential addressing, identify a commonly addressed page and concurrently write data accompanying the set of write requests into respective ones of the multiple planes in a manner corresponding to the commonly addressed page. Samsung's Accused Instrumentalities are able to receive and execute a Write command which is a data write request having sequential addressing. See: **3.3.7 Write command** The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2). Source: ZNS Specification, at 20.

	the set of write requests into respective ones of the multiple planes in a manner corresponding to the commonly addressed page.	**2.1.1.2 Zone Types** **2.1.1.2.1 Sequential Write Required Zones** A zone type of Sequential Write Required requires the set of logical block addresses of a zone to be written sequentially. **2.1.1.2.1.1 Writing in Sequential Write Required Zones** The following commands may be used to write to logical blocks in a specific zone of zone type Sequential Write Required: • Write command; • Write Zeroes command; • Write Uncorrectable command; • Copy command; and • Zone Append command. A write pointer is maintained for each zone in the zoned namespace that indicates the next writeable logical block address in that zone. The write pointer is valid for a subset of the zone states as defined in Figure 4. Source: ZNS Specification, at 9. Samsung's Accused Instrumentalities are able to identify a commonly addressed page and write data accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes in a manner corresponding to the commonly addressed page. See:

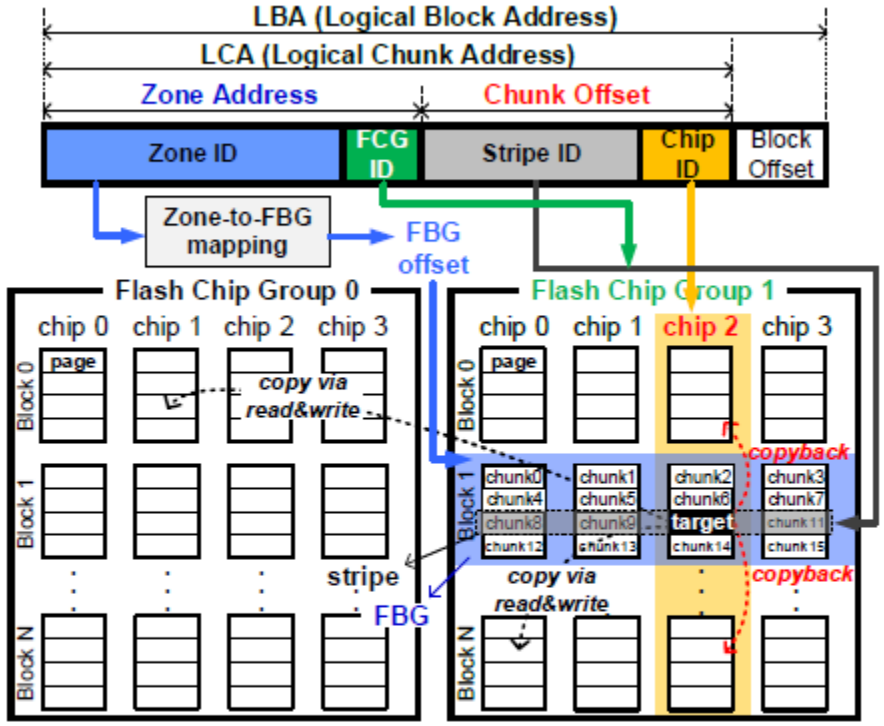
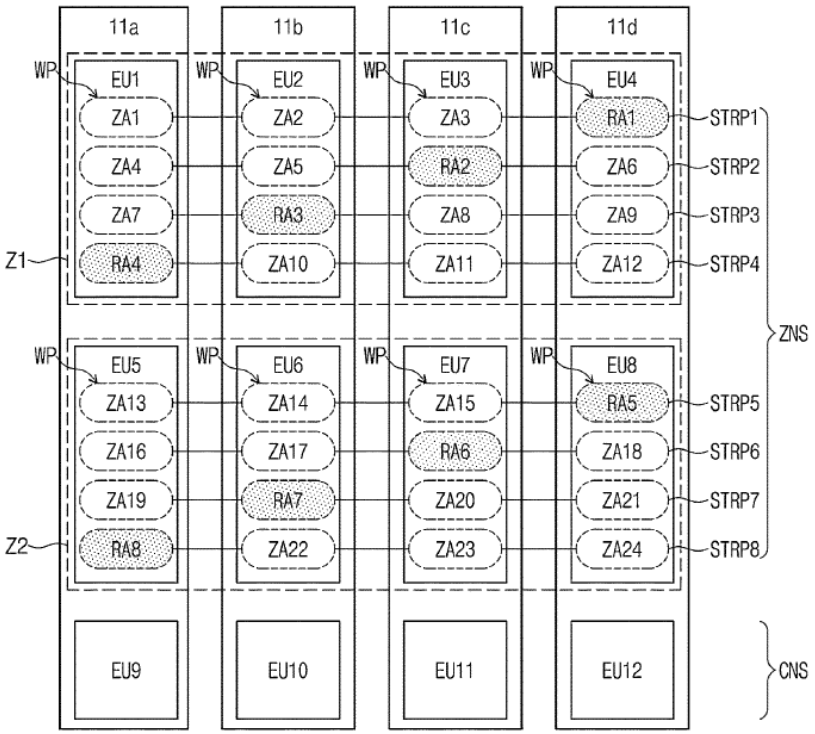


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

		To maximize the flash operation parallelism, the flash blocks from a set of flash chips accessible in parallel will compose the FBG of a zone, and the chunks of a zone need to be interleavingly placed on the parallel flash chips. The number of parallel flash chips for the chunk interleaving is referred to as the *zone interleaving degree* D_{zone}, and the set of logically consecutive chunks across the parallel flash chips is referred to as a **stripe**. For a coarse-grained zone-to-FBG mapping, an FBG has flash blocks at the same block offset in the parallel flash chips, and the chunks of a stripe are located at the same page offset in different flash blocks. Source: Han, at 149.

FIG. 7


11a
11b
11c
11d
WP
EU1
EU2
EU3
EU4
ZA1
ZA2
ZA3
RA1
ZA4
ZA5
RA2
ZA6
ZA7
RA3
ZA8
ZA9
Z1
RA4
ZA10
ZA11
ZA12
STRP1
STRP2
STRP3
STRP4
ZNS
EU5
EU6
EU7
EU8
ZA13
ZA14
ZA15
RA5
ZA16
ZA17
RA6
ZA18
ZA19
RA7
ZA20
ZA21
Z2
RA8
ZA22
ZA23
ZA24
STRP5
STRP6
STRP7
STRP8
EU9
EU10
EU11
EU12
CNS

		[0027] Each of the plurality of storage devices **11***a*, **11***b*, **11***c*, and **11***d* may include a nonvolatile memory device configured to retain data stored therein even when a power is turned off. For example, each of the plurality of storage devices **11***a*, **11***b*, **11***c*, and **11***d* may be (or include) a solid state drive. Though the electronic device **10** is illustrated as including four (4) storage devices, the plurality of storage devices is not limited thereto, and may include, e.g., more storage devices than illustrated. Each of the plurality of storage devices **11***a*, **11***b*, **11***c*, and **11***d* may include a first storage area ZNS and a second storage area CNS. **[0028]** The first storage area ZNS may be used as a zoned namespace. Each of the plurality of storage devices **11***a*, **11***b*, **11***c*, and **11***d* may be configured to permit sequential writes (or only sequential writes), which are based on sequential logical addresses, in the first storage area ZNS. A portion of the first storage area ZNS, for example, a portion of a storage space may be used to implement Redundant Array of Inexpensive Disks (RAID). For example, the storage space may be used based on a RAID (e.g., RAID**5** and/or RAID **4**) standard.	

[0113] FIG. **7** illustrates an example in which a first zone Z**1** and a second zone Z**2** are generated in the storage devices **11***a*, **11***b*, **11***c*, and **11***d*. Referring to FIGS. **1**, **2**, and **7**, in at least one embodiment, one zone may correspond to four erase units.

[0114] The storage devices **11***a*, **11***b*, **11***c*, and **11***d* may assign the first to fourth erase units EU**1** to EU**4** respectively included in the storage devices **11***a*, **11***b*, **11***c*, and **11***d* to the first zone Z**1**. Continuous logical addresses may be assigned to the first to 12th zone areas ZA**1** to ZA**12** that are included in the first to fourth RAID stripes STRP**1** to STRP**4** defined in the first to fourth erase units EU**1** to EU**4**.

[0115] Continuous logical addresses may be assigned to each of the first to fourth RAID areas RA**1** to RA**4** that are included in the first to fourth RAID stripes STRP**1** to STRP**4** defined in the first to fourth erase units EU**1** to EU**4**. Logical addresses of one RAID area may be continuous or discontinuous to logical addresses of any other RAID area and may be continuous or discontinuous to logical addresses of a zone area.

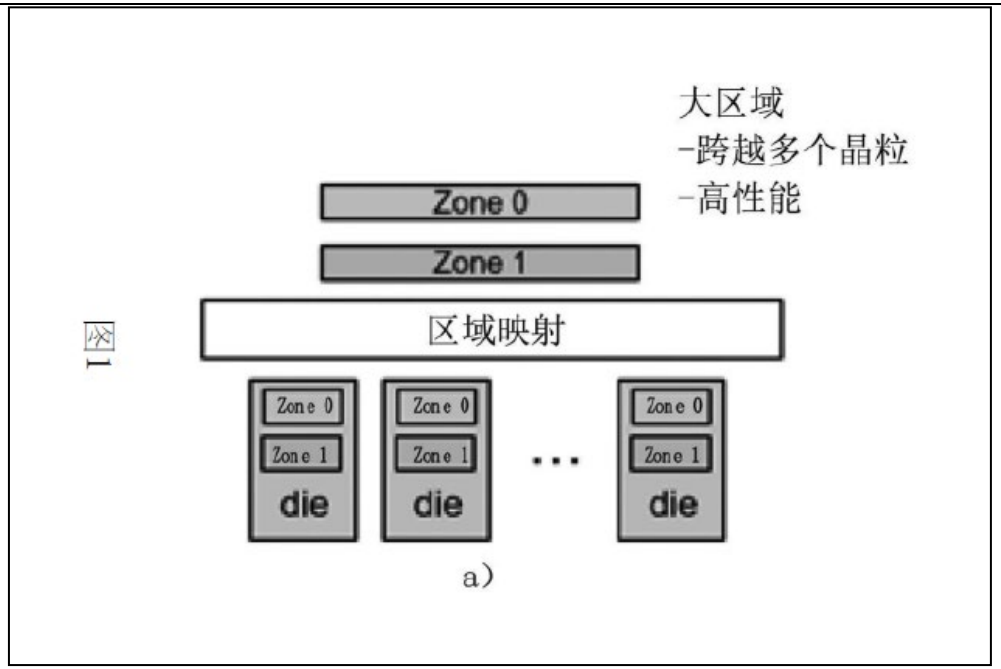
[0116] A write pointer WP of the first storage device **11***a* may point out the first (or LSB) logical address among logical addresses where data are not written, for example, a logical address of the first zone area ZA**1**. Data may be written in a storage space of the first storage device **11***a*, which is pointed out by the write pointer WP. When the data is written in the storage space pointed out by the write pointer WP, the first storage device **11***a* may update the write pointer WP to point out the first (or least significant bit (LSB)) logical address belonging to the first storage device **11***a* from among the logical addresses of the first zone Z**1**.

[0117] Likewise, the write pointer WP of the second storage device **11***b* may point out the second zone area ZA**2**, and the write pointer WP of the third storage device **11***c* may point out the third zone area ZA**3**.

[0118] In each RAID stripe, when logical addresses of the RAID area are continuous to logical addresses of zone areas, the RAID area may also be pointed out by the write pointer WP. For example, when the logical addresses of the first RAID area RA**1** are continuous to the logical addresses of the third zone area ZA**3**, the write pointer WP of the fourth storage device **11***d* may point out the first RAID area RA**1**.

[0119] In each RAID stripe, when logical addresses of the RAID area are managed to be independent of (or are not continuous to) logical addresses of zone areas, the RAID area may not be pointed out by the write pointer WP. For example, the write pointer WP of the fourth storage device

		11*d* may skip the first RAID area RA**1** and may point out the sixth zone area ZA**6** (or may jump to the sixth zone area ZA**6**). **[0120]** The storage devices **11***a*, **11***b*, **11***c*, and **11***d* may assign the fifth to eighth erase units EU**5** to EU**8** respectively included in the storage devices **11***a*, **11***b*, **11***c*, and **11***d* to the second zone Z**2**. Continuous logical addresses may be assigned to the 13th to 24th zone areas ZA**13** to ZA**24** that are included in the fifth to eighth RAID stripes STRP**5** to STRP**8** defined in the fifth to eighth erase units EU**5** to EU**8**. **[0121]** Continuous logical addresses may be assigned to each of the fifth to eighth RAID areas RA**5** to RA**8** that are included in the fifth to eighth RAID stripes STRP**5** to STRP**8** defined in the fifth to eighth erase units EU**5** to EU**8**. Logical addresses of one RAID area may be continuous or discontinuous to logical addresses of any other RAID area and may be continuous or discontinuous to logical addresses of a zone area. Source: Samsung's '890 Patent Application at Fig. 7, [0027]–[0028], [0113]–[0121].



Translation:

Left side of figure: "Figure 1"

White box under Zone 1 box: "Zone Mapping"

Top right caption:

"Large Zone

- Spanning multiple dies
- High performance"

[0003] 当ZNS SSD的区域的尺寸较小时,区域跨越较少的晶粒(die),因此,针对区域的数据传输带宽较小;当ZNS SSD的区域的尺寸较大时,区域跨越较多的晶粒(die),因此,可以提高针对区域的数据传输带宽。当区域的尺寸较大时,该区域可能存储具有不同寿命的文件,因此该区域存储的文件可能无法被同时无效,从而导致该区域无法被及时重置回收,在这种情况下,ZNS SSD的存储空间利用率低。

Translation:

"[0003] When the size of a ZNS SSD zone is small, the zone spans fewer dies, so the data transmission bandwidth for the zone is small; when the size of a ZNS SSD zone is large, the zone spans more dies, so the data transmission bandwidth for the zone can be improved. When the size of a zone is large, the zone may store files with different lifespans, so the files stored in the zone may not be invalidated at the same time, resulting in the zone being unable to be reset and recycled in time. In this case, the storage space utilization rate of the ZNS SSD is low."

[0042] 图1示出了针对ZNS SSD的区域(Zone)的数据传输带宽与区域的尺寸的关系的示意图。

[0043] 参照图1中的a),当ZNS SSD的区域的尺寸较大时,其可以跨越多个晶粒,这样能够保证较高的数据传输带宽。参照图1中的b),当区域的尺寸较小时,其仅跨越单个晶粒,这样针对区域的数据传输带宽小,从而降低数据传输性能。

Translation:

"[0042] Fig. 1 shows a schematic diagram of the relationship between the data transmission bandwidth of a zone and the size of the zone for a ZNS SSD.

[0043] Referring to a) in Fig. 1, when the size of a ZNS SSD zone is large, it can span multiple dies, which can ensure a higher data transmission bandwidth.

		Referring to b) in Fig. 1, when the size of a zone is small, it only spans a single die, which reduces the data transmission bandwidth for the zone and reduces the data transmission performance." Source: Samsung's '816 Patent Application at Fig. 1a, [0003], [0042]–[0043].
19pre	A storage device comprising:	To the extent the preamble is considered a limitation, Samsung's Accused Instrumentalities contain a storage device.
19a	flash memory comprising erase units; and	Samsung's Accused Instrumentalities contain a flash memory comprising erase units. Samsung's Accused Instrumentalities contain NAND flash memory. For example, Samsung's PM1733 contains NAND flash memory. See: **Samsung PM1733 specifications** Form factor: U.2 / 2.5" Capacity: 1.92 TB, 3.84 TB, 7.68TB, and 15.36TB Host interface: PCIe Gen 3/4 x4 Spec Compliance: NVMe spec rev. 1.3; PCI Express base specification rev. 4.0 NAND flash memory: Samsung V-NAND® Source: Samsung PM1733 Product Brief (annotated). The flash memory in Samsung's Accused Instrumentalities comprises blocks (erase units). See:

Writing and Erasing NAND

Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept

Source: Samsung SSD White Paper, at 17.

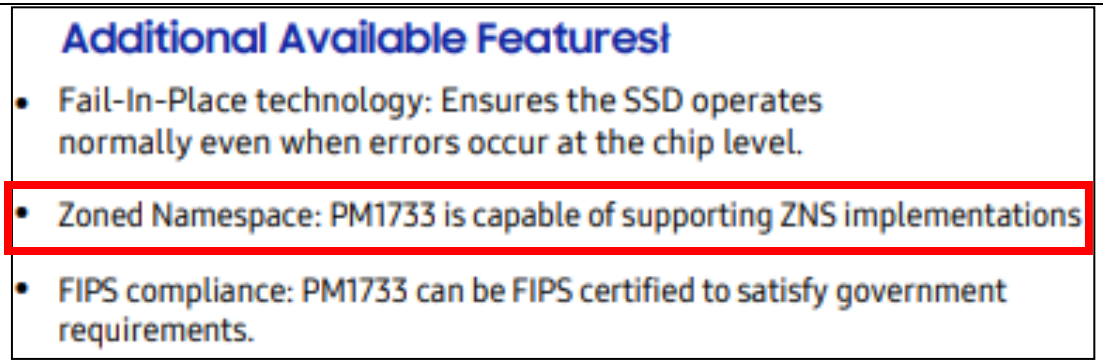
Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP).

Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block.

Source: Boukhobza, at 22 (highlighted).

		2.2.1. *Erase-before-write constraint* Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: *Id.* at 24.
19b	circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is accompanied by incoming address information;	Samsung's Accused Instrumentalities contain circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is accompanied by incoming address information. Samsung's Accused Instrumentalities are able to receive and execute a Write command from a host. A Write command is a data write request. A Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. See: **3.3.6 Write command** The Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. The host may also specify protection information to include as part of the operation. Source: NVM Command Set Specification, at 53. The Write command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which data is to be written. The NSID and SLBA together are incoming address information. See:



Source: Samsung PM1733 Product Brief (annotated).

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field (07) Standard Command	Opcode by Field (06:02) Function	Opcode by Field (01:00) Data Transfer [3]	Combined Opcode [1]	Command [2]	Reference
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVM Command Set Specification, at 107.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2

Source: ZNS Specification, at 18 (annotated).

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

Samsung's Accused Instrumentalities are able to receive and execute a Zone Management Send command specifying a Reset Zone action (erase request) from the host. See:

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

Source: ZNS Specification, at 29.

