

Exhibit 6
U.S. Patent No. 11,347,656

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”).

No.	Claim Language	Samsung's Accused Instrumentalities												
<i>Ipre</i>	A storage device comprising:	To the extent the preamble is considered a limitation, Samsung's Accused Instrumentalities contain a storage device.												
<i>Ia</i>	flash memory having physical erase units; and	<p>Samsung's Accused Instrumentalities contain a flash memory having physical erase units.</p> <p>Samsung's Accused Instrumentalities contain NAND flash memory.</p> <p>For example, Samsung's PM1733 contains NAND flash memory. See:</p> <table><tr><th colspan="2">Samsung PM1733 specifications</th></tr><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> <p>Source: Samsung PM1733 Product Brief (annotated).</p> <p>The flash memory in Samsung's Accused Instrumentalities has physical blocks (erase units). See:</p>	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®
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®													

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.

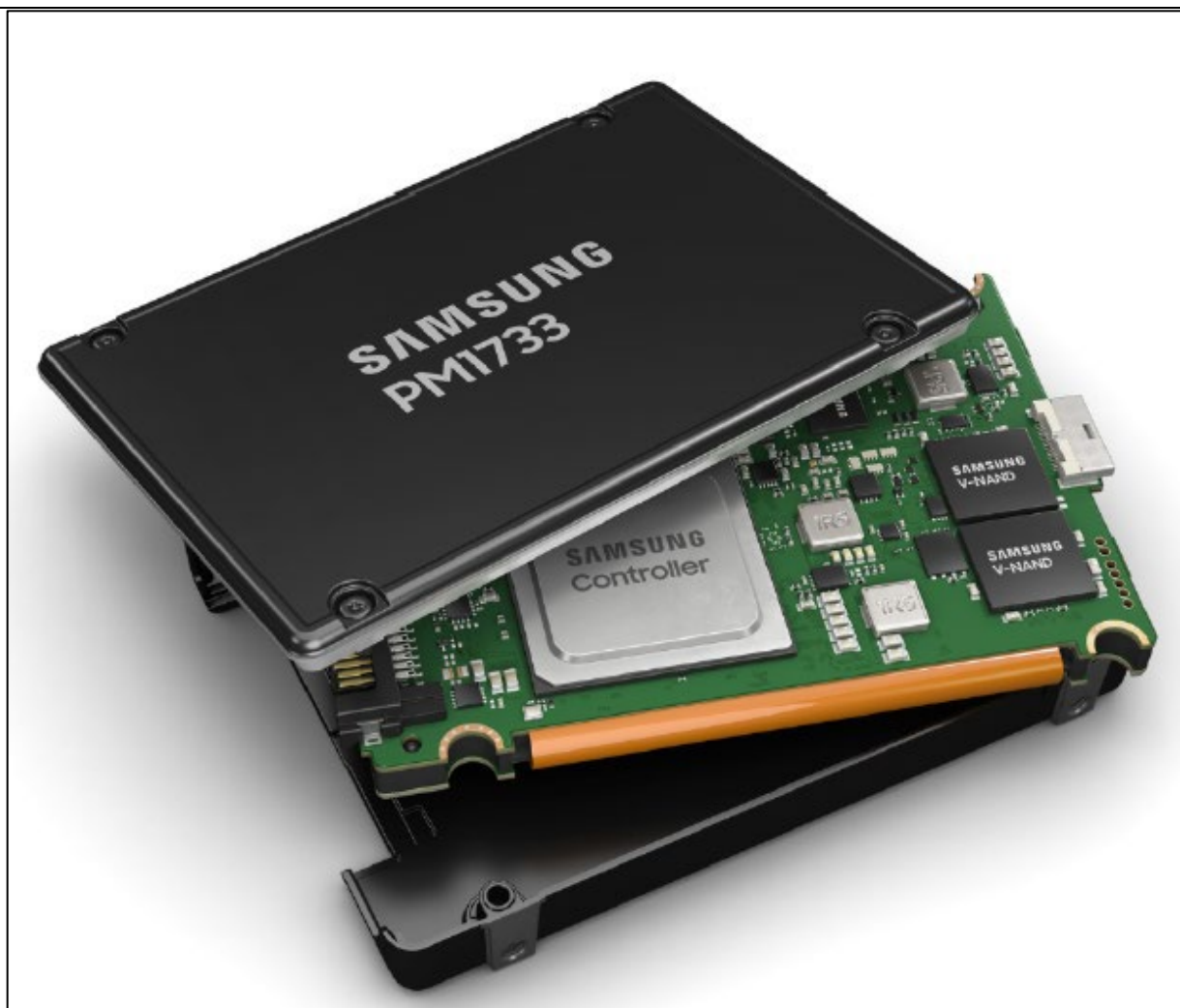
Since overwriting is impossible with NAND flash, existing data must first be erased in order to write new data to that cell, which slows down overall write performance of the SSD. Generally, it takes longer to erase data than to write it because, as mentioned previously, write operations are carried 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).

		<p>2.2.1. Erase-before-write constraint</p> <p>Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (<i>in-place</i> update). <u>Before performing any write operation on a page which already contains data, this page has to be erased.</u> However, the erase operation affects <u>an entire block</u>, 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</p> <p>Source: Boukhobza, at 24.</p>
<i>Ib</i>	circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:	<p>Samsung's Accused Instrumentalities contain circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information.</p> <p>Samsung's Accused Instrumentalities contain a processor or memory controller that executes firmware (circuitry). See:</p>



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.

Samsung's Accused Instrumentalities are able to receive and execute incoming Read commands from a host.

A Read command is a data access request.

A Read command reads data and metadata, if applicable, from the I/O controller for the logical block addresses (“LBAs”) indicated. See:

3.3.4 Read command

The Read command reads data and metadata, if applicable, from the I/O controller for the LBAs indicated. The command may specify protection information to be checked as part of the read operation.