3.4.3.1.4 Reset Zone

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

a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. The Zone Management Send command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which the Zone Management Send command applies. The NSID and SLBA together are incoming address information. See:

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

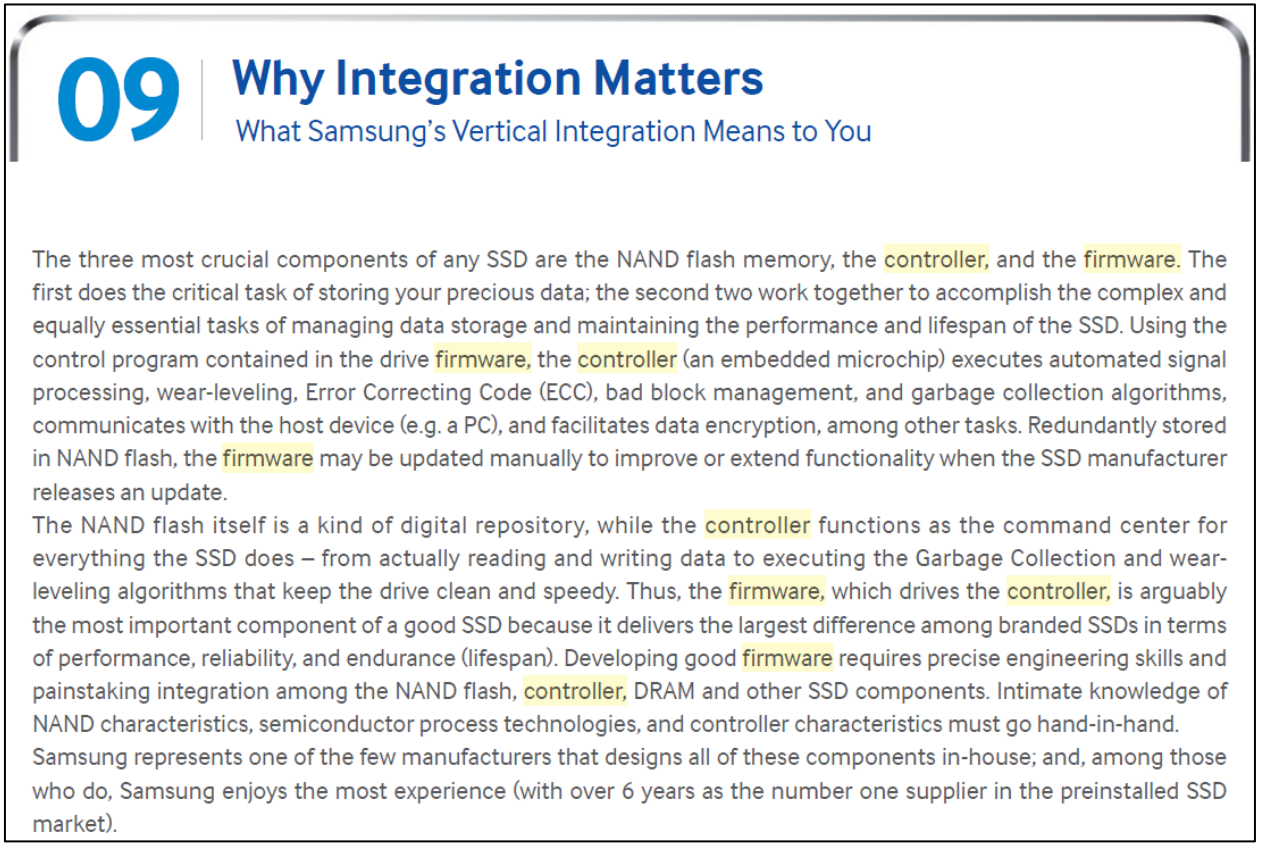
Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVM Command Set Specification, at 107. **3.4.3 Zone Management Send command** The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11, and Command Dword 13 fields. All other command specific fields are reserved. **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. Source: ZNS Specification, at 29.
19c	wherein said circuitry is to,	Samsung's Accused Instrumentalities comprise circuitry to, for each of incoming requests, perform what is described in elements [19d] through [19g]. See below.

	for each of the incoming requests:	Samsung's Accused Instrumentalities contain a processor or memory controller that executes firmware (circuitry). See:  Source: Samsung PM1733 Product Brief.

		 09 Why Integration Matters What Samsung's Vertical Integration Means to You The three most crucial components of any SSD are the NAND flash memory, the controller, and the firmware. The first does the critical task of storing your precious data; the second two work together to accomplish the complex and equally essential tasks of managing data storage and maintaining the performance and lifespan of the SSD. Using the control program contained in the drive firmware, the controller (an embedded microchip) executes automated signal processing, wear-leveling, Error Correcting Code (ECC), bad block management, and garbage collection algorithms, communicates with the host device (e.g. a PC), and facilitates data encryption, among other tasks. Redundantly stored in NAND flash, the firmware may be updated manually to improve or extend functionality when the SSD manufacturer releases an update. The NAND flash itself is a kind of digital repository, while the controller functions as the command center for everything the SSD does – from actually reading and writing data to executing the Garbage Collection and wear-leveling algorithms that keep the drive clean and speedy. Thus, the firmware, which drives the controller, is arguably the most important component of a good SSD because it delivers the largest difference among branded SSDs in terms of performance, reliability, and endurance (lifespan). Developing good firmware requires precise engineering skills and painstaking integration among the NAND flash, controller, DRAM and other SSD components. Intimate knowledge of NAND characteristics, semiconductor process technologies, and controller characteristics must go hand-in-hand. Samsung represents one of the few manufacturers that designs all of these components in-house; and, among those who do, Samsung enjoys the most experience (with over 6 years as the number one supplier in the preinstalled SSD market). Source: Source: Samsung SSD White Paper, at 30.
19d	derive, from the accompanying incoming address information, a first address portion and a	Samsung's Accused Instrumentalities contain said circuitry to derive, from the accompanying incoming address information, a first address portion and a second address portion. Samsung's Accused Instrumentalities are able to derive, from the accompanying incoming address information, a Namespace Identifier or NSID (a first address portion) and a starting LBA (a second address portion). See:

second address portion,

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVMe Base Specification, at 107.

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

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.

Source: ZNS Specification, at 29.

19e	identify an addressed block device from the first address portion,	Samsung's Accused Instrumentalities contain said circuitry to identify an addressed block device from the first address portion. Samsung's Accused Instrumentalities are able to identify an addressed namespace (block device) from the Namespace Identifier or NSID (the first address portion). See:

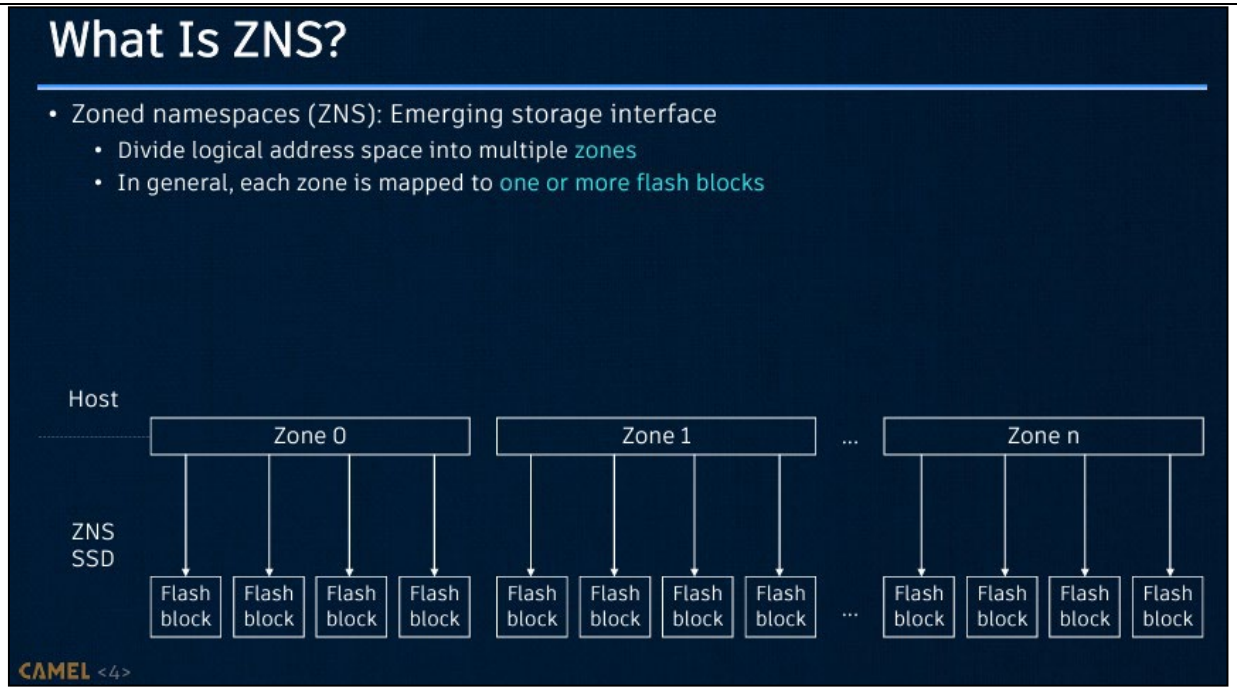
Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** | **Description** 07:04 | **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
19f	use a first operation on the second address portion to subdivide the incoming address information into first bit fields, and to identify an addressed segment within the addressed block device from a first one	Samsung's Accused Instrumentalities contain said circuitry to use a first operation on the second address portion to subdivide the incoming address information into first bit fields, and to identify an addressed segment within the addressed block device from a first one of the first bit fields, wherein the addressed segment corresponds to a subset of the erase units of the flash memory. Samsung's Accused Instrumentalities contain zones (segments) that correspond to subsets of the erase units of the flash memory. See:

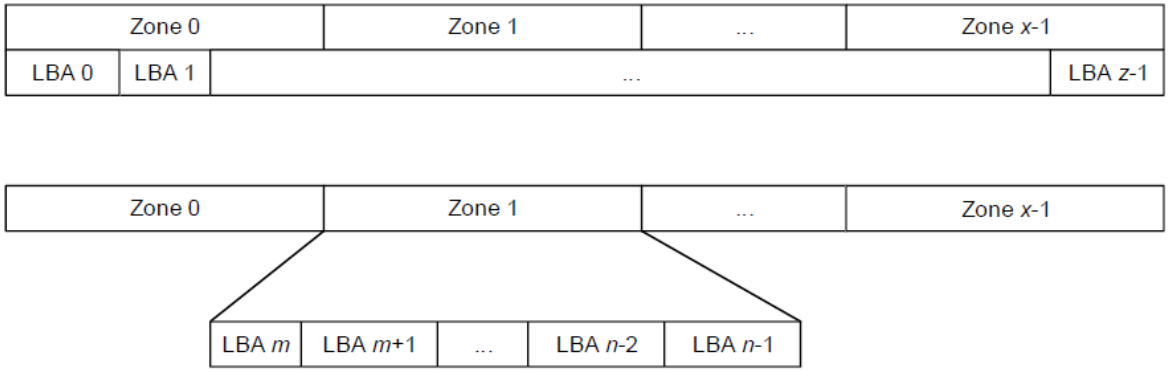
	of the first bit fields, wherein the addressed segment corresponds to a subset of the erase units of the flash memory, and	 Source: Bae, at 4. This new attribute was introduced to allow for the zone size to remain a power of two number of logical blocks (facilitating logical block to zone number conversions) while allowing optimized mapping of a zone storage capacity to the underlying media characteristics. For instance, in the case a flash based device, a zone capacity can be aligned to the size of flash erase blocks without requiring that the device implements a power-of-two sized erased block. Source: Zoned Storage Introduction.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

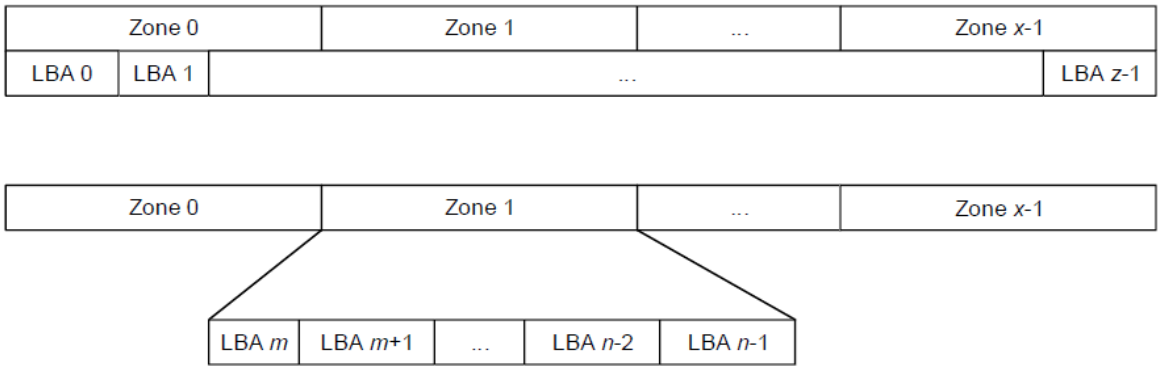
Samsung's Accused Instrumentalities are able to use a first operation on the starting LBA to subdivide it into bit fields and to identify an addressed zone (segment) within the addressed namespace (block device) from a first bit field. All zones are equally sized with contiguous non-overlapping ranges of LBAs. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.*

3.3.7 Write command

The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

3.3.7.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19.

Figure 19: Write – Command Specific Status Values

Value	Description
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.
BCh	**Zone Invalid Write:** The write to a zone was not at the write pointer.
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.

Source: *Id.* at 20.

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

If the command completes successfully, depending on the Zone Send Action field and the current states of the zones specified by the command, then that command may affect zones in various ways, including the following:

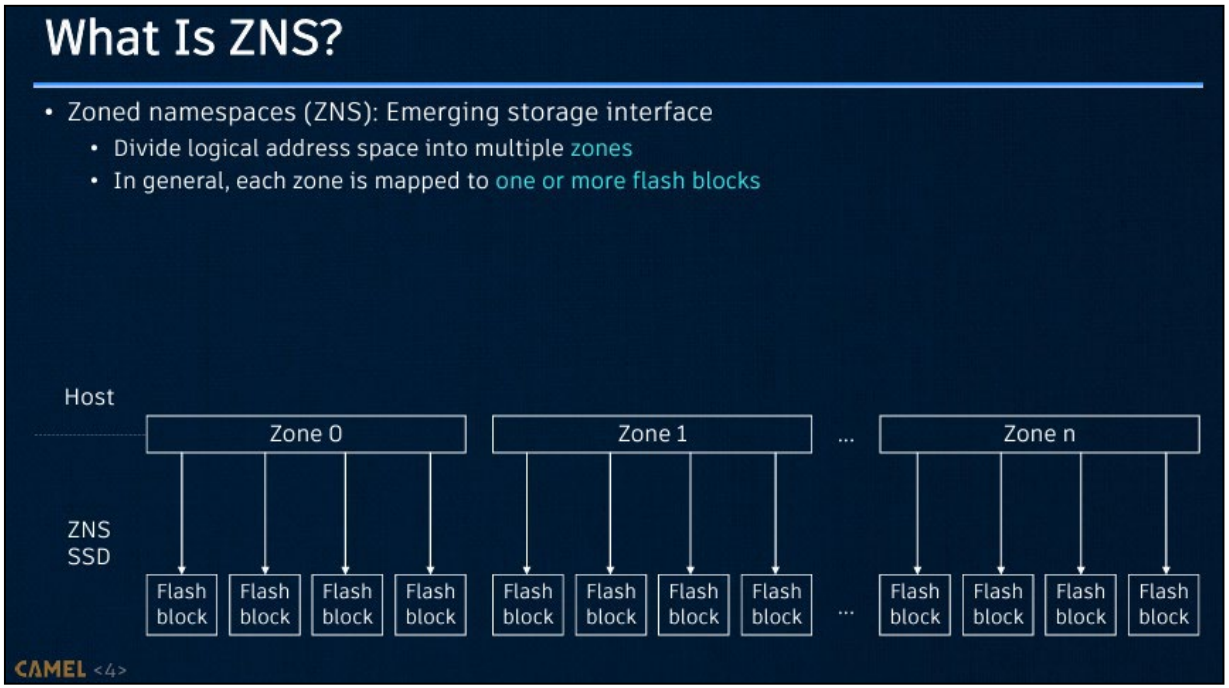
a) the zone state may change;
b) the Zone Descriptor Extension data may change; and
c) the Zone Descriptor Extension Valid zone attribute bit may change.

Source: *Id.* at 29.

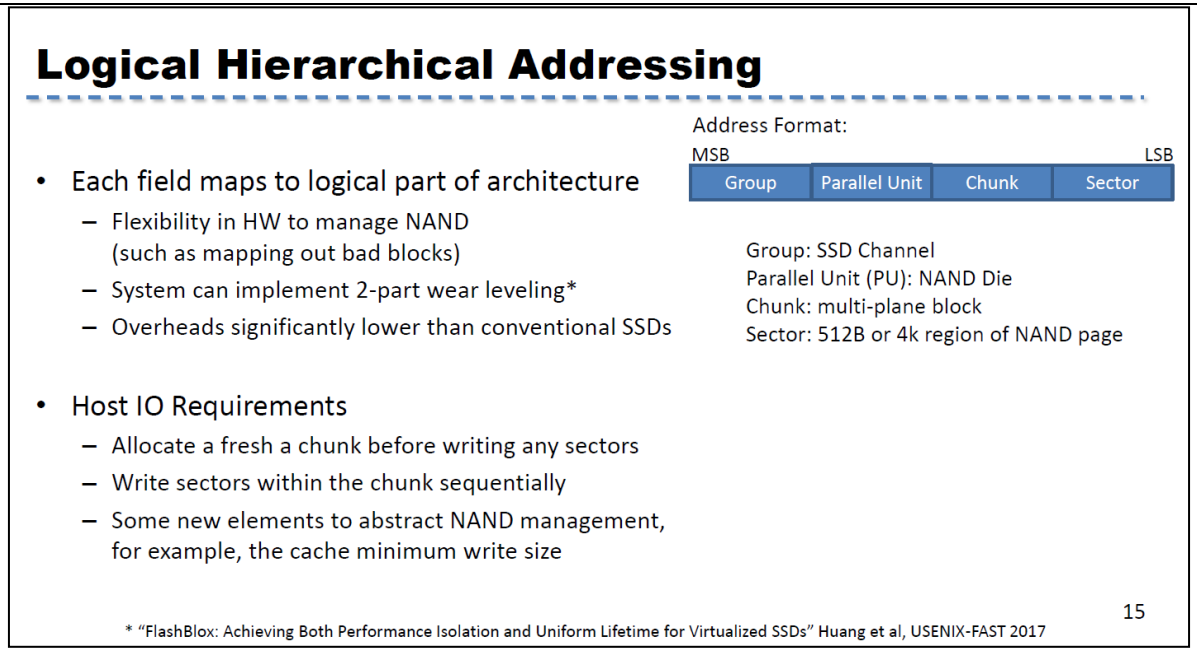
Figure 40: Zone Management Send – Command Specific Status Values

Value	Description
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.

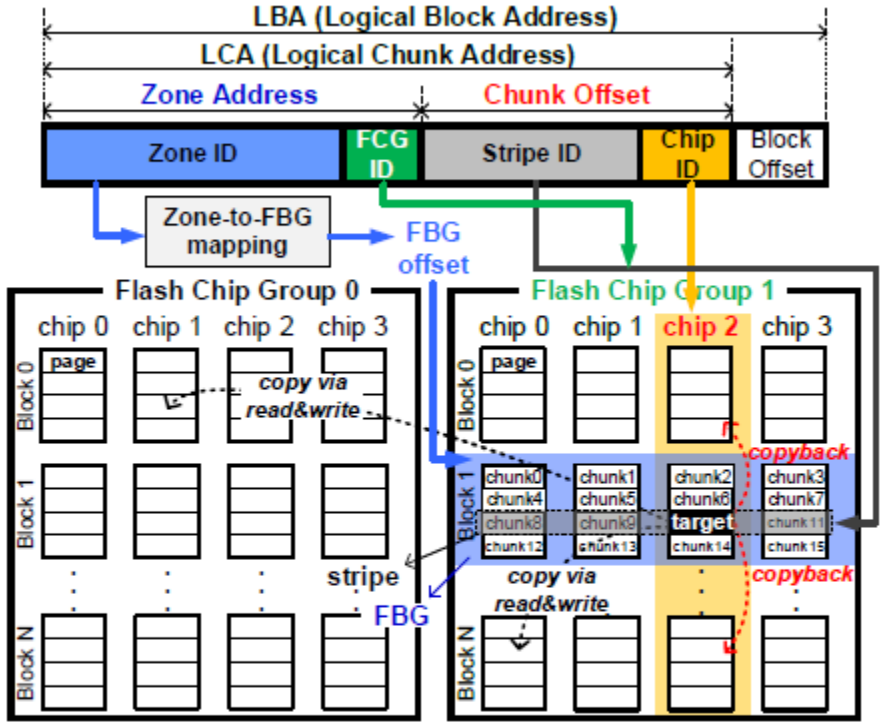
Source: *Id.* at 32.



Source: Bae, at 4.



Source: Microsoft Denali Presentation, at 15.

		 Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted. Source: Han, at 149.
19g	for each of the data write requests, use a second one of the first bit	Samsung's Accused Instrumentalities contain said circuitry to, for each of the data write requests, use a second one of the first bit fields as a third address portion, and use a second division operation to subdivide third address portion into second bit fields, to identify an address value to select one of the

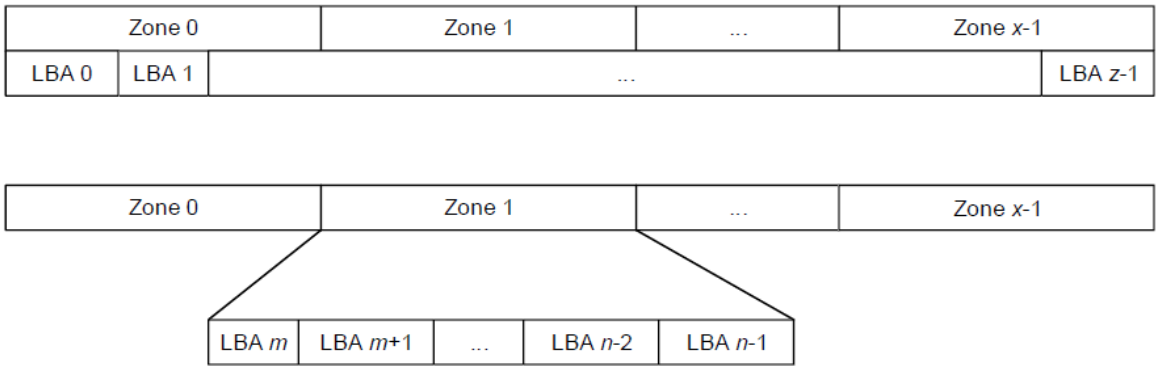
	fields as a third address portion and use a second operation to subdivide the third address portion into second bit fields, to identify an address value to select one of the erase units from a first one of the second bit fields and to identify a storage location within the selected one of the erase units from a second one of the second bit fields;	erase units from a first one of the second bit fields and to identify a storage location within the selected one of the erase units from a second one of the second bit fields. Samsung's Accused Instrumentalities are able to, for each Write command (data write request), use a second bit field of the starting LBA as a third address portion. Samsung's Accused Instrumentalities are able to further subdivide the third address portion to identify an address value to select one of the erase units corresponding to the zone addressed by the first bit field, and to identify a logical block and/or page (storage location) within such erase unit, using a second division operation. Samsung's Accused Instrumentalities contain multiple equally sized erase units in each zone such that the third address portion is further subdivided to identify an address value to select a particular erase unit within a zone. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

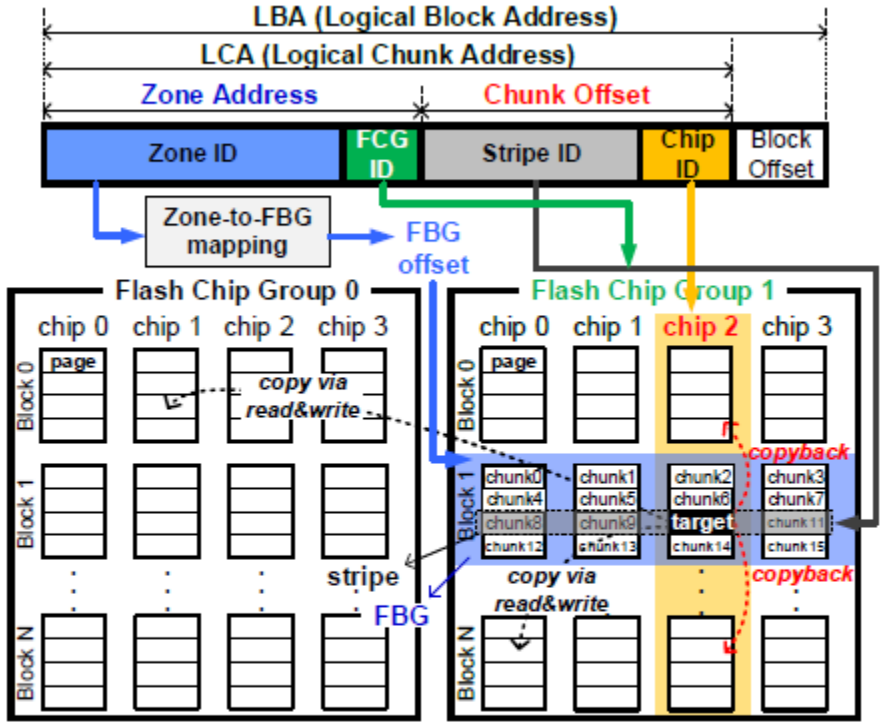
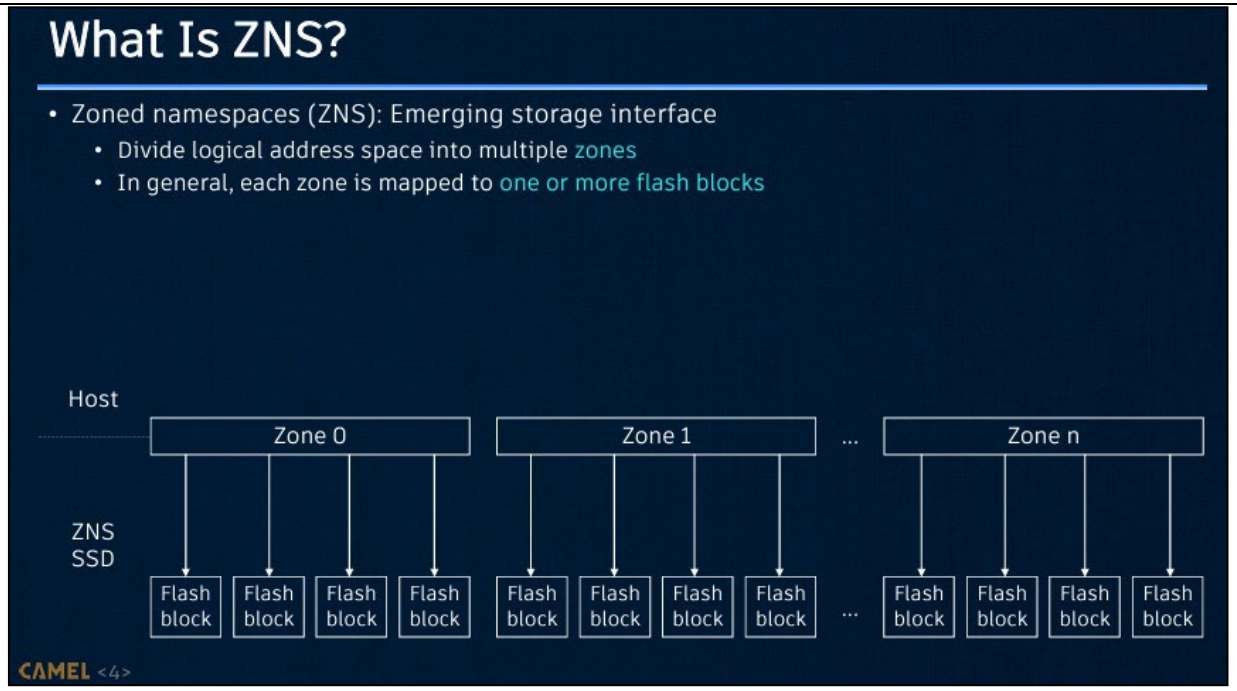
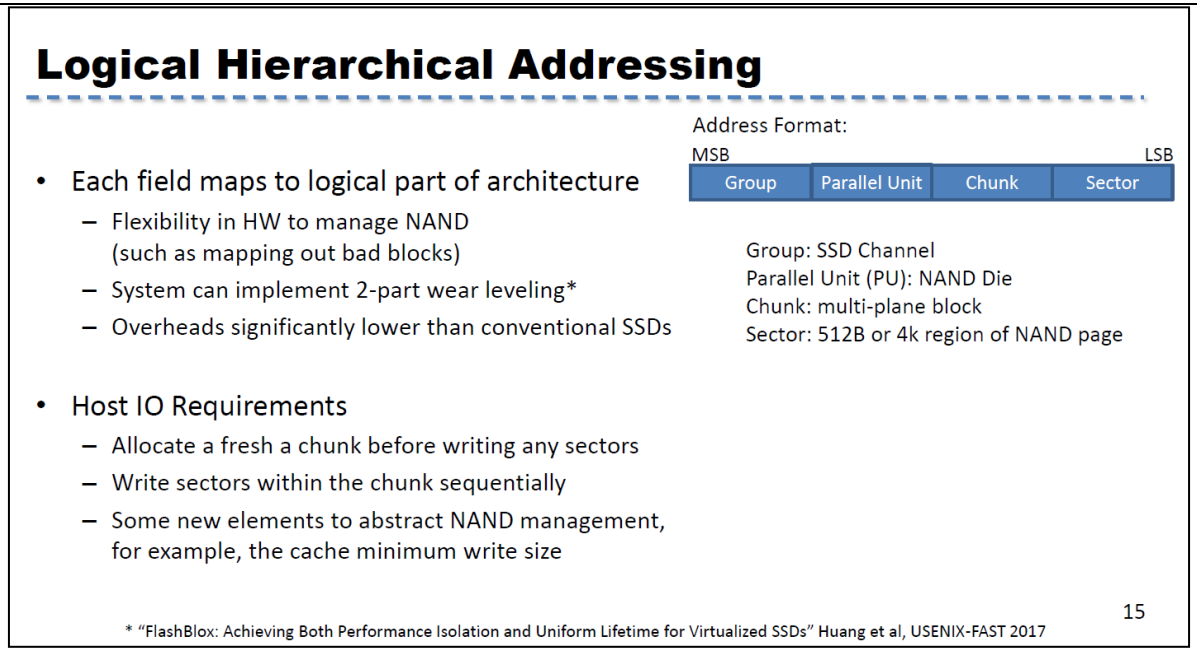


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

		 Source: Microsoft Denali Presentation, at 15.
19h	wherein said circuitry is to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the	Samsung's Accused Instrumentalities contain said circuitry to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the data write requests. Samsung's Accused Instrumentalities are able to execute data write requests by writing (programming) data into the identified logical block and/or page (storage location). See:

	data write requests;	**3.3.7 Write command** The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2). **3.3.7.1 Command Completion** Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19. **Figure 19: Write – Command Specific Status Values** **Value** / **Description** 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. BCh — **Zone Invalid Write:** The write to a zone was not at the write pointer. 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. Source: ZNS Specification, at 20.

		Figure 88: Common Command Format **Bytes** | **Description** 07:04 | **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107. **Figure 69: Write – Command Dword 10 and Command Dword 11** **Bits** | **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. Source: NVM Command Set Specification, at 54.
19i	wherein said circuitry is also to execute each one of the incoming erase requests by controlling erasure in the flash memory	Samsung's Accused Instrumentalities contain said circuitry to execute each one of the incoming erase requests by controlling erasure in the flash memory of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests. Samsung's Accused Instrumentalities are able to execute the Zone Management Send command specifying a Reset Zone action (erase request) by controlling erasure of one or more of the erase units corresponding to the addressed zone (segment). See:

	of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests.	The addressed zone was identified from the incoming address information accompanying the one of the incoming erase requests. See claim element [19f]. **Figure 39: Zone Management Send – Command Dword 13** **Bits** / **Description** 31:09 — Reserved 08 — **Select All:** If this bit is set to '1', then the SLBA field is 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. **Value** / **Description** / **Refer to section** 00h — Reserved 01h — **Close Zone:** Close one or more zones. — 3.4.3.1.1 02h — **Finish Zone:** Finish one or more zones. — 3.4.3.1.2 03h — **Open Zone:** Open one or more zones. — 3.4.3.1.3 04h — **Reset Zone:** Reset one or more zones. — 3.4.3.1.4 05h — **Offline Zone:** Offline one or more zones. — 3.4.3.1.5 06h to 0Fh — Reserved 10h — **Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone. — 3.4.3.1.6 11h to FFh — Reserved Source: ZNS Specification, at 29. **3.4.3.1.4 Reset Zone** If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the: a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17. Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP). Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

		– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block. Source: Boukhobza, at 22 (highlighted). **2.2.1. *Erase-before-write constraint*** Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: Boukhobza at 24 (highlighted).
20pre	A storage device comprising:	To the extent the preamble is considered a limitation, Samsung's Accused Instrumentalities contain a storage device.
20a	flash memory, the flash memory to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate	Samsung's Accused Instrumentalities comprise flash memory which is to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment corresponding to a respective, mutually exclusive set of one or more erase units, each segment being associated with only one of the block devices, each erase unit being associated with only one of the segments. Samsung's Accused Instrumentalities comprise NAND flash memory. For example, Samsung's PM1733 contains NAND flash memory. See:

	level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment corresponding to a respective, mutually exclusive set of one or more erase units, each segment being associated with only one of the block devices, each erase unit being associated with only one of the segments; and	**Samsung PM1733 specifications** Form factor: U.2 / 2.5" Capacity: 1.92 TB, 3.84 TB, 7.68TB, and 15.36TB Host interface: PCIe Gen 3/4 x4 Spec Compliance: NVMe spec rev. 1.3; PCI Express base specification rev. 4.0 NAND flash memory: Samsung V-NAND® Source: Samsung PM1733 Product Brief (annotated). The NAND flash memory in Samsung's Accused Instrumentalities comprises blocks (erase units). See: **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17.

Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP).

Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block.

Source: Boukhobza, at 22 (highlighted).

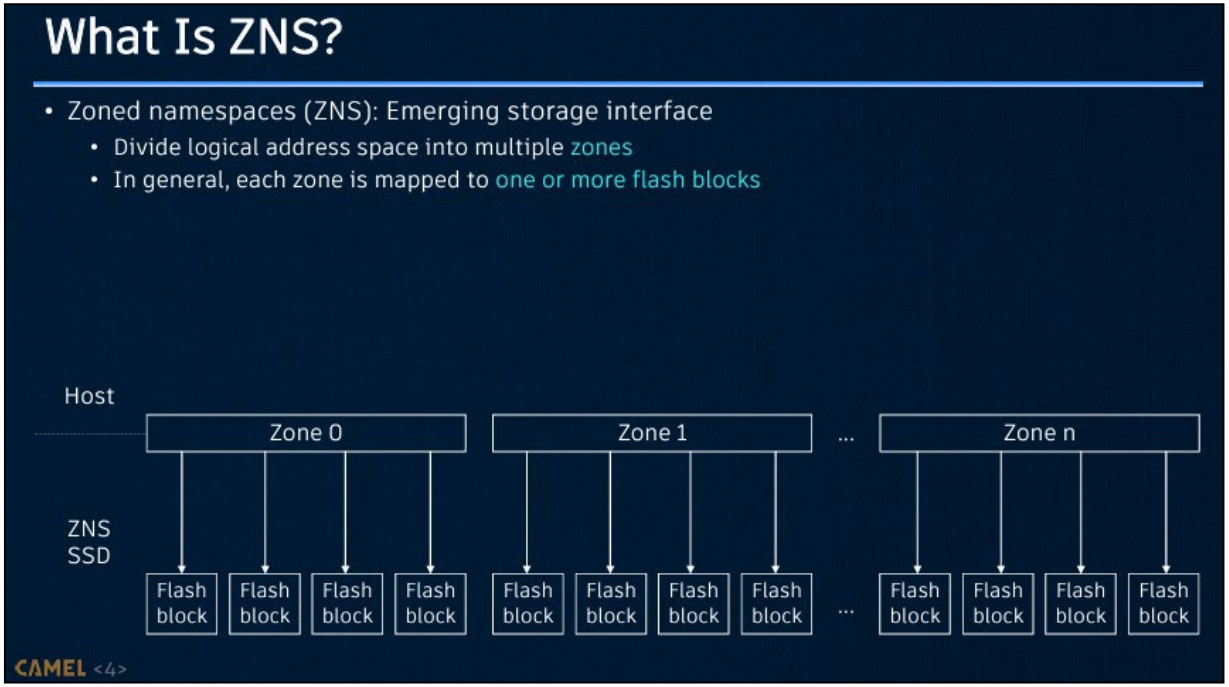
2.2.1. *Erase-before-write constraint*

Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of

Source: *Id.* at 24.

Samsung's Accused Instrumentalities are able to have zones (segments) at a hierarchically-intermediate level. Each zone corresponds to a mutually exclusive set of one or more erase units which are associated

with only one zone. Samsung's Accused Instrumentalities are able to addressable storage locations within each zone at a hierarchically-inferior level. See:



Source: Bae, at 4.

> This new attribute was introduced to allow for the zone size to remain a power of two number of logical blocks (facilitating logical block to zone number conversions) while allowing optimized mapping of a zone storage capacity to the underlying media characteristics. For instance, in the case a flash based device, a zone capacity can be aligned to the size of flash erase blocks without requiring that the device implements a power-of-two sized erased block.

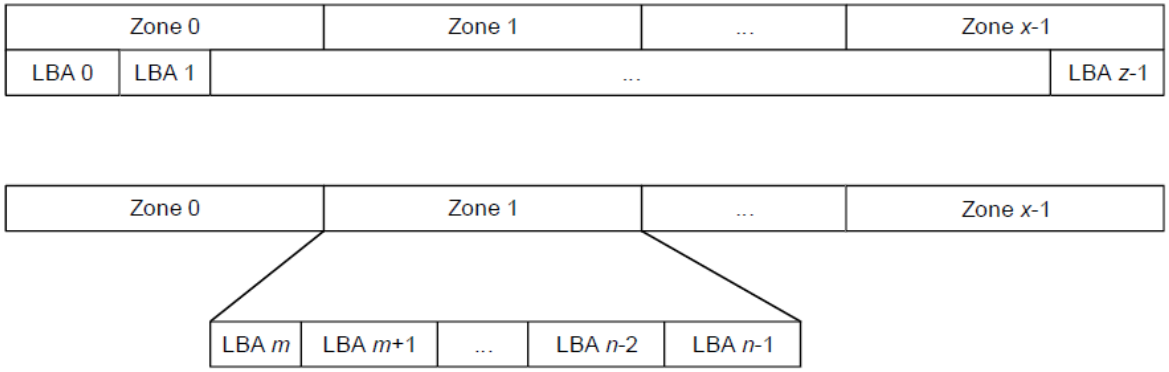
Source: Zoned Storage Introduction.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

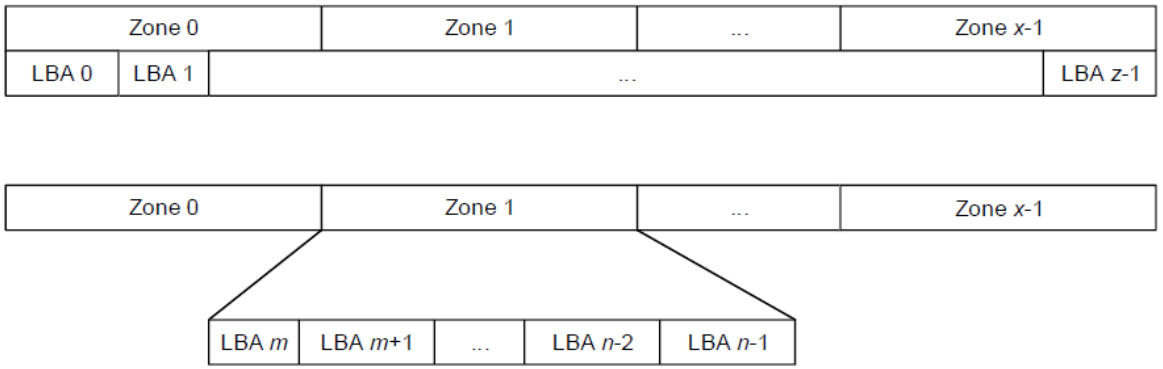
Samsung's Accused Instrumentalities are able to have one or more namespaces (block devices) at a hierarchically-superior level. Each zone is associated with only one of the namespaces. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.*

1.5.37 namespace

A formatted quantity of non-volatile memory that may be directly accessed by a host.

1.5.38 namespace ID (NSID)

An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID.

Source: NVMe Base Specification, at 22.

A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace.

Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
20b	circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is	Samsung's Accused Instrumentalities comprise circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is accompanied by incoming address information. Samsung's Accused Instrumentalities are able to receive and execute a Write command from a host. A Write command is a data write request. A Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. See: **3.3.6 Write command** The Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. The host may also specify protection information to include as part of the operation.