Source: NVM Command Set Specification, at 48.

The Read 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 from which data is to be read. The NSID and SLBA together are accompanying address information. See:

Figure 88: Common Command Format

Bytes	Description
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Source: NVMe Base 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 NVMe Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVMe Express Base Specification				Reservation Release	NVM Express Base Specification
NVM Command Set commands implemented by this specification					
Refer to the NVMe Command Set Specification				Dataset Management	NVM Command Set Specification
NVM Command Set commands modified by this specification					
Refer to the NVMe Command Set Specification				Write	3.3.7
Refer to the NVMe Command Set Specification				Read	3.3.3
Refer to the NVMe Command Set Specification				Write Uncorrectable	3.3.8
Refer to the NVMe Command Set Specification				Compare	3.3.1
Refer to the NVMe Command Set Specification				Write Zeroes	3.3.9
Refer to the NVMe Command Set Specification				Verify	3.3.6
Refer to the NVMe Command Set Specification				Copy	3.3.2

Source: ZNS Specification, at 18 (annotated).

Figure 52: Read – 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 read as part of the operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63: 32.

Source: NVMe Command Set Specification at 49.

		<p>Samsung’s Accused Instrumentalities are able to receive and execute incoming Write commands from a host.</p> <p>A Write command is a data access request.</p> <p>A Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. See:</p> <div data-bbox="533 516 1864 667" style="border: 1px solid black; padding: 5px;"> <p>3.3.6 Write command</p> <p>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.</p> </div> <p>Source: NVM Command Set Specification at 53.</p> <p>The Write command 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 accompanying address information. See:</p>
--	--	--

		Figure 88: Common Command Format					
		<table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>	
Bytes	Description						
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>						
		Source: NVMe Base Specification, at 107.					

		<table><tr><th colspan="6">Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands</th></tr><tr><th colspan="3">Opcode by Field</th><th rowspan="3">Combined Opcode¹</th><th rowspan="3">Command²</th><th rowspan="3">Reference</th></tr><tr><th>(07)</th><th>(06:02)</th><th>(01:00)</th></tr><tr><th>Standard Command</th><th>Function</th><th>Data Transfer³</th></tr><tr><td colspan="4">Refer to the NVM Express Base Specification</td><td>Reservation Acquire</td><td>NVM Express Base Specification</td></tr><tr><td colspan="4">Refer to the NVM Express Base Specification</td><td>Reservation Release</td><td>NVM Express Base Specification</td></tr><tr><td colspan="6">NVM Command Set commands implemented by this specification</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Dataset Management</td><td>NVM Command Set Specification</td></tr><tr><td colspan="6">NVM Command Set commands modified by this specification</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Write</td><td>3.3.7</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Read</td><td>3.3.3</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Write Uncorrectable</td><td>3.3.8</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Compare</td><td>3.3.1</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Write Zeroes</td><td>3.3.9</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Verify</td><td>3.3.6</td></tr><tr><td colspan="4">Refer to the NVM Command Set Specification</td><td>Copy</td><td>3.3.2</td></tr></table>	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
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																																																																																							
		<p>Source: ZNS Specification, at 18 (annotated).</p>																																																																																										
		<table><tr><th colspan="2">Figure 69: Write – Command Dword 10 and Command Dword 11</th></tr><tr><th>Bits</th><th>Description</th></tr><tr><td>63:00</td><td>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.</td></tr></table>	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.																																																																																				
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.																																																																																											
		<p>Source: NVM Command Set Specification at 54.</p>																																																																																										
Ic	derive, from the	Samsung’s Accused Instrumentalities comprise circuitry to derive, from the accompanying address information, a first address portion, a second address portion and a third address portion.																																																																																										

	<p>accompanying address information, a first address portion, a second address portion and a third address portion,</p>	<p>Samsung's Accused Instrumentalities are able to derive, from the accompanying address information, a Namespace Identifier or NSID (a first address portion), a zone identifier information (a second address portion), and an LBA and/or a zone offset (a third address portion). See:</p> <div data-bbox="535 334 1711 717"> <p>Additional Available Features</p> <ul style="list-style-type: none"> ▪ Fail-In-Place technology: Ensures the SSD operates normally even when errors occur at the chip level. ▪ Zoned Namespace: PM1733 is capable of supporting ZNS implementations ▪ FIPS compliance: PM1733 can be FIPS certified to satisfy government requirements. </div> <p>Source: Samsung PM1733 Product Brief (annotated).</p>
--	---	--

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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Source: NVMe Base Specification, at 107.

Figure 52: Read – Command Dword 10 and Command Dword 11

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

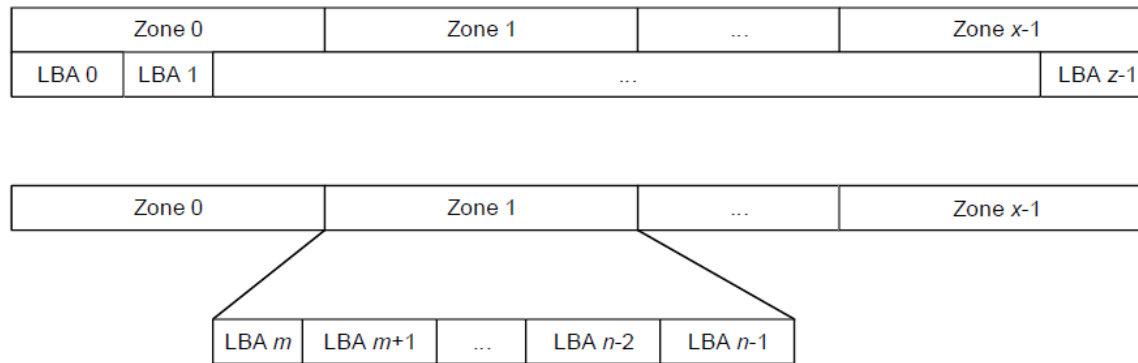
Source: NVM Command Set Specification, at 49.