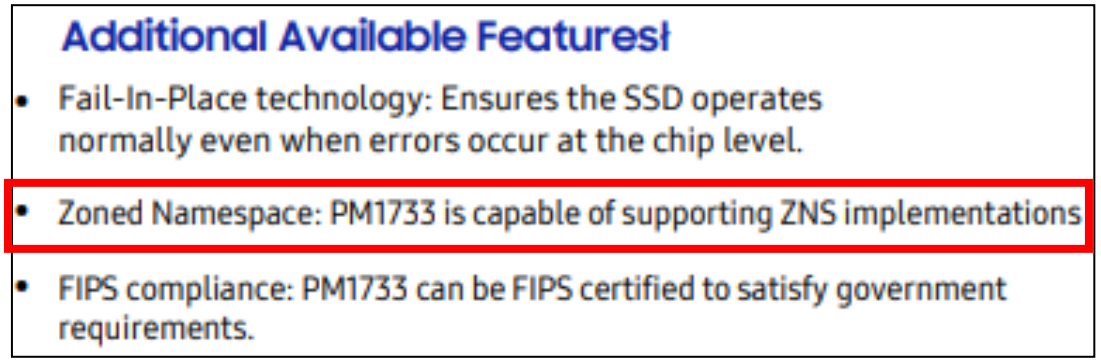
	accompanied by incoming address information;	Source: NVM Command Set Specification, at 53. The Write command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which data is to be written. The NSID and SLBA together are incoming address information. See:  Source: Samsung PM1733 Product Brief (annotated).

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVM Command Set Specification, at 107.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2

Source: ZNS Specification, at 18 (annotated).

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

Samsung's Accused Instrumentalities are able to receive and execute a Zone Management Send command specifying a Reset Zone action (erase request) from the host. See:

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

Source: ZNS Specification, at 29.

3.4.3.1.4 Reset Zone

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

a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. The Zone Management Send command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which the Zone Management Send command applies. The NSID and SLBA together are incoming address information. See:

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVM Command Set Specification, at 107. **3.4.3 Zone Management Send command** The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11, and Command Dword 13 fields. All other command specific fields are reserved. **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. Source: ZNS Specification, at 29.
20c	wherein said circuitry is to,	Samsung's Accused Instrumentalities comprise circuitry to, for each of the incoming requests, perform what is described in elements [20d] through [20g]. See below.

	for each of incoming requests:	Samsung's Accused Instrumentalities contain a processor or memory controller that executes firmware (circuitry). See: 

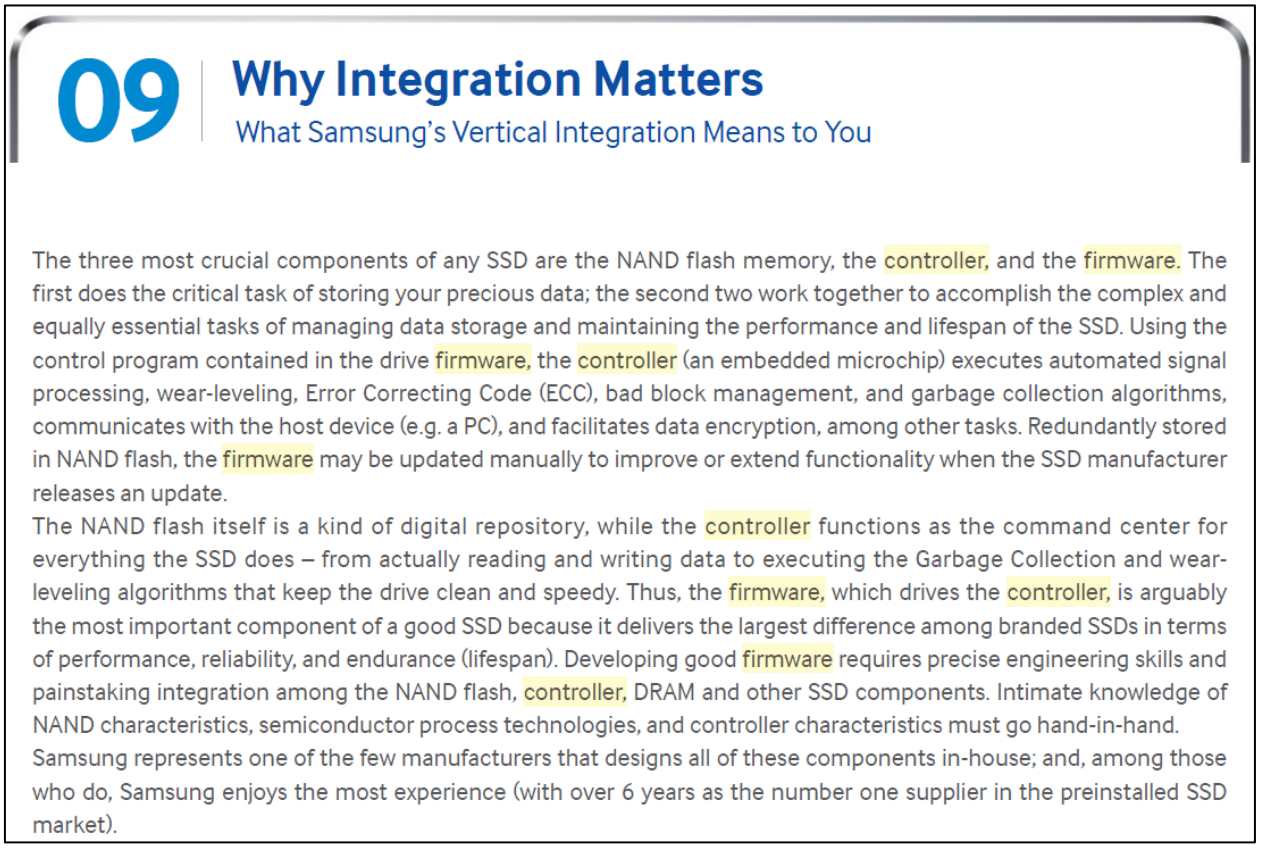
		Source: Samsung PM1733 Product Brief.  09 Why Integration Matters What Samsung's Vertical Integration Means to You The three most crucial components of any SSD are the NAND flash memory, the controller, and the firmware. The first does the critical task of storing your precious data; the second two work together to accomplish the complex and equally essential tasks of managing data storage and maintaining the performance and lifespan of the SSD. Using the control program contained in the drive firmware, the controller (an embedded microchip) executes automated signal processing, wear-leveling, Error Correcting Code (ECC), bad block management, and garbage collection algorithms, communicates with the host device (e.g. a PC), and facilitates data encryption, among other tasks. Redundantly stored in NAND flash, the firmware may be updated manually to improve or extend functionality when the SSD manufacturer releases an update. The NAND flash itself is a kind of digital repository, while the controller functions as the command center for everything the SSD does – from actually reading and writing data to executing the Garbage Collection and wear-leveling algorithms that keep the drive clean and speedy. Thus, the firmware, which drives the controller, is arguably the most important component of a good SSD because it delivers the largest difference among branded SSDs in terms of performance, reliability, and endurance (lifespan). Developing good firmware requires precise engineering skills and painstaking integration among the NAND flash, controller, DRAM and other SSD components. Intimate knowledge of NAND characteristics, semiconductor process technologies, and controller characteristics must go hand-in-hand. Samsung represents one of the few manufacturers that designs all of these components in-house; and, among those who do, Samsung enjoys the most experience (with over 6 years as the number one supplier in the preinstalled SSD market). Source: Source: Samsung SSD White Paper, at 30.
20d	derive a first address portion and a second address portion,	Samsung's Accused Instrumentalities contain said circuitry to derive a first address portion and a second address portion. Samsung's Accused Instrumentalities are able to derive a Namespace Identifier or NSID (a first address portion) and a starting LBA (a second address portion). See:

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVMe Base Specification, at 107.

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

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.

Source: ZNS Specification, at 29.

20e	identify an addressed one of the block devices from the first address portion,	Samsung's Accused Instrumentalities comprise said circuitry to identify an addressed one of the block devices from the first address portion. Samsung's Accused Instrumentalities are able to identify an addressed namespace (block device) from the Namespace Identifier or NSID (the first address portion). See:

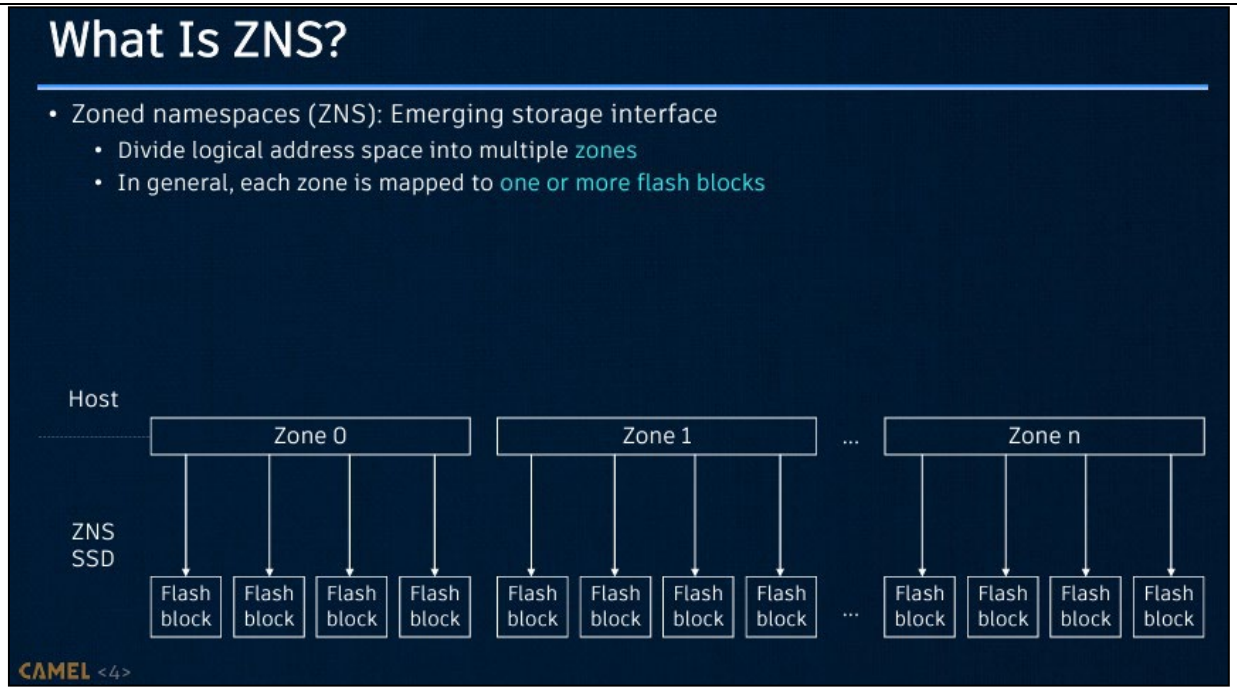
Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
20f	use a first operation to subdivide the second address portion into first bit fields, and to identify an addressed segment within the addressed block device from a first one of the first bit fields, and to identify a third	Samsung's Accused Instrumentalities comprise said circuitry to use a first operation to subdivide the second address portion into first bit fields, and to identify an addressed segment within the addressed block device from a first one of the first bit fields, and to identify a third address portion from a second one of the first bit fields, wherein the addressed segment corresponds to a subset of the erase units of the flash memory. Samsung's Accused Instrumentalities contain zones (segments) that correspond to subsets of the erase units of the flash memory. See:

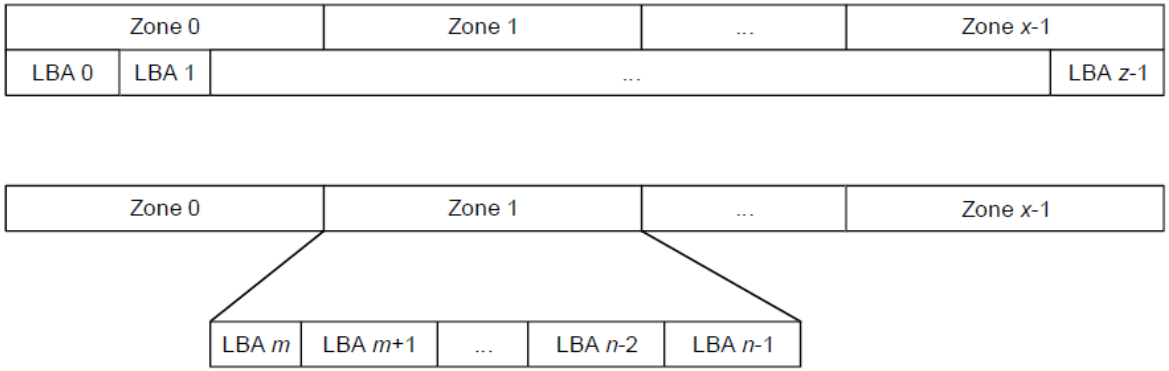
	address portion from a second one of the first bit fields, wherein the addressed segment corresponds to a subset of the erase units of the flash memory, and	 Source: Bae, at 4. This new attribute was introduced to allow for the zone size to remain a power of two number of logical blocks (facilitating logical block to zone number conversions) while allowing optimized mapping of a zone storage capacity to the underlying media characteristics. For instance, in the case a flash based device, a zone capacity can be aligned to the size of flash erase blocks without requiring that the device implements a power-of-two sized erased block. Source: Zoned Storage Introduction.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

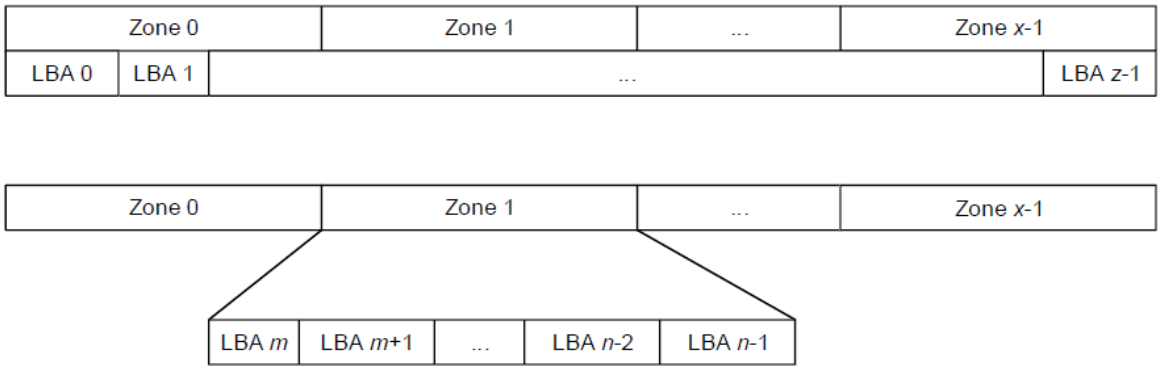
Samsung's Accused Instrumentalities are able to use a first operation on the starting LBA to subdivide it into bit fields and to identify an addressed zone (segment) within the addressed namespace (block device) from a first bit field. All zones are equally sized with contiguous non-overlapping ranges of LBAs. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.*

3.3.7 Write command

The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

3.3.7.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19.

Figure 19: Write – Command Specific Status Values

Value	Description
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.
BCh	**Zone Invalid Write:** The write to a zone was not at the write pointer.
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.

Source: *Id.* at 20.

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

If the command completes successfully, depending on the Zone Send Action field and the current states of the zones specified by the command, then that command may affect zones in various ways, including the following:

a) the zone state may change;
b) the Zone Descriptor Extension data may change; and
c) the Zone Descriptor Extension Valid zone attribute bit may change.

Source: *Id.* at 29.

Figure 40: Zone Management Send – Command Specific Status Values

Value	Description
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.

Source: *Id.* at 32.

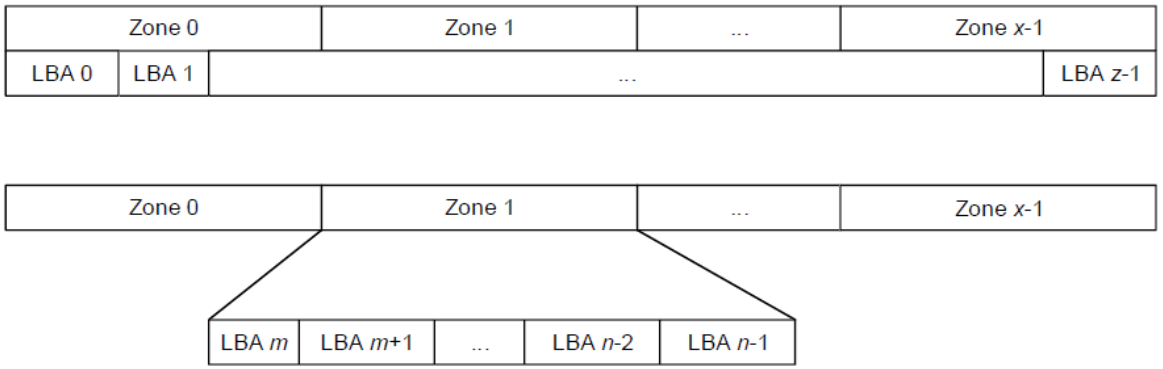
Samsung's Accused Instrumentalities are able to identify a third address portion from a second bit field. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.* at 8.

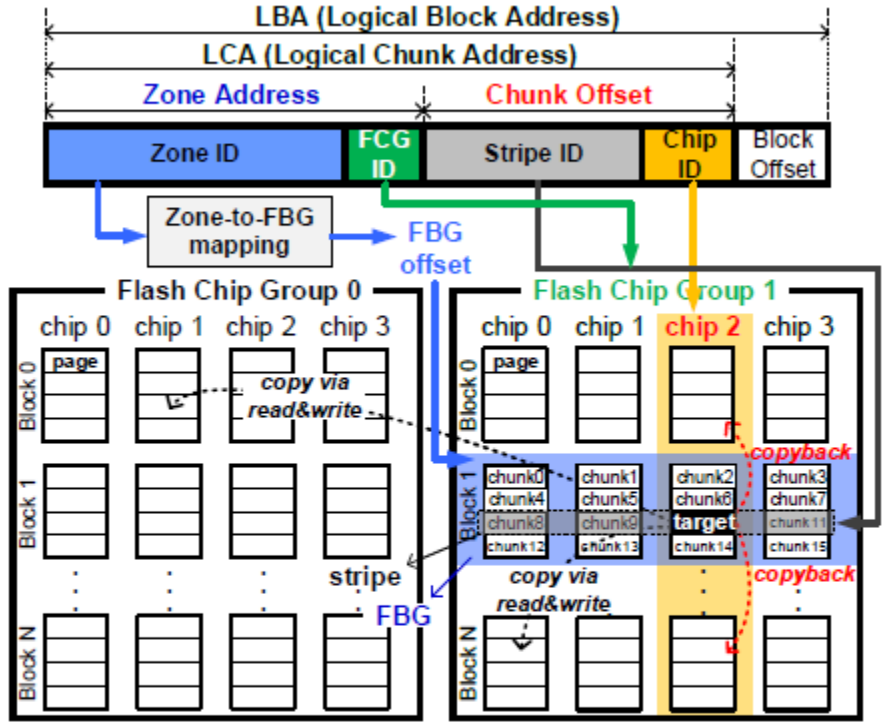
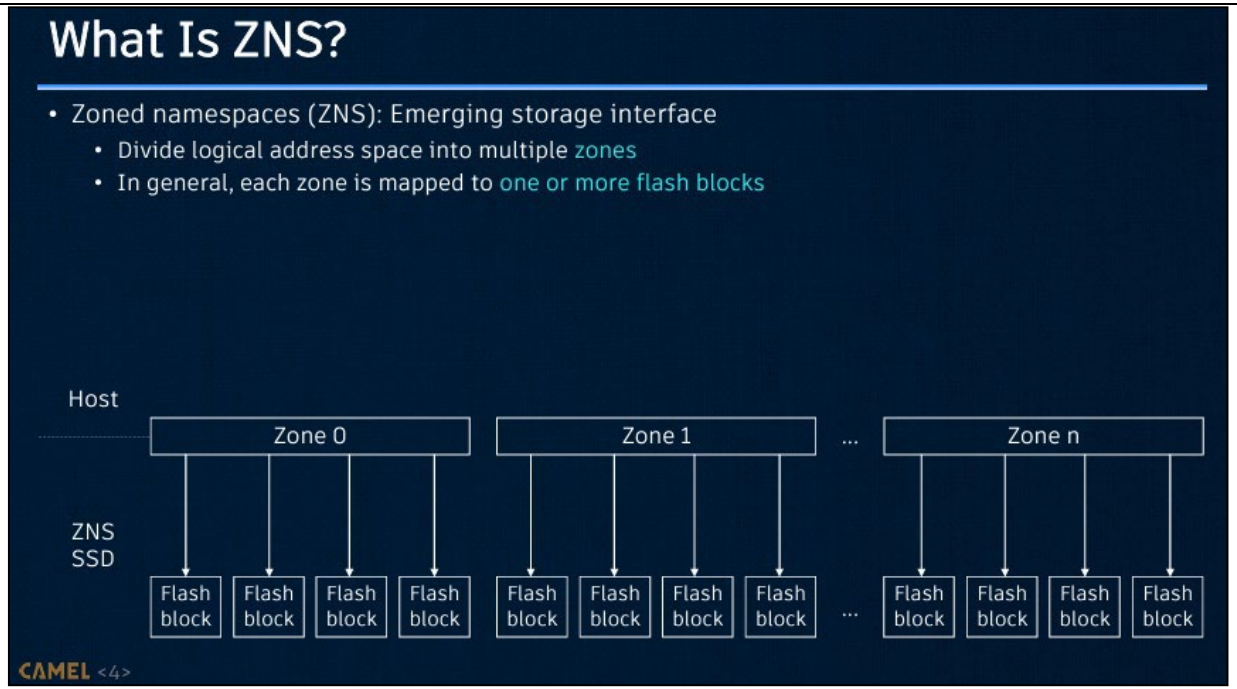
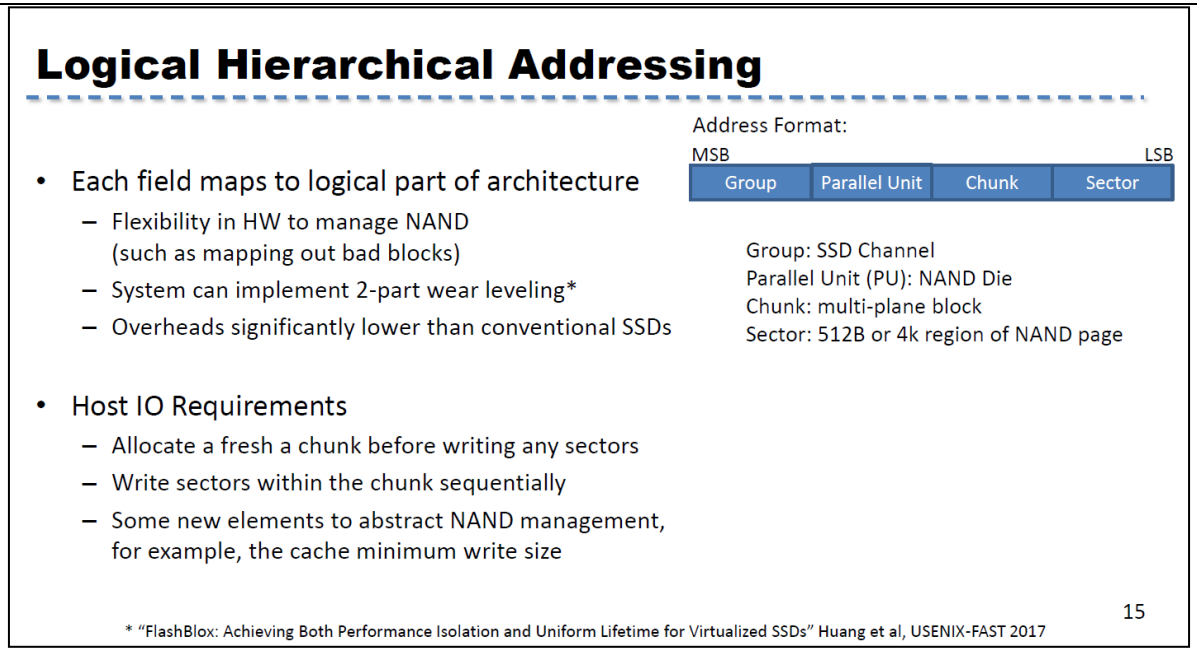


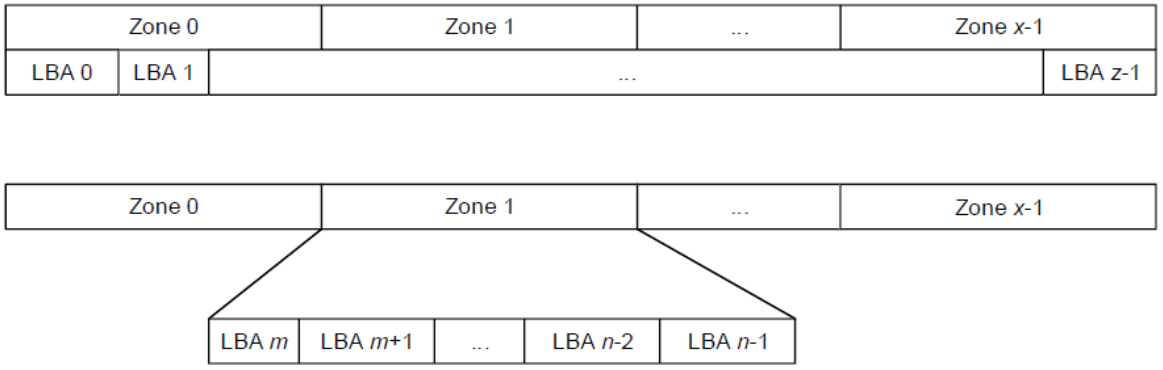
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

		 Source: Microsoft Denali Presentation, at 15.
20g	for each of the data write requests, use a second operation to subdivide the third address portion into second bit fields, to identify an address value to select one of the	Samsung's Accused Instrumentalities contain said circuitry to, for each of the data write requests, use a second operation to subdivide third address portion into second bit fields, to identify an address value to select one of the erase units from a first one of the second bit fields, and to identify a storage location within the selected one of the erase units from a second one of the second bit fields. Samsung's Accused Instrumentalities are able to, for each Write command, use a second bit field of the starting LBA as a third address portion. Samsung's Accused Instrumentalities are able to further subdivide the third address portion to identify an address value to select one of the erase units corresponding to the zone addressed by the first bit field, and to identify a storage location within such erase unit, using a second division operation. Samsung's Accused Instrumentalities contain multiple equally sized erase units in each zone such that the third address portion is subdivided to identify an address value to select a particular erase unit within a zone. See:

	erase units from a first one of the second bit fields, and to identify a storage location within the selected one of the erase units from a second one of the second bit fields;	**2.1.1 Namespaces** A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification. A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA. **Figure 2: Zones in a Zoned Namespace**  The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification). Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors. Source: ZNS Specification, at 8.

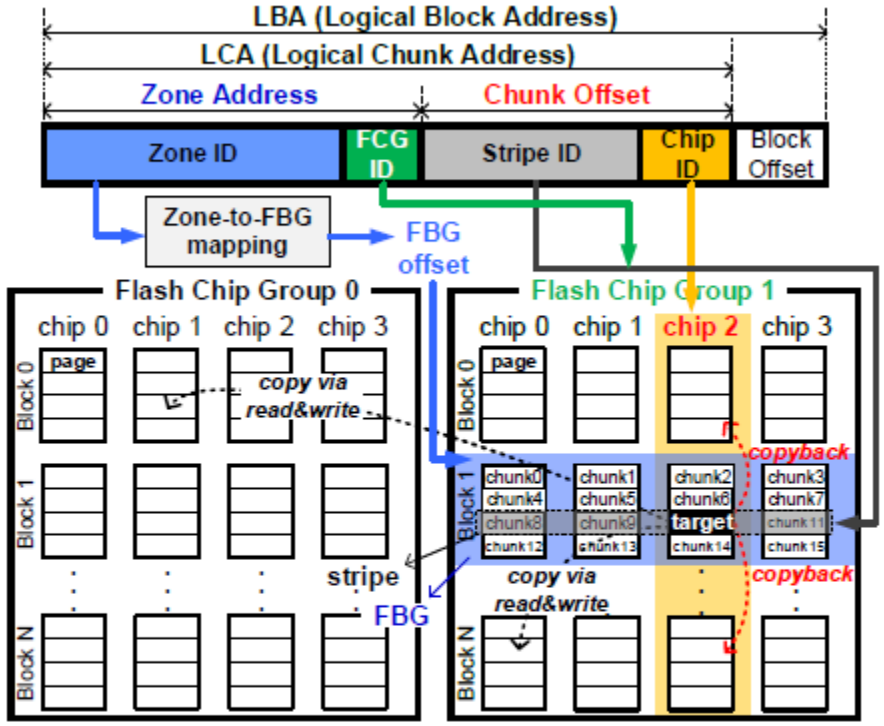
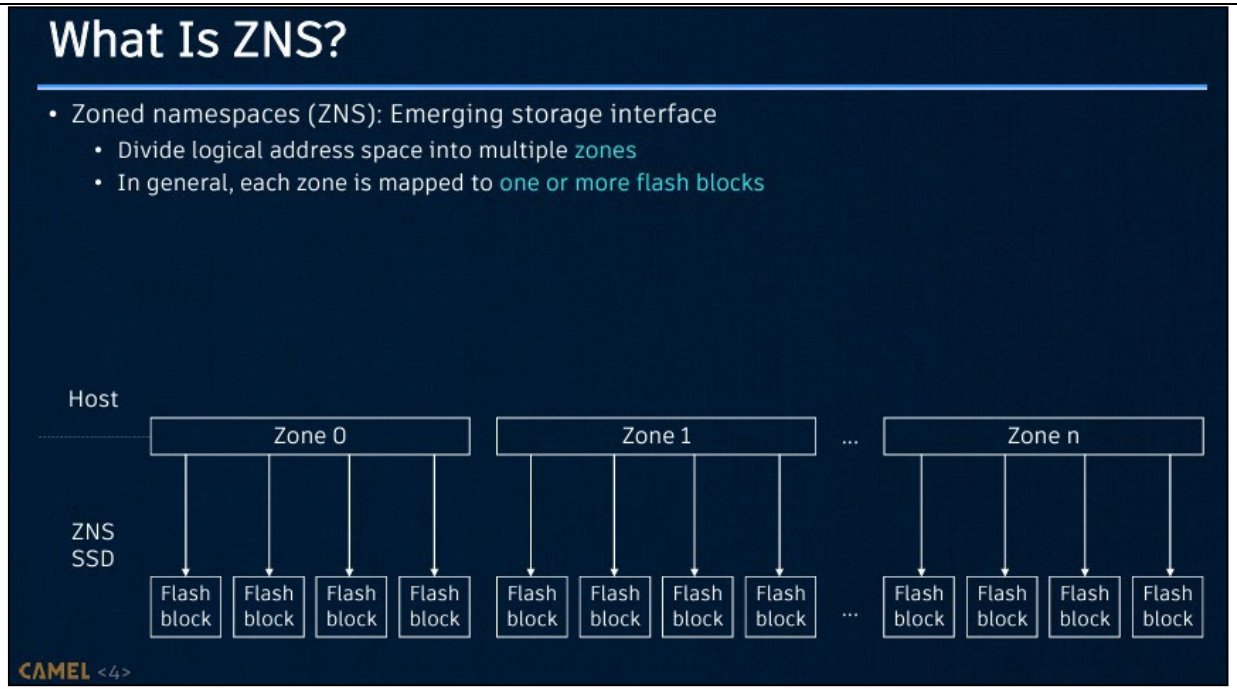
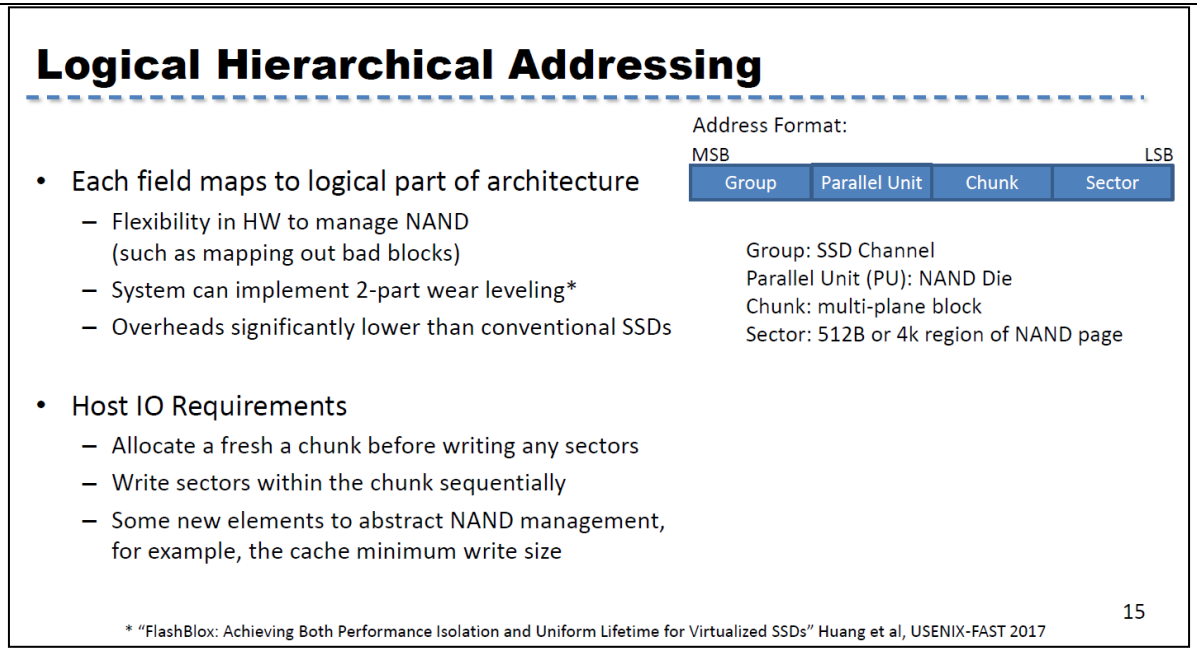


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

		 Source: Microsoft Denali Presentation, at 15.
20h	wherein said circuitry is to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the	Samsung's Accused Instrumentalities comprise said circuitry to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the data write requests. Samsung's Accused Instrumentalities are able to execute data write requests by writing (programming) data into the identified storage location. See:

	data write requests;	**3.3.7 Write command** The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2). **3.3.7.1 Command Completion** Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19. **Figure 19: Write – Command Specific Status Values** **Value** / **Description** 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. BCh — **Zone Invalid Write:** The write to a zone was not at the write pointer. 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. Source: ZNS Specification, at 20.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107. **Figure 69: Write – Command Dword 10 and Command Dword 11** **Bits** / **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. Source: NVM Command Set Specification, at 54.
20i	wherein said circuitry is also to execute each one of the incoming erase requests by controlling erasure in the flash memory	Samsung's Accused Instrumentalities contain said circuitry to execute each one of the incoming erase requests by controlling erasure in the flash memory of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests. Samsung's Accused Instrumentalities are able to execute the Zone Management Send command specifying a Reset Zone action (erase request) by controlling erasure of one or more of the erase units corresponding to the addressed zone. See:

	of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests.	**Figure 39: Zone Management Send – Command Dword 13** **Bits** / **Description** 31:09 — Reserved 08 — **Select All:** If this bit is set to '1', then the SLBA field is 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. **Value** / **Description** / **Refer to section** 00h — Reserved 01h — **Close Zone:** Close one or more zones. — 3.4.3.1.1 02h — **Finish Zone:** Finish one or more zones. — 3.4.3.1.2 03h — **Open Zone:** Open one or more zones. — 3.4.3.1.3 04h — **Reset Zone:** Reset one or more zones. — 3.4.3.1.4 05h — **Offline Zone:** Offline one or more zones. — 3.4.3.1.5 06h to 0Fh — Reserved 10h — **Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone. — 3.4.3.1.6 11h to FFh — Reserved Source: ZNS Specification, at 29. **3.4.3.1.4 Reset Zone** If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the: a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17. Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP). Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

		– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block. Source: Boukhobza, at 22 (highlighted). **2.2.1. *Erase-before-write constraint*** Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: Boukhobza at 24 (highlighted).
21pre	A storage device comprising:	To the extent the preamble is considered a limitation, Samsung's Accused Instrumentalities contain a storage device.
21a	flash memory, the flash memory to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate	Samsung's Accused Instrumentalities comprise flash memory which is to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment corresponding to a respective, mutually exclusive set of one or more erase units, each segment being associated with only one of the block devices, each erase unit being associated with only one of the segments. Samsung's Accused Instrumentalities comprise NAND flash memory. For example, Samsung's PM1733 contains NAND flash memory. See:

	level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment corresponding to a respective, mutually exclusive set of one or more erase units, each segment being associated with only one of the block devices, each erase unit being associated with only one of the segments; and	**Samsung PM1733 specifications** Form factor: U.2 / 2.5" Capacity: 1.92 TB, 3.84 TB, 7.68TB, and 15.36TB Host interface: PCIe Gen 3/4 x4 Spec Compliance: NVMe spec rev. 1.3; PCI Express base specification rev. 4.0 NAND flash memory: Samsung V-NAND® Source: Samsung PM1733 Product Brief (annotated). The flash memory in Samsung's Accused Instrumentalities comprises blocks (erase units). See: **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17.

Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP).

Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block.

Source: Boukhobza, at 22 (highlighted).

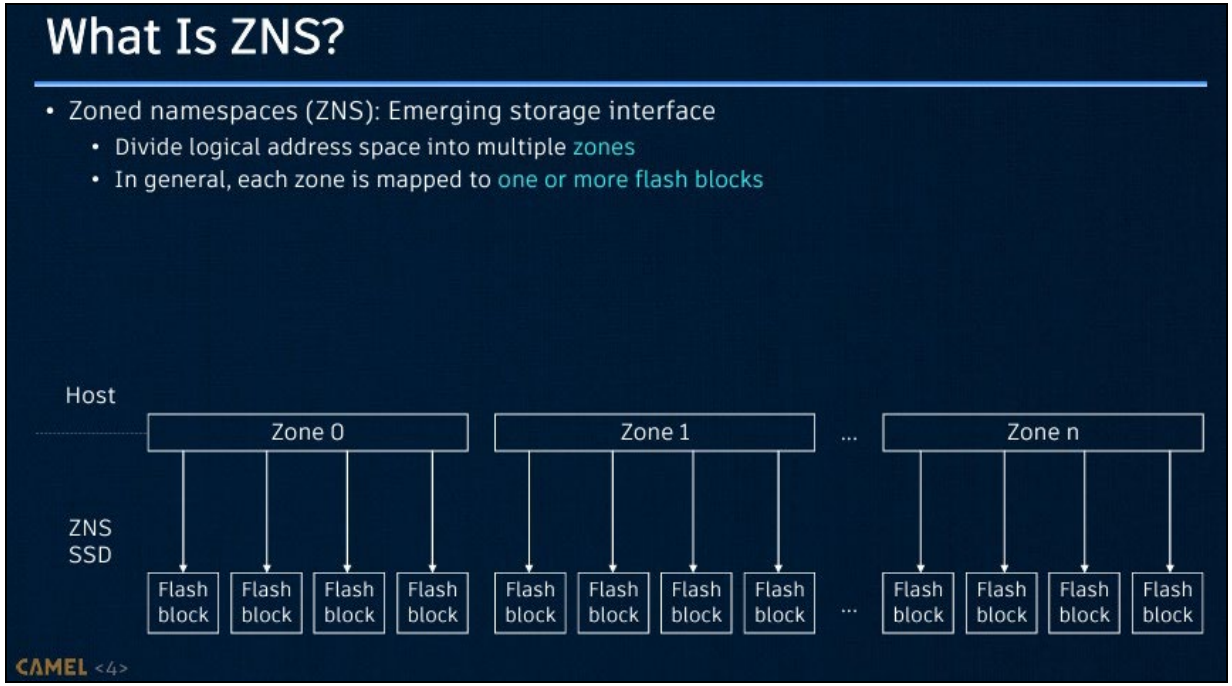
2.2.1. *Erase-before-write constraint*

Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of

Source: *Id.* at 24.

Samsung's Accused Instrumentalities are able to have zones (segments) at a hierarchically-intermediate level. Each zone corresponds to a mutually exclusive set of one or more erase units which are associated

with only one zone. Samsung's Accused Instrumentalities are able to addressable storage locations within each zone at a hierarchically-inferior level. See:



Source: Bae, at 4.