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.

3.3.5 Read command

The Read 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).

3.3.5.1 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 17.

Figure 17: Read – Command Specific Status Values

Value	Description
B8h	Zone Boundary Error: The command specifies logical blocks in more than one zone.
BBh	Zone Is Offline: The accessed zone is in the ZSO:Offline state.

Source: ZNS Specification, at 20.

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.

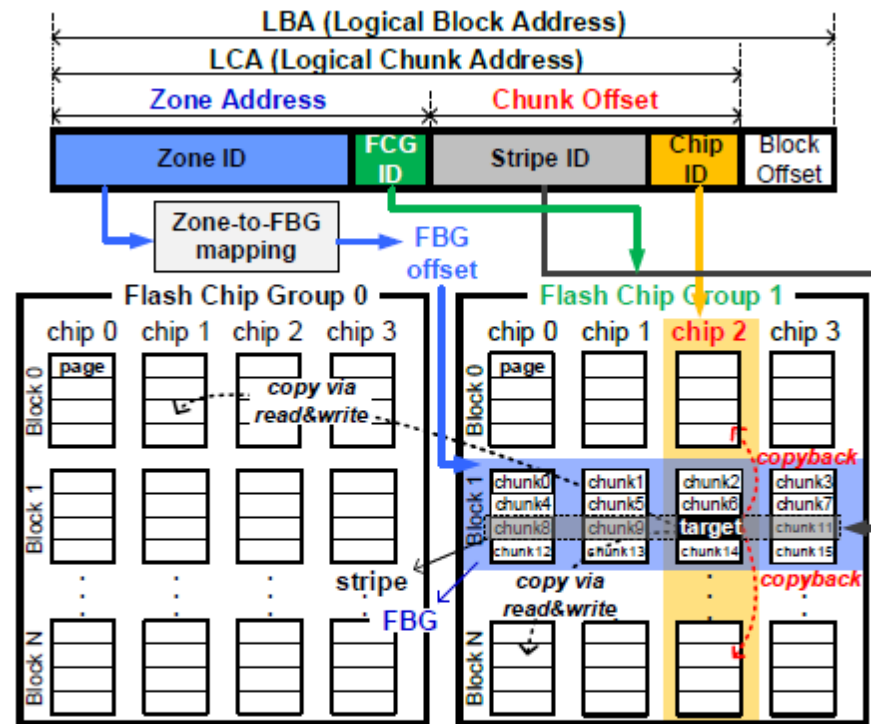
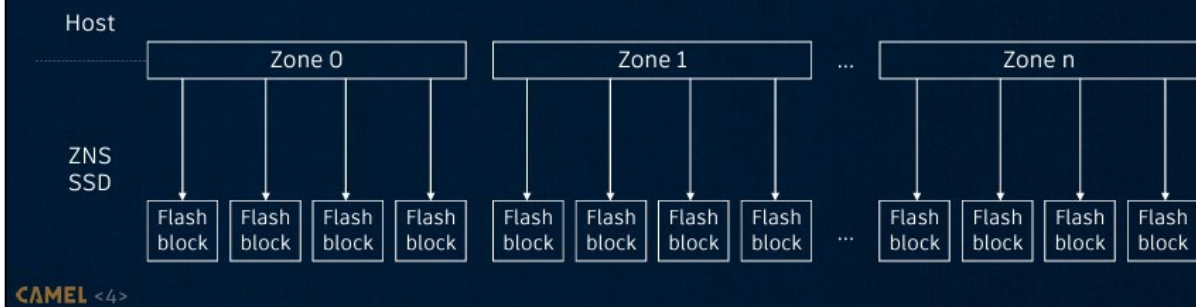


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



Source: Bae, at 4.

		<div data-bbox="537 191 1711 833"> <h2>Logical Hierarchical Addressing</h2> <hr/> <div> <div> <ul style="list-style-type: none"> Each field maps to logical part of architecture <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 </div> <div> <p>Address Format:</p> <div> <div>MSB</div> <div>Group</div> <div>Parallel Unit</div> <div>Chunk</div> <div>Sector</div> <div>LSB</div> </div> <p>Group: SSD Channel Parallel Unit (PU): NAND Die Chunk: multi-plane block Sector: 512B or 4k region of NAND page</p> </div> </div> <p>* "FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs" Huang et al, USENIX-FAST 2017</p> <p>15</p> </div> <p>Source: Microsoft Denali Presentation, at 15.</p>
<i>Id</i>	identify an addressed block device from the first address portion,	<p>Samsung's Accused Instrumentalities comprise circuitry to identify an addressed block device from the first address portion.</p> <p>Samsung's Accused Instrumentalities are able to identify an addressed namespace (block device) from the Namespace Identifier or NSID (first address portion). See:</p>