> This new attribute was introduced to allow for the zone size to remain a power of two number of logical blocks (facilitating logical block to zone number conversions) while allowing optimized mapping of a zone storage capacity to the underlying media characteristics. For instance, in the case a flash based device, a zone capacity can be aligned to the size of flash erase blocks without requiring that the device implements a power-of-two sized erased block.

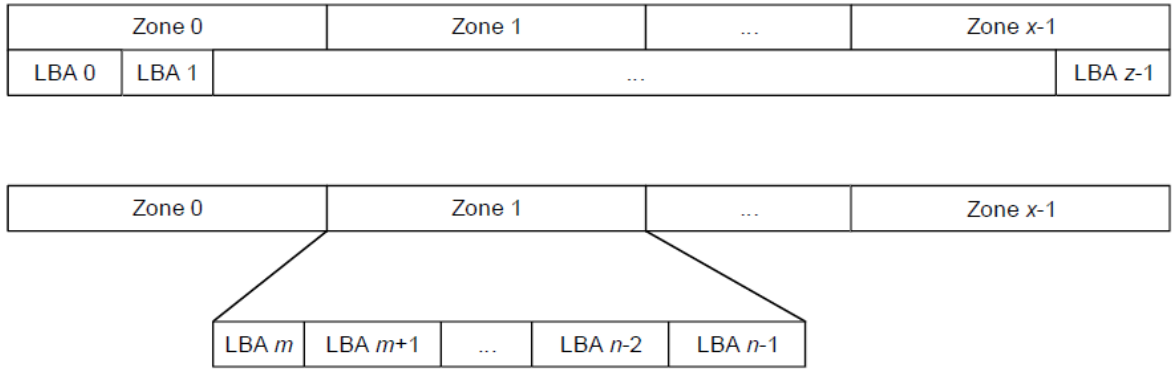
Source: Zoned Storage Introduction.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

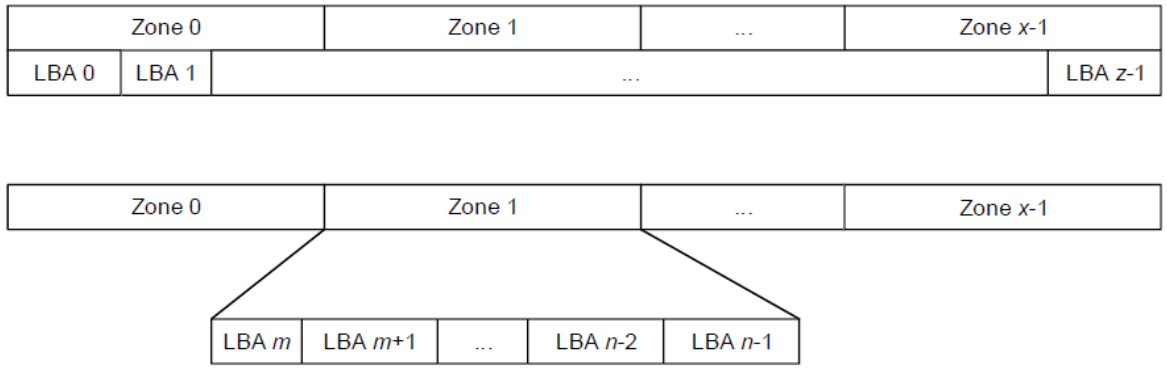
Samsung's Accused Instrumentalities are able to have one or more namespaces (block devices) at a hierarchically-superior level. Each zone is associated with only one of the namespaces. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.*

1.5.37 namespace

A formatted quantity of non-volatile memory that may be directly accessed by a host.

1.5.38 namespace ID (NSID)

An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier (refer to Figure 88). Refer to section 3.2.1 for the definitions of valid NSID, invalid NSID, active NSID, inactive NSID, allocated NSID, and unallocated NSID.

Source: NVMe Base Specification, at 22.

A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command. The Identify Namespace data structures (refer to section 1.5.30) indicate capabilities and settings that are specific to a particular namespace.

Source: NVMe Base Specification, at 28.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
21b	circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is	Samsung's Accused Instrumentalities comprise circuitry to receive from a host, and control execution of, incoming requests including data write requests and erase requests, wherein each of the incoming requests is accompanied by incoming address information. Samsung's Accused Instrumentalities are able to receive and execute a Write command from a host. A Write command is a data write request. A Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. See: **3.3.6 Write command** The Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. The host may also specify protection information to include as part of the operation.

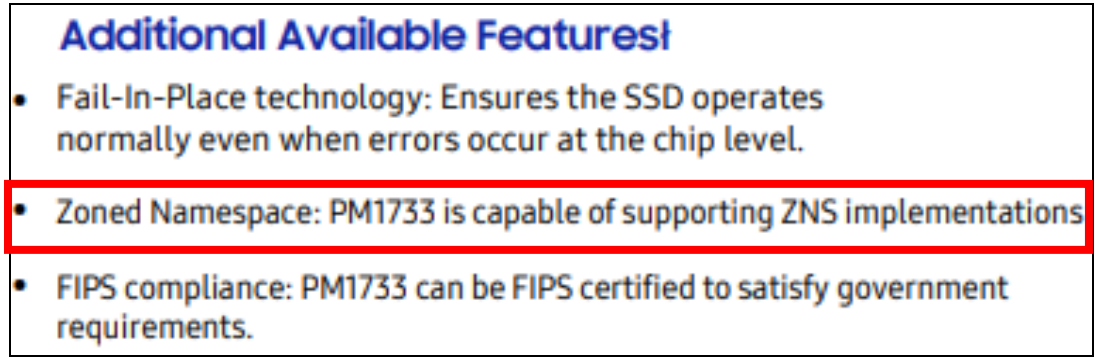
	accompanied by incoming address information;	Source: NVM Command Set Specification, at 53. The Write command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which data is to be written. The NSID and SLBA together are incoming address information. See:  Source: Samsung PM1733 Product Brief (annotated).

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVM Command Set Specification, at 107.

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2

Source: ZNS Specification, at 18 (annotated).

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

Samsung's Accused Instrumentalities are able to receive and execute a Zone Management Send command specifying a Reset Zone action (erase request) from the host. See:

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

Source: ZNS Specification, at 29.

3.4.3.1.4 Reset Zone

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

a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. The Zone Management Send command is accompanied by a Namespace Identifier ("NSID") that indicates the namespace to which the command applies, and a Starting LBA ("SLBA"), which indicates the starting storage location to which the Zone Management Send command applies. The NSID and SLBA together are incoming address information. See:

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVM Command Set Specification, at 107.

3.4.3 Zone Management Send command

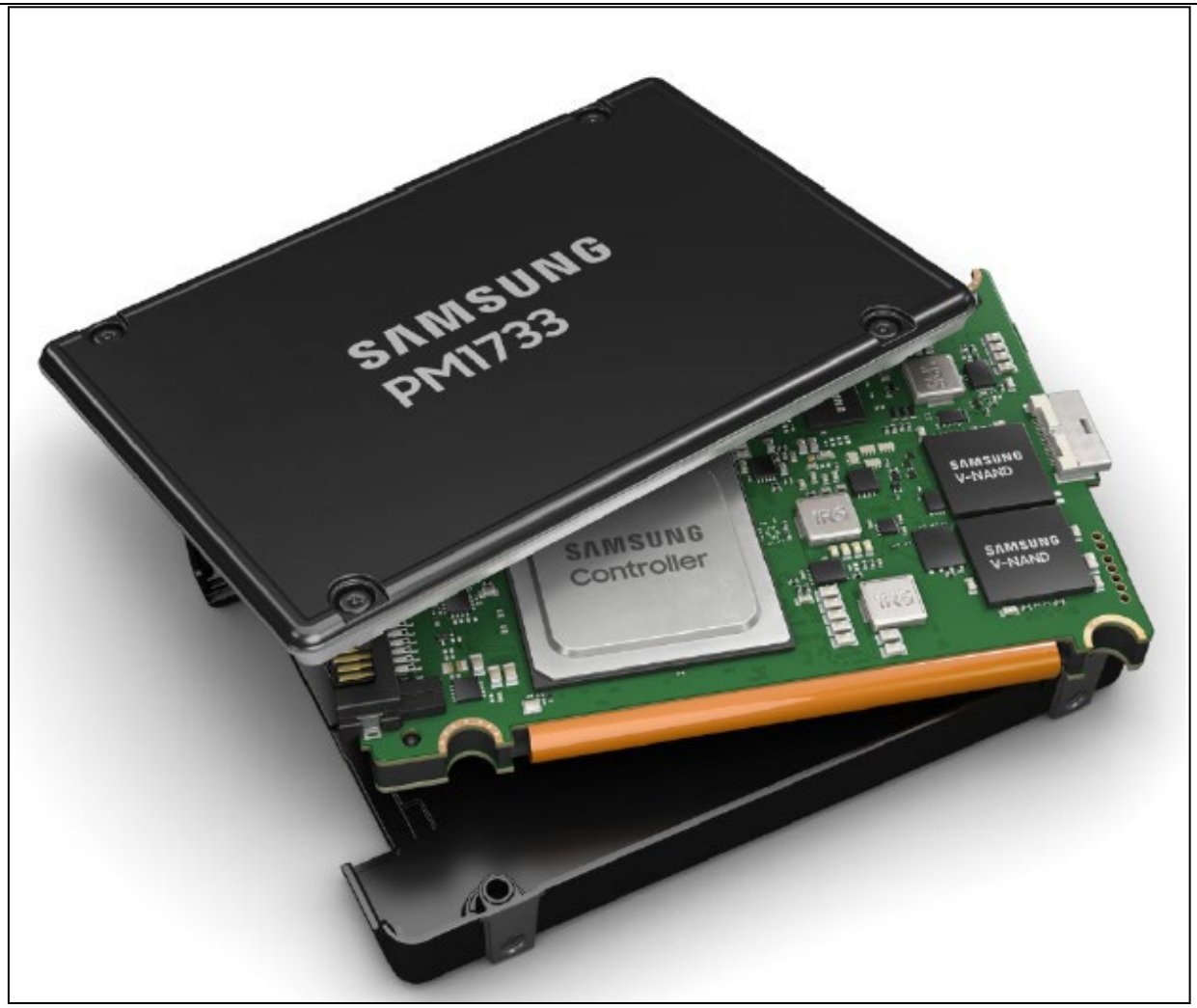
The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11, and Command Dword 13 fields. All other command specific fields are reserved.

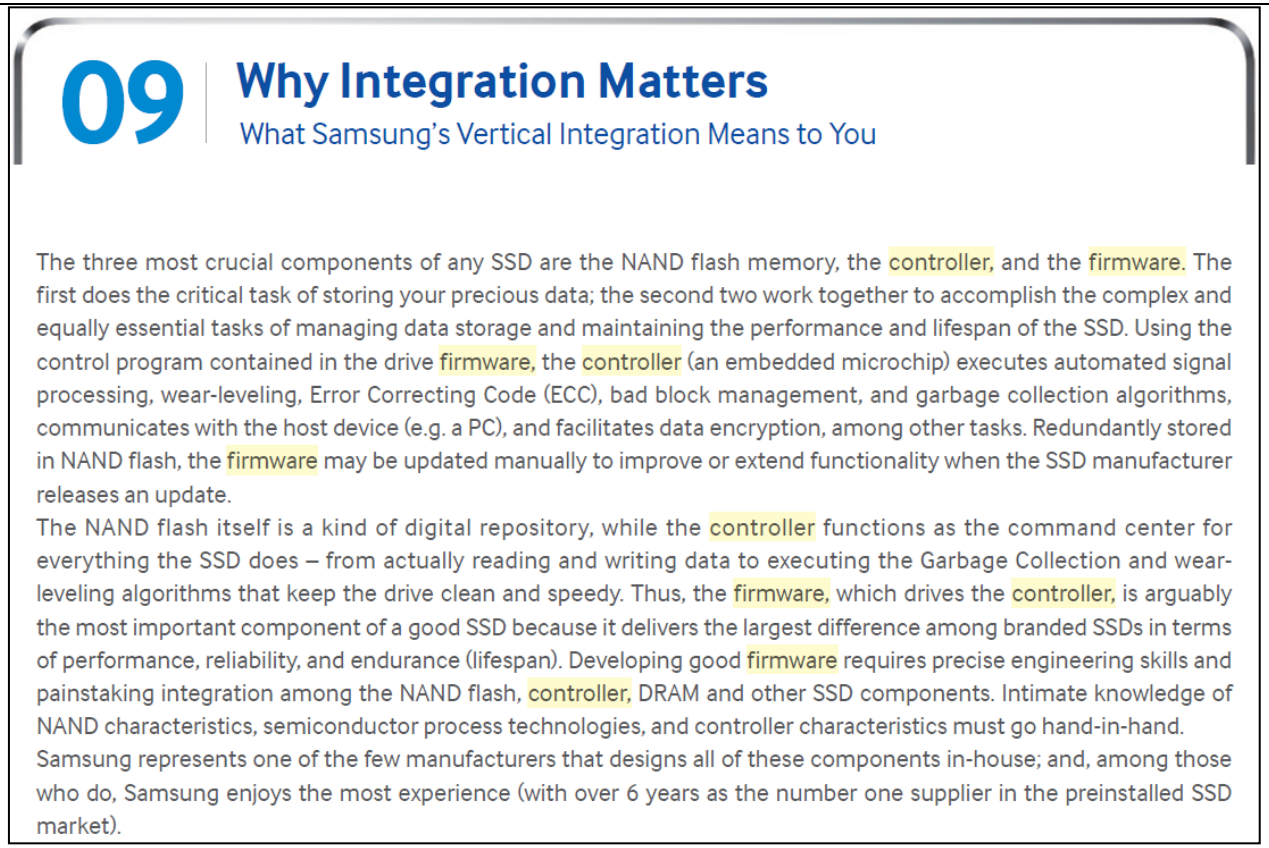
Figure 38: Zone Management Send – Command Dword 10 and Command Dword 11

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.

Source: ZNS Specification, at 29.

21c	wherein said circuitry is to, for each of	Samsung's Accused Instrumentalities comprise circuitry to, for each of the incoming requests, perform what is described in elements [20d] through [20g]. See below. Samsung's Accused Instrumentalities contain a processor or memory controller that executes firmware (circuitry). See:

	incoming requests:	 Source: Samsung PM1733 Product Brief.

		 09 Why Integration Matters What Samsung's Vertical Integration Means to You The three most crucial components of any SSD are the NAND flash memory, the controller, and the firmware. The first does the critical task of storing your precious data; the second two work together to accomplish the complex and equally essential tasks of managing data storage and maintaining the performance and lifespan of the SSD. Using the control program contained in the drive firmware, the controller (an embedded microchip) executes automated signal processing, wear-leveling, Error Correcting Code (ECC), bad block management, and garbage collection algorithms, communicates with the host device (e.g. a PC), and facilitates data encryption, among other tasks. Redundantly stored in NAND flash, the firmware may be updated manually to improve or extend functionality when the SSD manufacturer releases an update. The NAND flash itself is a kind of digital repository, while the controller functions as the command center for everything the SSD does – from actually reading and writing data to executing the Garbage Collection and wear-leveling algorithms that keep the drive clean and speedy. Thus, the firmware, which drives the controller, is arguably the most important component of a good SSD because it delivers the largest difference among branded SSDs in terms of performance, reliability, and endurance (lifespan). Developing good firmware requires precise engineering skills and painstaking integration among the NAND flash, controller, DRAM and other SSD components. Intimate knowledge of NAND characteristics, semiconductor process technologies, and controller characteristics must go hand-in-hand. Samsung represents one of the few manufacturers that designs all of these components in-house; and, among those who do, Samsung enjoys the most experience (with over 6 years as the number one supplier in the preinstalled SSD market). Source: Source: Samsung SSD White Paper, at 30.
21d	derive, from the accompanying incoming address information, a first address portion and a	Samsung's Accused Instrumentalities contain said circuitry to derive, from the accompanying incoming address information, a first address portion and a second address portion. Samsung's Accused Instrumentalities are able to derive a Namespace Identifier or NSID (a first address portion) and a starting LBA (a second address portion) from the accompanying incoming address information. See:

second address portion,

Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

Figure 88: Common Command Format

Bytes	Description
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified.

Source: NVMe Base Specification, at 107.

Figure 69: Write – Command Dword 10 and Command Dword 11

Bits	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.

Source: NVM Command Set Specification, at 54.

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.

Source: ZNS Specification, at 29.

21e	identify an addressed one of the block devices from the first address portion,	Samsung's Accused Instrumentalities comprise said circuitry to identify an addressed one of the block devices from the first address portion. Samsung's Accused Instrumentalities are able to identify an addressed namespace (block device) from the Namespace Identifier or NSID (the first address portion). See:

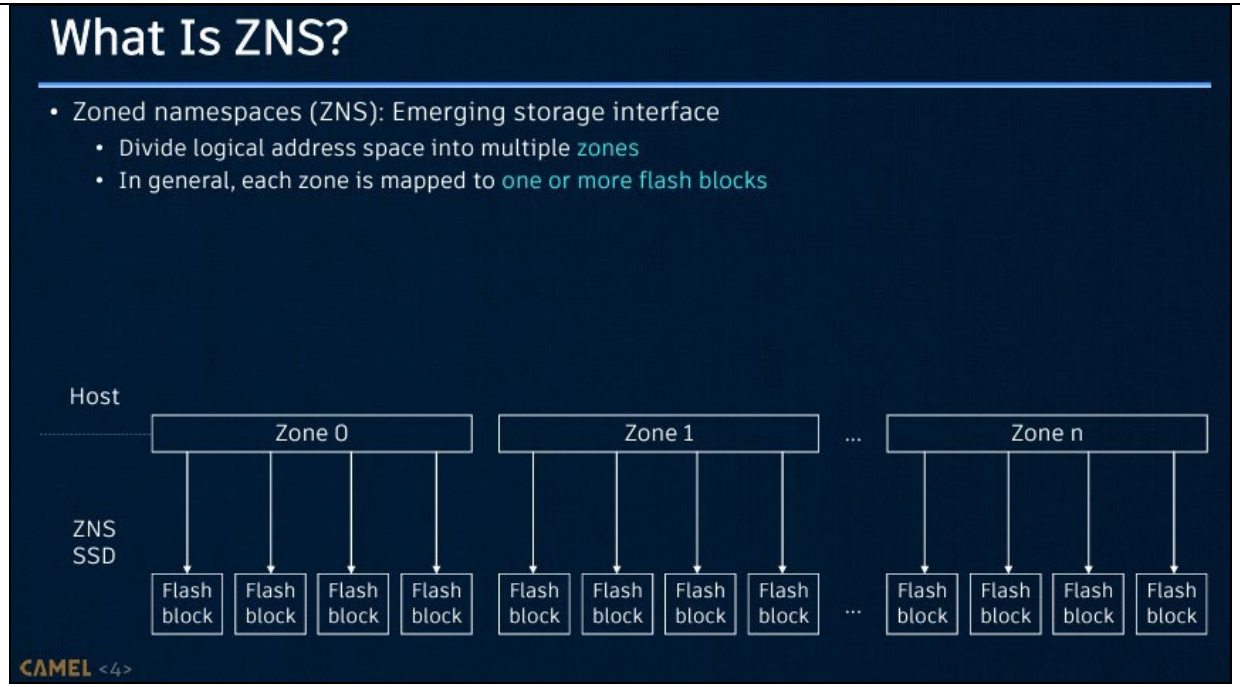
Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands

Opcode by Field			Combined Opcode [1]	Command [2]	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer [3]			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVM Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVM Command Set Specification				Write	3.3.7
Refer to the NVM Command Set Specification				Read	3.3.3
Refer to the NVM Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVM Command Set Specification				Compare	3.3.1
Refer to the NVM Command Set Specification				Write Zeroes	3.3.9
Refer to the NVM Command Set Specification				Verify	3.3.6
Refer to the NVM Command Set Specification				Copy	3.3.2
I/O commands defined in this specification					
0b	111 10b	01b	79h	Zone Management Send	3.4.3
0b	111 10b	10b	7Ah	Zone Management Receive	3.4.2
0b	111 11b	01b	7Dh	Zone Append	3.4.1

Notes:
1. Opcodes not listed are defined in the NVM Express Base Specification and in the NVM Command Set Specification.
2. All Zoned Namespace Command Set Commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

Source: ZNS Specification, at 18.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107.
21f	use a first operation to subdivide the second address portion into first bit fields, and to identify an addressed segment within the addressed block device from a first one of the first bit fields and a third address	Samsung's Accused Instrumentalities comprise said circuitry to use a first operation to subdivide the second address portion into first bit fields, and to identify an addressed segment within the addressed block device from a first one of the first bit fields and a third address portion from a second one of the first bit fields, wherein the addressed segment corresponds to a subset of the erase units of the flash memory. Samsung's Accused Instrumentalities contain zones (segments) that correspond to subsets of the erase units of the flash memory. See:

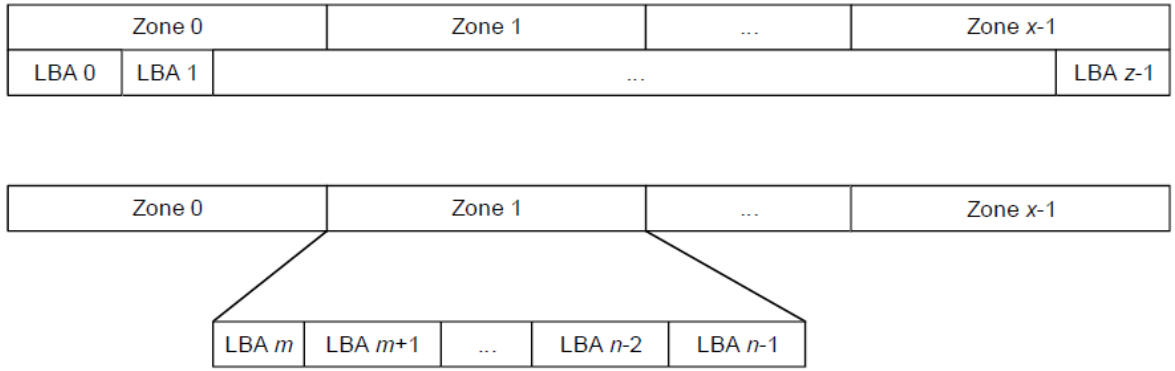
	portion from a second one of the first bit fields, wherein the addressed segment corresponds to a subset of the erase units of the flash memory, and	 Source: Bae, at 4. This new attribute was introduced to allow for the zone size to remain a power of two number of logical blocks (facilitating logical block to zone number conversions) while allowing optimized mapping of a zone storage capacity to the underlying media characteristics. For instance, in the case a flash based device, a zone capacity can be aligned to the size of flash erase blocks without requiring that the device implements a power-of-two sized erased block. Source: Zoned Storage Introduction.