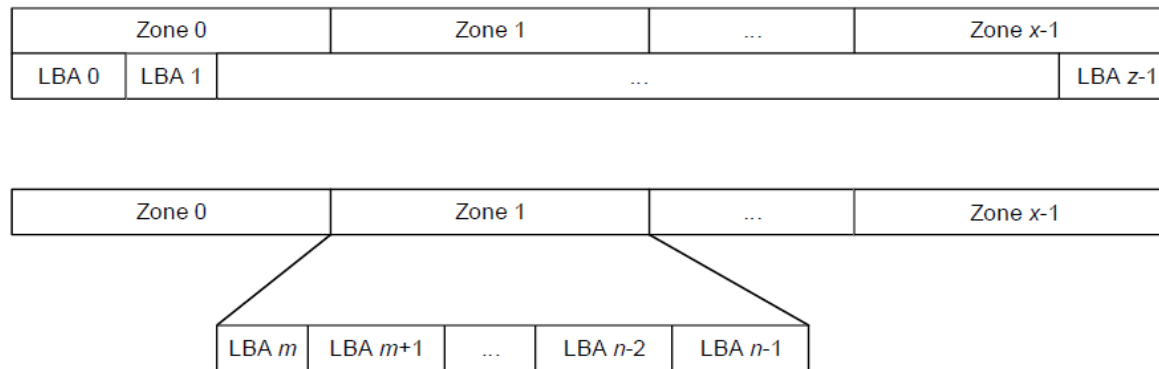
		<div><div><div><div><div><div></div><div>Figure 88: Common Command Format</div></div></div><table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>Namespace Identifier (NSID): This field specifies the namespace that this command applies to. If the namespace identifier is not used for 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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table></div><div>Source: NVMe Base Specification, at 107.</div></div></div>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Bytes	Description					
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>					
le	identify an addressed segment within the addressed block device from the second address portion, and	<div>Samsung’s Accused Instrumentalities comprise circuitry to identify an addressed segment within the addressed block device from the second address portion.</div> <div>Samsung’s Accused Instrumentalities are able to identify an addressed zone (segment) within the addressed namespace from the zone identifier information (second address portion). See:</div>				

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.

3.3.5 Read command

The Read 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).

3.3.5.1 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 17.

Figure 17: Read – Command Specific Status Values

Value	Description
B8h	Zone Boundary Error: The command specifies logical blocks in more than one zone.
BBh	Zone Is Offline: The accessed zone is in the ZSO:Offline state.

Source: ZNS Specification, at 20.

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.

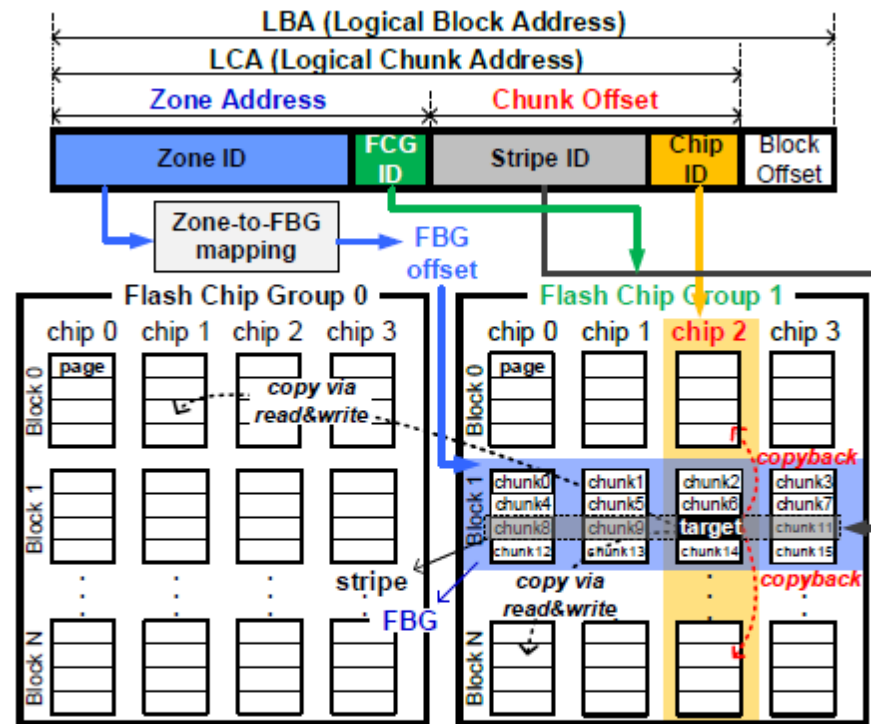
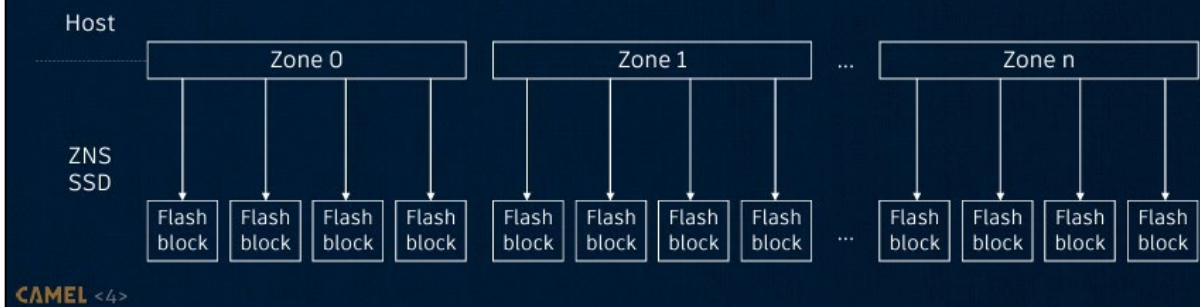


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



Source: Bae, at 4.

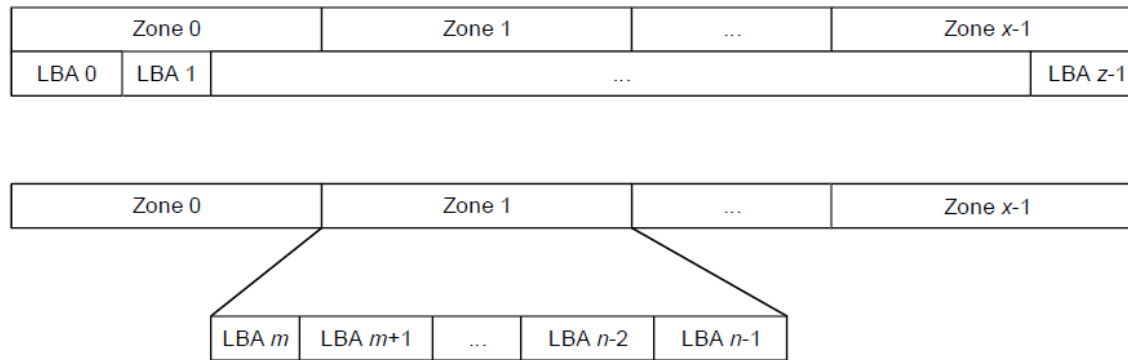
		<div data-bbox="537 191 1711 833"> <h2>Logical Hierarchical Addressing</h2> <hr/> <div> <div> <ul style="list-style-type: none"> Each field maps to logical part of architecture <ul style="list-style-type: none"> Flexibility in HW to manage NAND (such as mapping out bad blocks) System can implement 2-part wear leveling* Overheads significantly lower than conventional SSDs Host IO Requirements <ul style="list-style-type: none"> 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 </div> <div> <p>Address Format:</p> <div> <div>MSB</div> <div>Group</div> <div>Parallel Unit</div> <div>Chunk</div> <div>Sector</div> <div>LSB</div> </div> <p>Group: SSD Channel Parallel Unit (PU): NAND Die Chunk: multi-plane block Sector: 512B or 4k region of NAND page</p> </div> </div> <p>* "FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs" Huang et al, USENIX-FAST 2017</p> <p>15</p> </div> <p>Source: Microsoft Denali Presentation, at 15.</p>
<i>If</i>	identify an addressed storage location within the addressed segment from the third address portion,	<p>Samsung's Accused Instrumentalities comprise circuitry to identify an addressed storage location within the addressed segment from the third address portion.</p> <p>Samsung's Accused Instrumentalities are able to identify an addressed storage location, including a logical block, page, and/or erase unit, (addressed storage location) within the addressed zone (addressed segment) from the LBA and/or the zone offset (the third address portion). See:</p>

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.

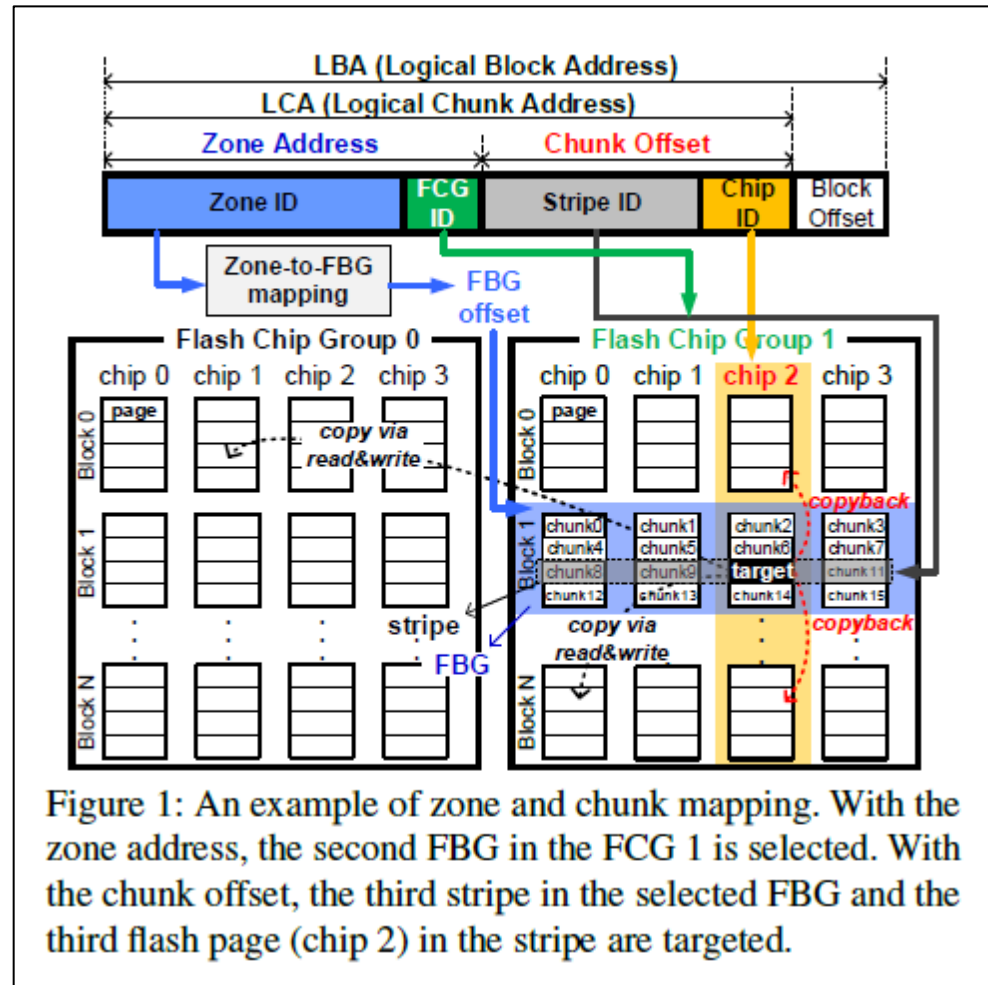
1.4.3.3 logical block

The smallest addressable data unit for Read and Write commands.

1.4.3.4 logical block address (LBA)

The address of a logical block, referred to commonly as LBA.