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

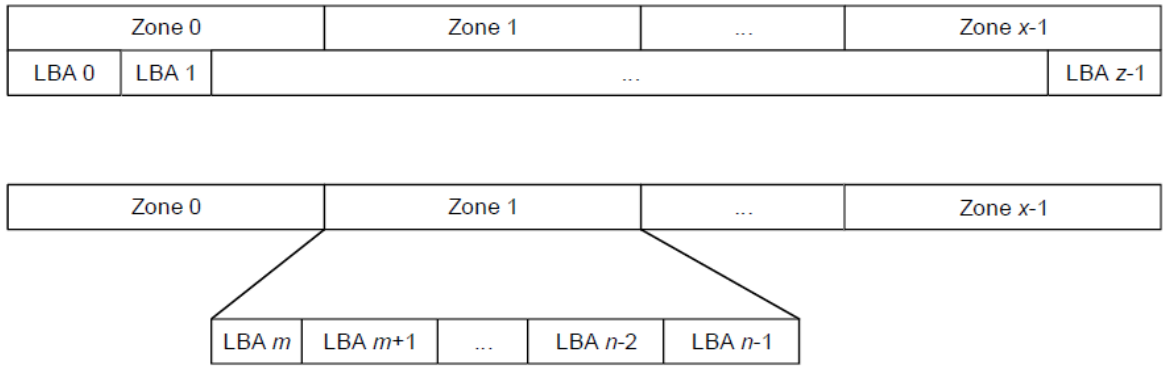
Samsung's Accused Instrumentalities are able to use a first operation on the starting LBA to subdivide it into bit fields and to identify an addressed zone (segment) within the addressed namespace (block device) from a first bit field. All zones are equally sized with contiguous non-overlapping ranges of LBAs. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.*

3.3.7 Write command

The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

3.3.7.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19.

Figure 19: Write – Command Specific Status Values

Value	Description
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.
BCh	**Zone Invalid Write:** The write to a zone was not at the write pointer.
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.

Source: *Id.* at 20.

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

If the command completes successfully, depending on the Zone Send Action field and the current states of the zones specified by the command, then that command may affect zones in various ways, including the following:

a) the zone state may change;
b) the Zone Descriptor Extension data may change; and
c) the Zone Descriptor Extension Valid zone attribute bit may change.

Source: *Id.* at 29.

Figure 40: Zone Management Send – Command Specific Status Values

Value	Description
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.

Source: *Id.* at 32.

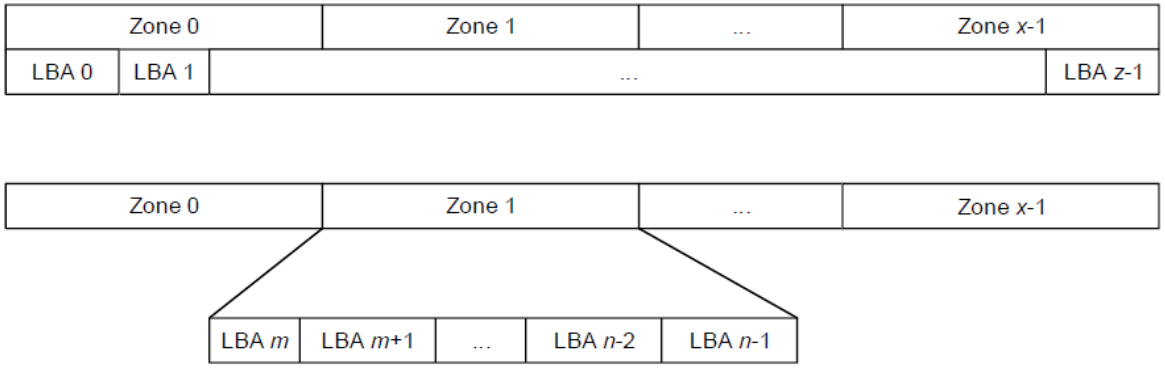
Samsung's Accused Instrumentalities are able to identify a third address portion from a second bit field. See:

2.1.1 Namespaces

A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification.

A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA.

Figure 2: Zones in a Zoned Namespace



The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: *Id.* at 8.

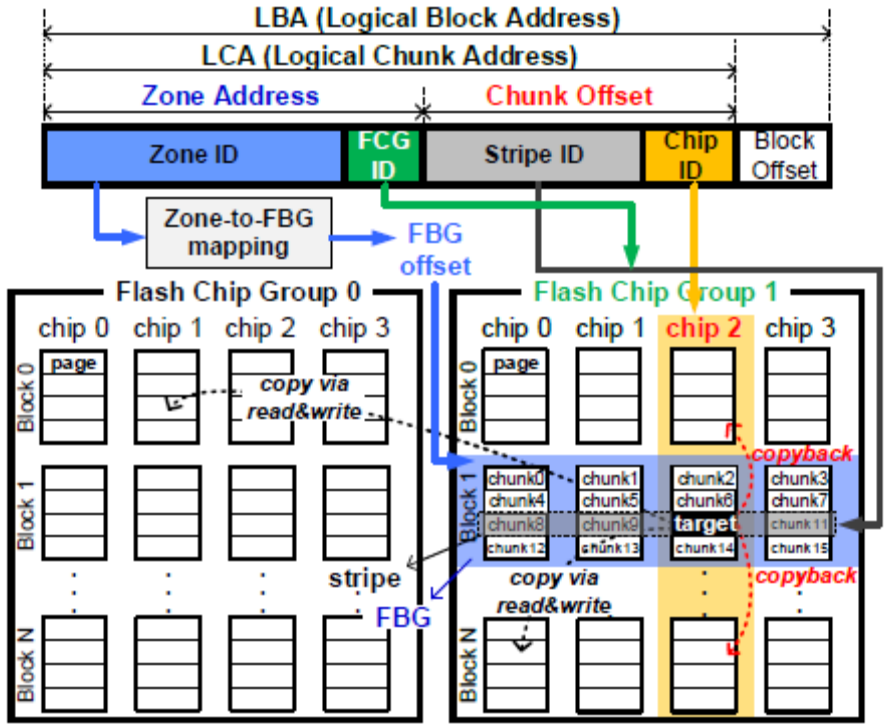
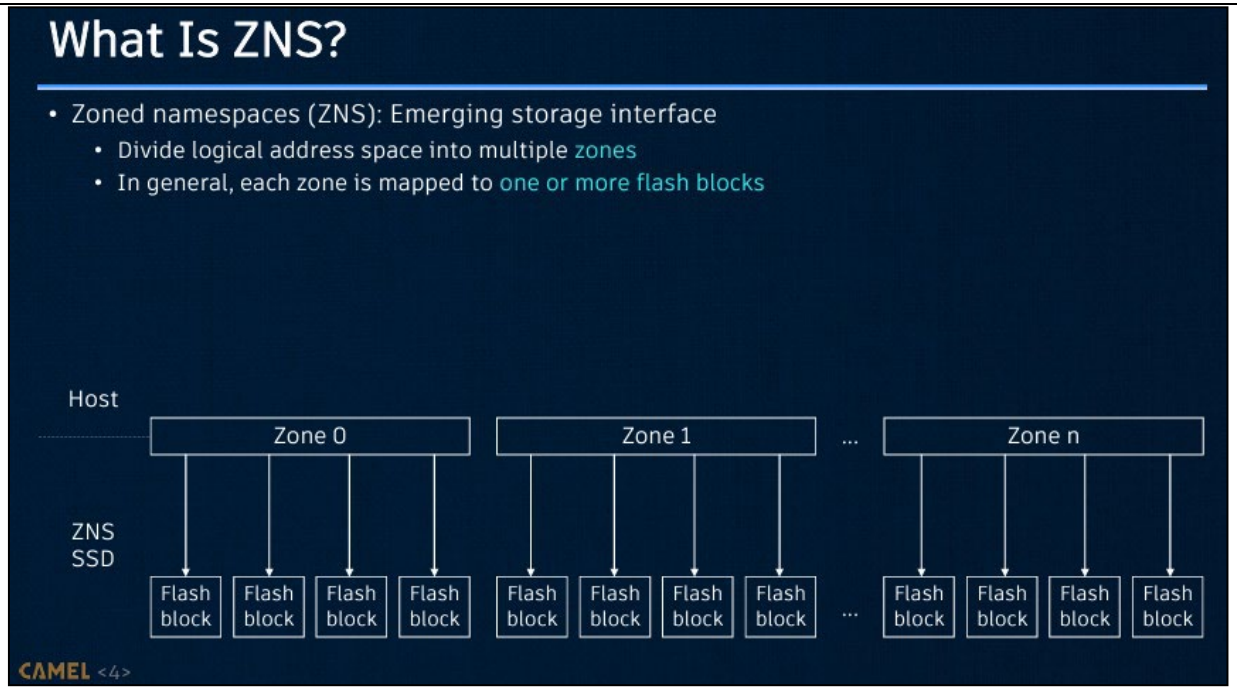
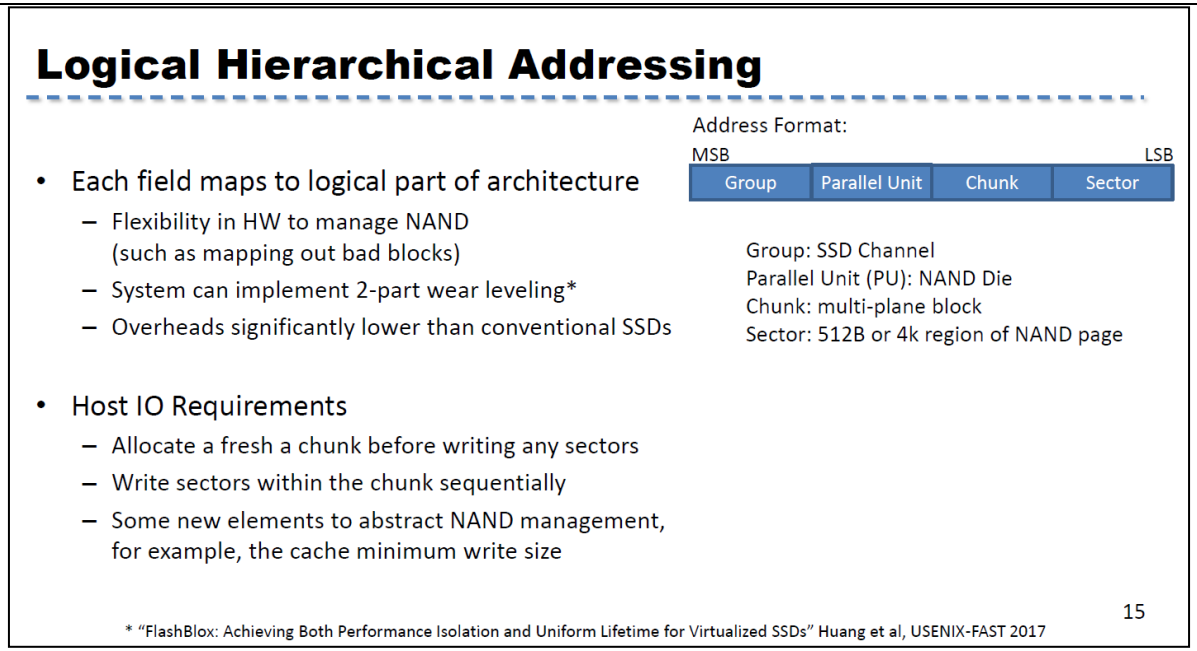


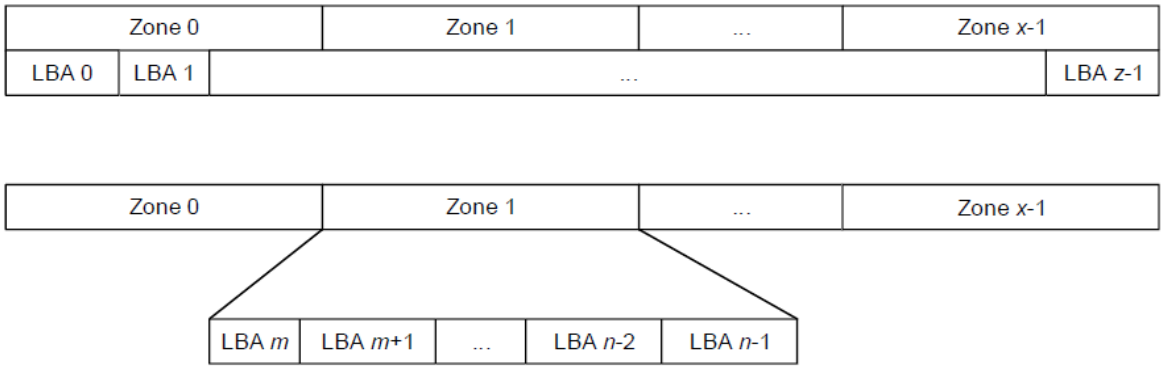
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

		 Source: Microsoft Denali Presentation, at 15.
21g	for each of the data write requests, use a second operation to subdivide the third address portion into second bit fields, to identify, from a first one of the second bit	Samsung's Accused Instrumentalities contain said circuitry to, for each of the data write requests, use a second operation to subdivide the third address portion into second bit fields, to identify, from a first one of the second bit fields, an address value to select one of the erase units and, from a second one of the second bit fields a storage location within the selected one of the erase units. Samsung's Accused Instrumentalities are able to, for each Write command, use a second bit field of the starting LBA as a third address portion. Samsung's Accused Instrumentalities are able to further subdivide the third address portion to identify an address value to select one of the erase units corresponding to the zone addressed by the first bit field, and to identify a storage location within such erase unit, using a second division operation. See:

	fields, an address value to select one of the erase units and, from a second one of the second bit fields a storage location within the selected one of the erase units;	**2.1.1 Namespaces** A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification. A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA. **Figure 2: Zones in a Zoned Namespace**  The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification). Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors. Source: ZNS Specification, at 8.

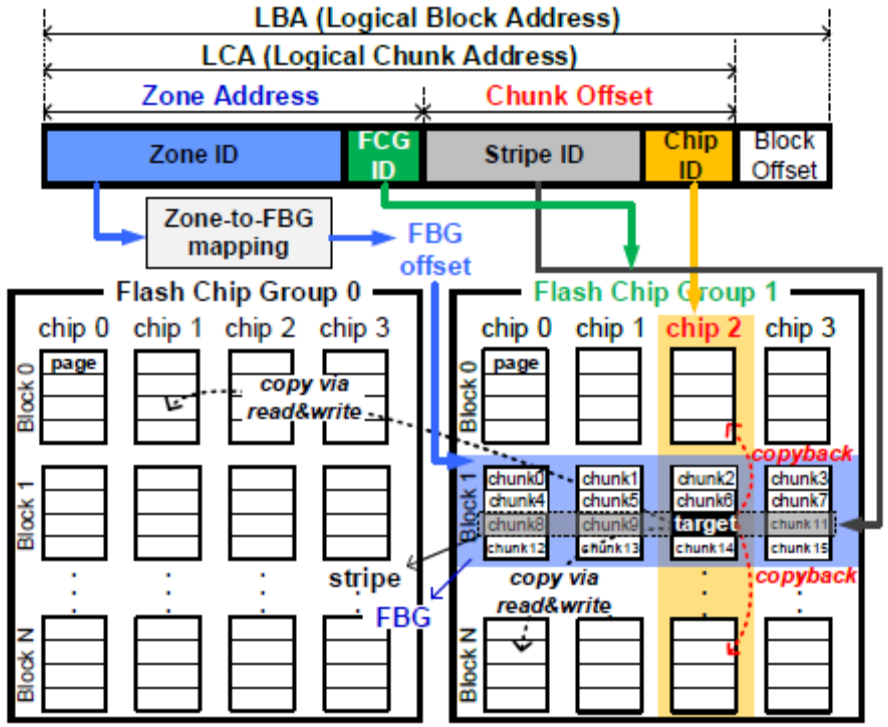
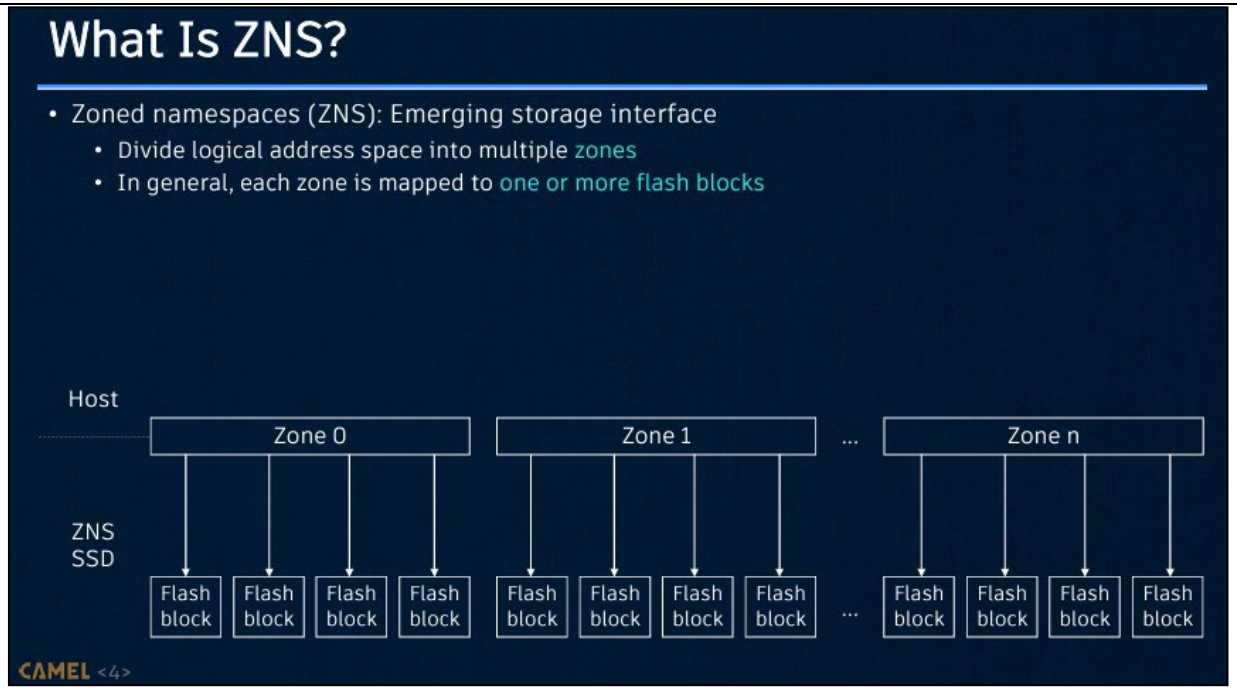
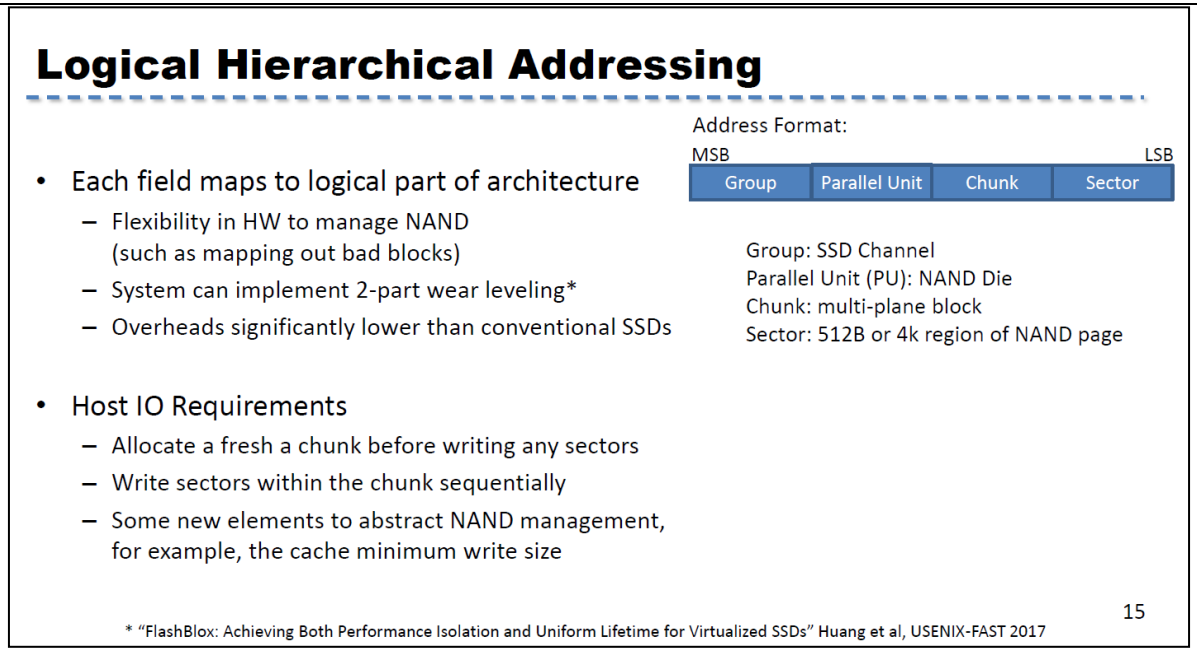