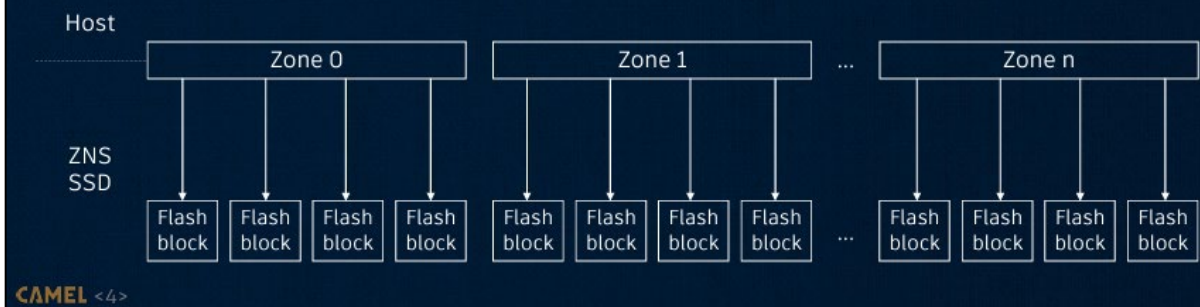
Source: NVM Command Set Specification, at 10.



Source: Han, at 149.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



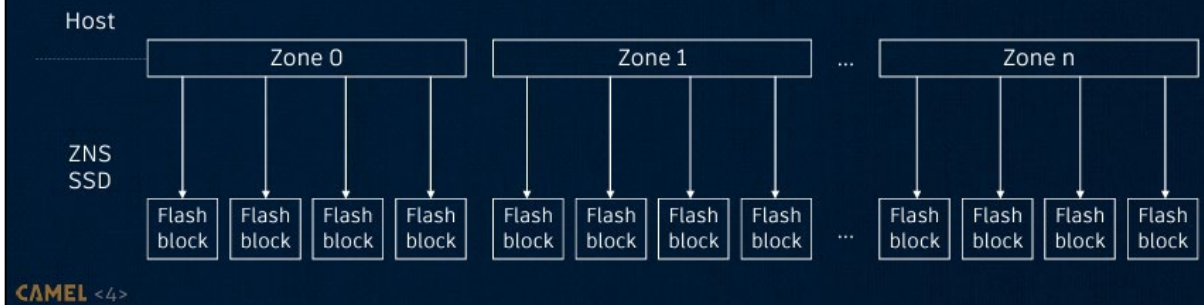
Source: Bae, at 4.

		<div><div><h2>Logical Hierarchical Addressing</h2><div><div><ul style="list-style-type: none">Each field maps to logical part of architecture<ul style="list-style-type: none">Flexibility in HW to manage NAND (such as mapping out bad blocks)System can implement 2-part wear leveling*Overheads significantly lower than conventional SSDsHost IO Requirements<ul style="list-style-type: none">Allocate a fresh a chunk before writing any sectorsWrite sectors within the chunk sequentiallySome new elements to abstract NAND management, for example, the cache minimum write size</div><div><div>Address Format:</div><div><div>MSB</div><div>Group</div><div>Parallel Unit</div><div>Chunk</div><div>Sector</div><div>LSB</div></div><div><div>Group: SSD Channel</div><div>Parallel Unit (PU): NAND Die</div><div>Chunk: multi-plane block</div><div>Sector: 512B or 4k region of NAND page</div></div></div></div><div><div>* "FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs" Huang et al, USENIX-FAST 2017</div><div>15</div></div></div><div>Source: Microsoft Denali Presentation, at 15.</div></div>
Ig	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,	<div><div><p>Samsung’s Accused Instrumentalities comprise circuitry to use a division operation to identify a logical erase unit within the addressed segment.</p><p>Samsung’s Accused Instrumentalities are able to use a division operation on at least one of the incoming address information or information derived therefrom to identify a logical erase unit within the addressed zone. Samsung’s Accused 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.</p><p>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:</p></div></div>

See:

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

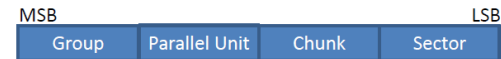


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

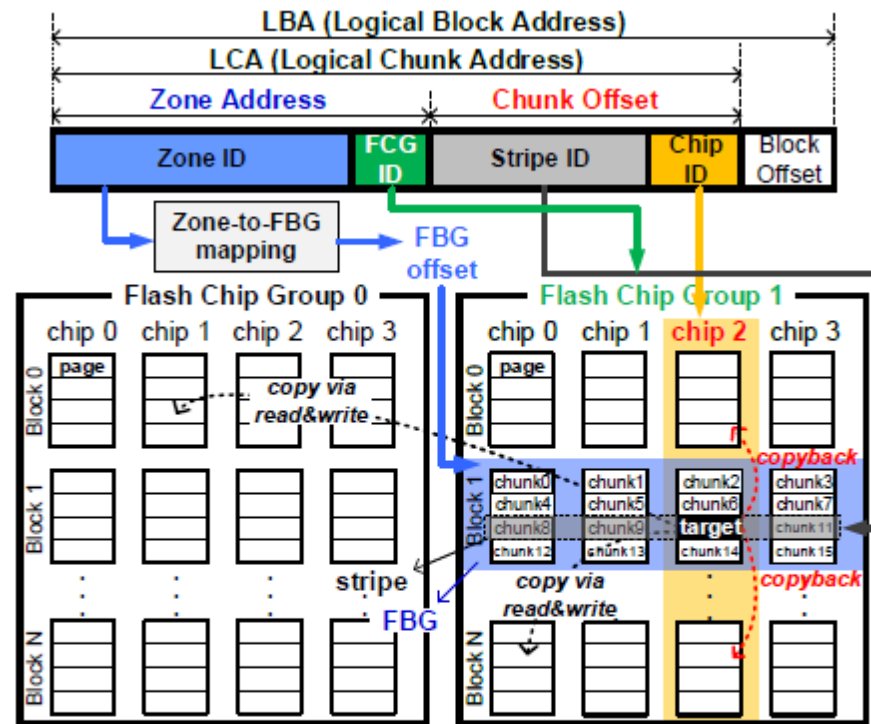


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

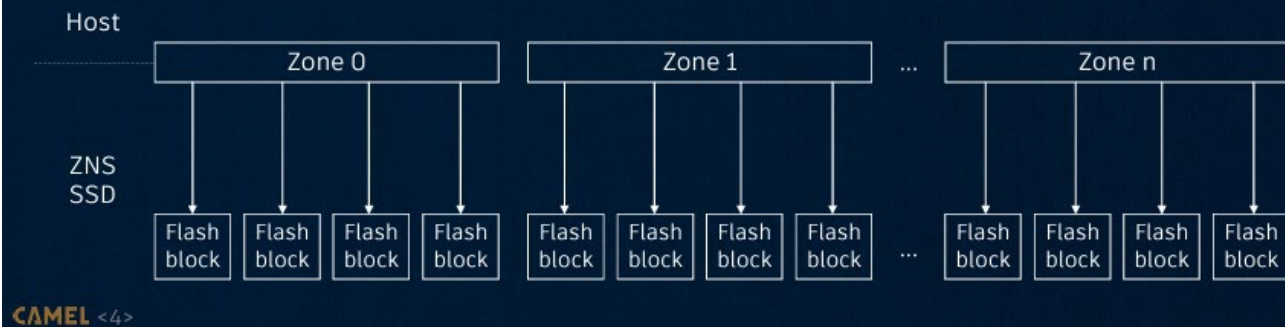
Source: Han, at 149.

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

		<p>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.</p> <p>As an example, there is no substantial difference between (1) identifying a logical erase unit within the addressed segment and (2) identifying a logical erase unit group within the addressed segment 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.</p> <p>As another example, there is no substantial difference between (1) identifying a logical erase unit within the addressed segment and (2) having the addressed segment as a logical erase unit group and 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.</p>
<i>1h</i>	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;	<p>Samsung's Accused Instrumentalities comprise circuitry to identify one of the physical erase units that corresponds to the identified logical erase unit.</p> <p>Samsung's Accused Instrumentalities are able to identify one of the physical blocks (erase units) that corresponds to the identified logical erase unit. 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:</p>

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

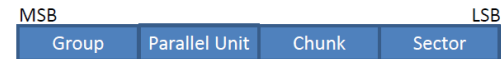


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

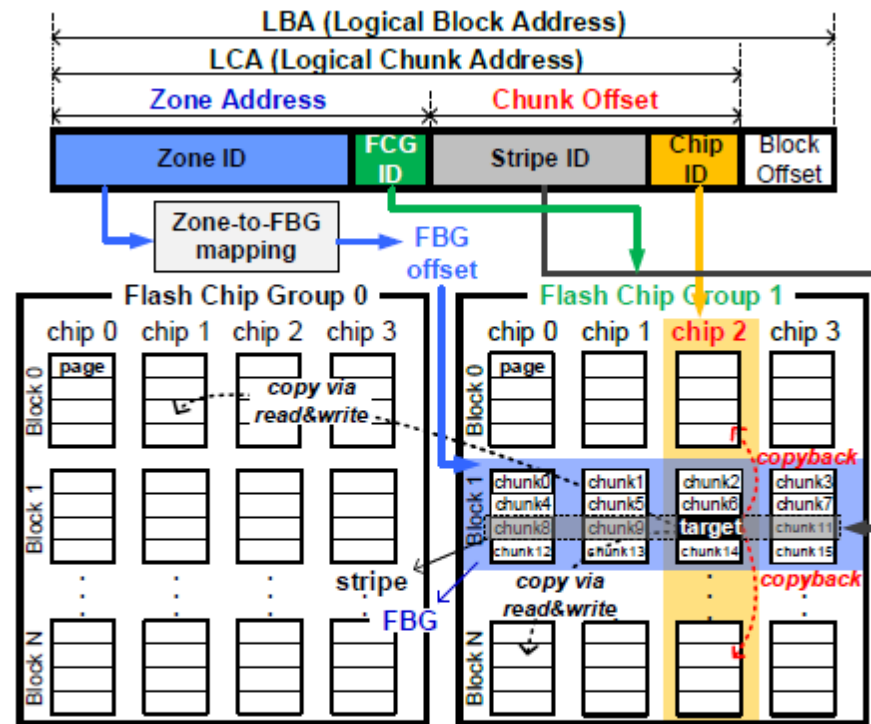


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

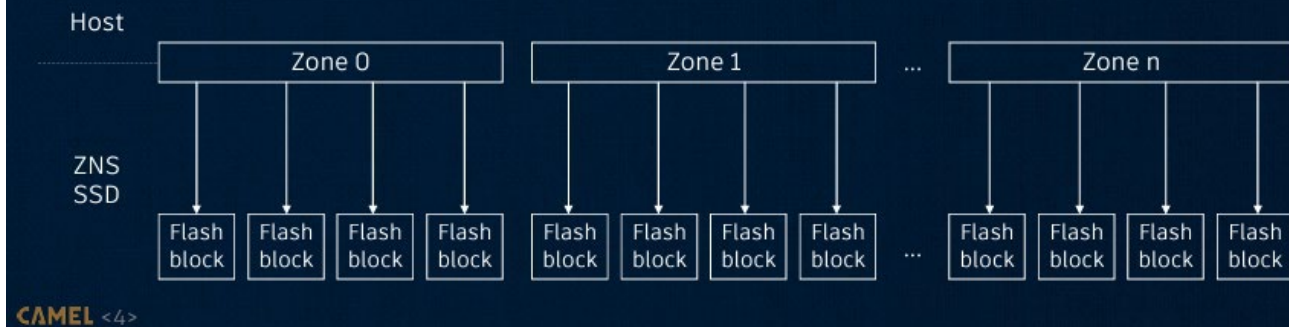
To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein