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.

		 Source: Microsoft Denali Presentation, at 15.
21h	wherein said circuitry is to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the	Samsung's Accused Instrumentalities comprise said circuitry to execute each one of the data write requests by programming accompanying data into the storage location that was identified for that one of the data write requests. Samsung's Accused Instrumentalities are able to execute data write requests by writing (programming) data into the identified storage location. See:

	data write requests;	**3.3.7 Write command** The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2). **3.3.7.1 Command Completion** Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19. **Figure 19: Write – Command Specific Status Values** **Value** / **Description** 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. BCh — **Zone Invalid Write:** The write to a zone was not at the write pointer. 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. Source: ZNS Specification, at 20.

		Figure 88: Common Command Format **Bytes** / **Description** 07:04 — **Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 3.2.1.2), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 140 and Figure 391 for commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 3.2.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 3.2.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with a status code of Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 140), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 1h to FFFFFFFFh, then the controller should abort the command with a status code of Invalid Field in Command, unless otherwise specified. Source: NVMe Base Specification, at 107. **Figure 69: Write – Command Dword 10 and Command Dword 11** **Bits** / **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. Source: NVM Command Set Specification, at 54.
21i	wherein said circuitry is also to execute each one of the incoming erase requests by controlling erasure in the flash memory	Samsung's Accused Instrumentalities contain said circuitry to execute each one of the incoming erase requests by controlling erasure in the flash memory of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests. Samsung's Accused Instrumentalities are able to execute the Zone Management Send command specifying a Reset Zone action (erase request) by controlling erasure of one or more of the erase units corresponding to the addressed zone. See:

	of one or more of the erase units in the subset corresponding to the addressed segment that was identified from the incoming address information accompanying the one of the incoming erase requests.	**Figure 39: Zone Management Send – Command Dword 13** **Bits** / **Description** 31:09 — Reserved 08 — **Select All:** If this bit is set to '1', then the SLBA field is 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. **Value** / **Description** / **Refer to section** 00h — Reserved 01h — **Close Zone:** Close one or more zones. — 3.4.3.1.1 02h — **Finish Zone:** Finish one or more zones. — 3.4.3.1.2 03h — **Open Zone:** Open one or more zones. — 3.4.3.1.3 04h — **Reset Zone:** Reset one or more zones. — 3.4.3.1.4 05h — **Offline Zone:** Offline one or more zones. — 3.4.3.1.5 06h to 0Fh — Reserved 10h — **Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone. — 3.4.3.1.6 11h to FFh — Reserved Source: ZNS Specification, at 29. **3.4.3.1.4 Reset Zone** If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the: a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;

b) ZSE:Empty state, then no change shall be made to the zone state; and
c) 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:

a) ZSIO:Implicity Opened state;
b) ZSEO:Explicitly Opened state;
c) ZSC:Closed state; and
d) the ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

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.

Source: *Id.* at 30–31.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

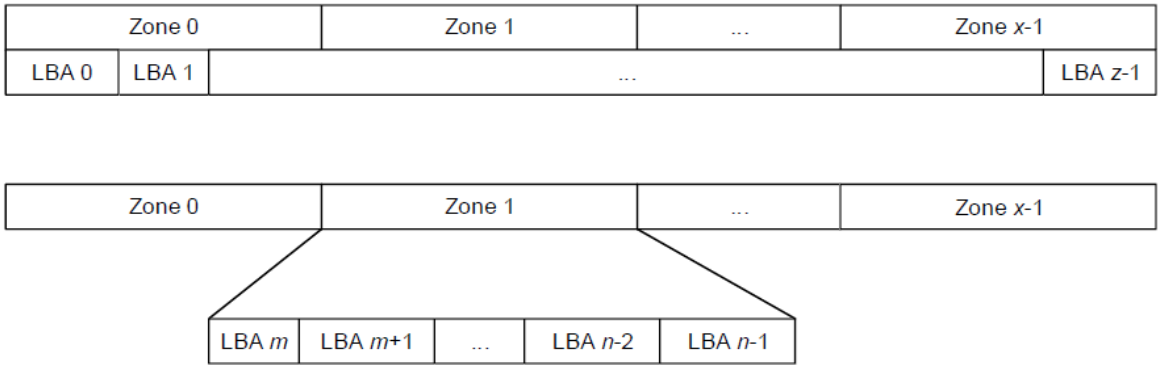
Source: ZNS Specification, at 42.

MB: Yes, George is right. What has been standardized today is that, for ZNS, you need to write sequentially. If you want to write to a specific page again, you need a "zone reset"—which erases the zone's contents and reverts the zone to an empty state—and then you can write to that page again. But inherently, with an SSD, you

Source: Bjørling.

		The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone. Source: ZNS Specification, at 10. 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; Source: *Id.* at 30–31. **Writing and Erasing NAND** Before we can understand the various maintenance algorithms that the controller employs to keep your SSD neat and tidy, we need to understand a few basics about how we read and write data to a NAND chip. Data is stored in a unit called a "page," which is finite in size and can only be written to when it is empty. Therefore, in order to write to a page that already contains data, it must first be erased. This would be a simple process, except each page belongs to a group of pages collectively known as a "block." While data is written in pages, it can only be erased in blocks. To help illustrate the concept Source: Samsung SSD White Paper, at 17. Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried out in pages while erase operations are executed in blocks. To alleviate this decrease in write performance, a process called garbage collection (GC) is implemented to create free blocks within the SSD. This technology secures free blocks by collecting valid pages into a single location and erasing the blocks consisting of invalid pages. However, this too may sometimes result in slower performance in the unexpected case that garbage collection interferes with the host write. Therefore, free space in the SSD is required to allow the firmware (FW) feature to run smoothly. This process in which extra space is allocated is called over-provisioning (OP). Source: Samsung Over-Provisioning White Paper, at 2 (highlighted).

		– *Erase operation*, which is applied at the level of an entire block's granularity. Therefore, the erase affects all the pages contained in a given block. Source: Boukhobza, at 22 (highlighted). **2.2.1. *Erase-before-write constraint*** Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (*in-place* update). Before performing any write operation on a page which already contains data, this page has to be erased. However, the erase operation affects an entire block, and not just a single page. Moreover, this operation has a significant latency. In the literature, this limitation is sometimes referred to under the name of Source: Boukhobza at 24 (highlighted).
22	The storage device of claim 1 wherein the second address portion comprises a first set of bits and a second set of bits, the bits of the first set of bits being mutually-exclusive to the bits of the second set of	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1. Samsung's Accused Instrumentalities comprise the storage device of claim 1 wherein the second address portion comprises a first set of bits and a second set of bits, the bits of the first set of bits being mutually-exclusive to the bits of the second set of bits, wherein the first division operation is to separate the first set of bits from the second set of bits, and the circuitry is to identify the addressed segment from the first set of bits. Samsung's Accused Instrumentalities are able to derive a starting LBA (second address portion) which comprises a first set of bits and a second set of bits which are mutually exclusive and to use the first division operation is to separate the first set of bits from the second set of bits. Samsung's Accused Instrumentalities are able to identify the addressed zone from the first set of bits. See:

	bits, wherein the first division operation is to separate the first set of bits from the second set of bits, and the circuitry is to identify the addressed segment from the first set of bits.	**2.1.1 Namespaces** A namespace is a collection of NVM and is defined in the NVM Express Base Specification and in the NVM Command Set Specification, as modified by this specification. A zoned namespace is a namespace that is associated with the Zoned Namespace Command Set. A zoned namespace is divided into a set of equally-sized zones, which are contiguous non-overlapping ranges of logical block addresses. Figure 2 shows a zoned namespace with *x* zones and *z* LBAs where LBA 0 is the lowest LBA of zone 0, LBA *z*-1 is the highest LBA of zone *x*-1, and for Zone 1, *m* is the lowest LBA and *n*-1 is its highest LBA. **Figure 2: Zones in a Zoned Namespace**  The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification). Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors. Source: ZNS Specification, at 8.

3.3.7 Write command

The Write command operates as defined in the NVM Command Set Specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.1.1.2).

3.3.7.1 Command Completion

Command Completion is as defined in the NVM Command Set Specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 19.

Figure 19: Write – Command Specific Status Values

Value	Description
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.
BCh	**Zone Invalid Write:** The write to a zone was not at the write pointer.
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.

Source: *Id.* at 20.

Figure 39: Zone Management Send – Command Dword 13

Bits	Description
31:09	Reserved
08	**Select All:** If this bit is set to '1', then the SLBA field is 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.

Value	Description	Refer to section
00h	Reserved	
01h	**Close Zone:** Close one or more zones.	3.4.3.1.1
02h	**Finish Zone:** Finish one or more zones.	3.4.3.1.2
03h	**Open Zone:** Open one or more zones.	3.4.3.1.3
04h	**Reset Zone:** Reset one or more zones.	3.4.3.1.4
05h	**Offline Zone:** Offline one or more zones.	3.4.3.1.5
06h to 0Fh	Reserved	
10h	**Set Zone Descriptor Extension:** Attach Zone Descriptor Extension data to a zone.	3.4.3.1.6
11h to FFh	Reserved	

If the command completes successfully, depending on the Zone Send Action field and the current states of the zones specified by the command, then that command may affect zones in various ways, including the following:

a) the zone state may change;
b) the Zone Descriptor Extension data may change; and
c) the Zone Descriptor Extension Valid zone attribute bit may change.

Source: *Id.* at 29.

Figure 40: Zone Management Send – Command Specific Status Values

Value	Description
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.

Source: *Id.* at 32.

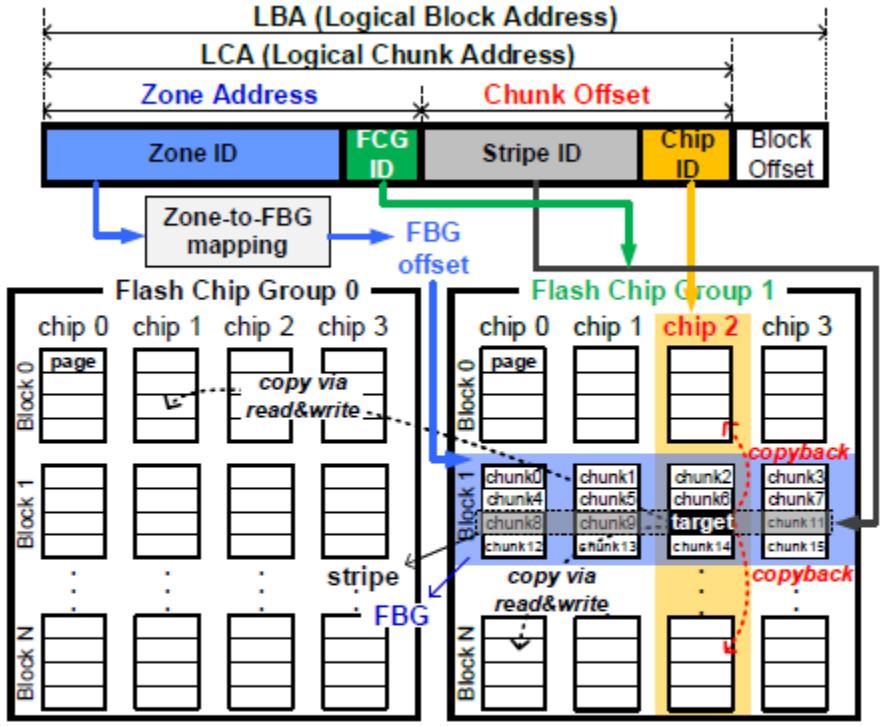
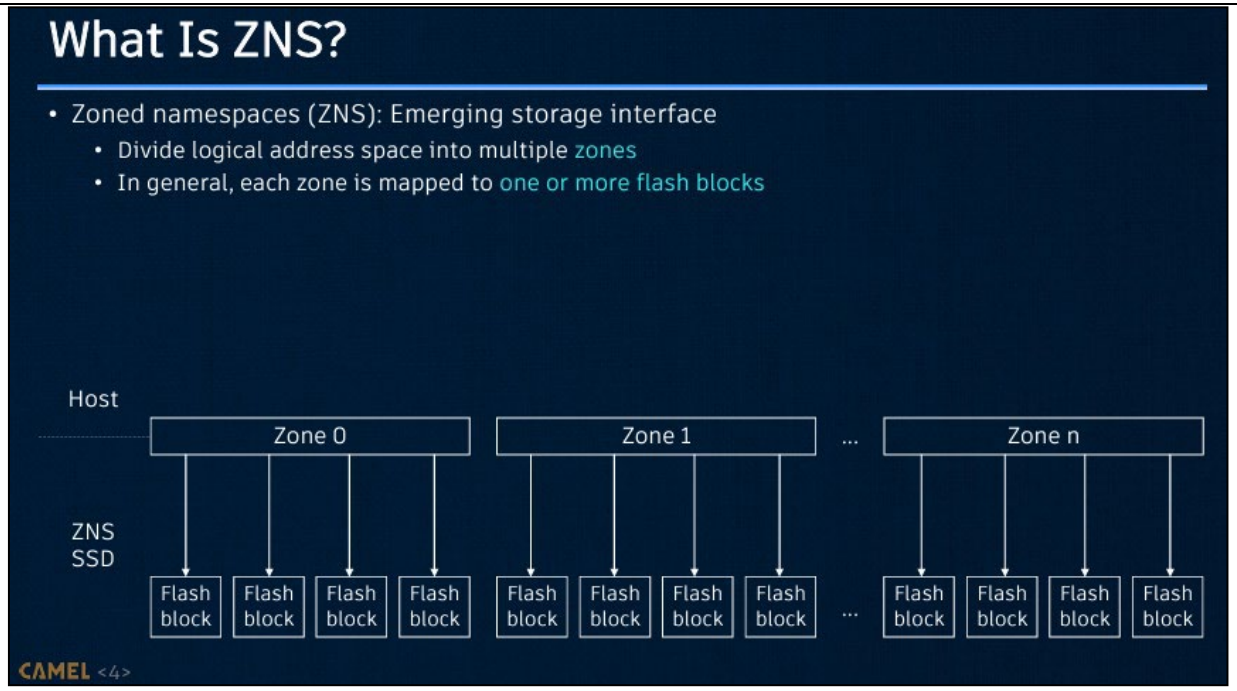
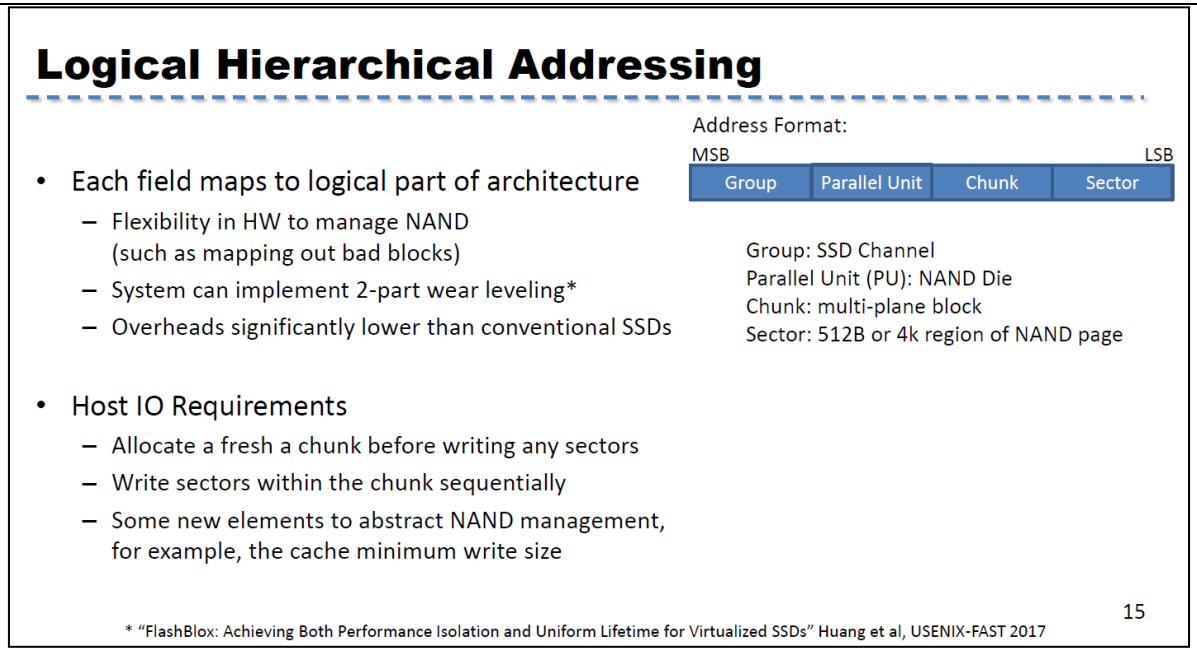


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.



Source: Bae, at 4.



Source: Microsoft Denali Presentation, at 15.