		<p>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.</p> <p>As an example, there is no substantial difference between (1) identifying a physical erase unit that corresponds to the identified logical erase unit and (2) identifying a physical erase unit group that corresponds to the identified 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.</p> <p>As another example, there is no substantial difference between (1) identifying a physical erase unit that corresponds to the identified logical erase unit and (2) having the addressed segment as a logical erase unit group which corresponds to a physical erase unit group and 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.</p>
<i>li</i>	<p>wherein the storage device is to detect a failure 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</p>	<p>Samsung's Accused Instrumentalities comprise a 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 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.</p> <p>Samsung's Accused Instrumentalities are able to detect a failure condition of physical blocks, retire the bad blocks, and substitute different blocks for the bad ones. Samsung's Accused Instrumentalities are able to remap each logical erase unit that had corresponded to one of the bad physical blocks (with the detected failure condition) to instead correspond to the replacement physical blocks for which there was not a failure condition and then utilize the remapped correspondence to service one or more of the incoming data access requests. See:</p>

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.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

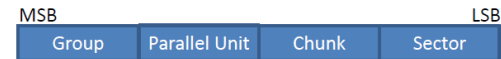


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

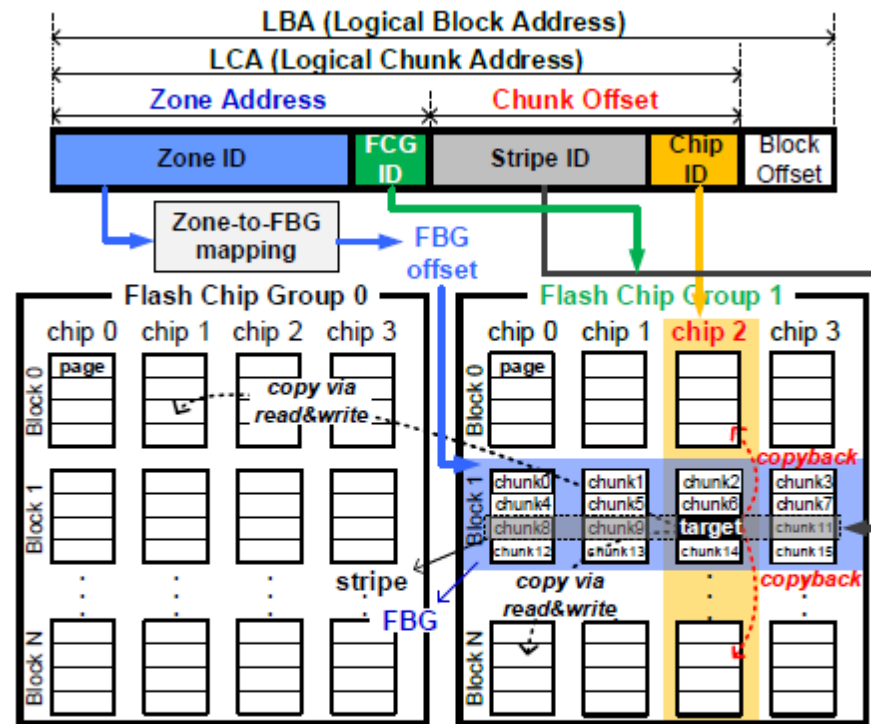


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

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

		<p>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.</p> <p>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.</p>
2	<p>The storage device of claim 1 wherein: said circuitry comprises at least one processor; and the storage device further comprises instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor.</p>	<p>Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.</p> <p>Samsung's Accused Instrumentalities comprise circuitry that comprises at least one processor and the storage device further comprising instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor.</p> <p>Samsung's Accused Instrumentalities contain a controller and firmware stored on a physical storage medium that when executed controls the controller.</p> <p>See:</p>



Source: Samsung PM1733 Product Brief.

		<div data-bbox="541 203 1780 1027"> <div data-bbox="583 240 1402 337"> <h2>09 Why Integration Matters</h2> <p>What Samsung's Vertical Integration Means to You</p> </div> <p>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.</p> <p>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).</p> </div> <p>Source: Source: Samsung SSD White Paper, at 30.</p>
4pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
4a	the storage device is to receive a maintenance request from a	Samsung's Accused Instrumentalities comprise a storage device that is to receive a maintenance request from a host which addresses a block device and a segment within the block device addressed by the maintenance request.

host which addresses a block device and a segment within the block device addressed by the maintenance request; and

Samsung's Accused Instrumentalities are able to receive and execute a Zone Management Send command specifying a Reset Zone action 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.																														
	<table><tr><th>Value</th><th>Description</th><th>Refer to section</th></tr><tr><td>00h</td><td>Reserved</td><td></td></tr><tr><td>01h</td><td>Close Zone: Close one or more zones.</td><td>3.4.3.1.1</td></tr><tr><td>02h</td><td>Finish Zone: Finish one or more zones.</td><td>3.4.3.1.2</td></tr><tr><td>03h</td><td>Open Zone: Open one or more zones.</td><td>3.4.3.1.3</td></tr><tr><td>04h</td><td>Reset Zone: Reset one or more zones.</td><td>3.4.3.1.4</td></tr><tr><td>05h</td><td>Offline Zone: Offline one or more zones.</td><td>3.4.3.1.5</td></tr><tr><td>06h to 0Fh</td><td>Reserved</td><td></td></tr><tr><td>10h</td><td>Set Zone Descriptor Extension: Attach Zone Descriptor Extension data to a zone.</td><td>3.4.3.1.6</td></tr><tr><td>11h to FFh</td><td>Reserved</td><td></td></tr></table>	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	
	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;

		<p>b) ZSE:Empty state, then no change shall be made to the zone state; and</p> <p>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.</p> <p>If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the:</p> <ul style="list-style-type: none"> a) ZSIO:Implicitly Opened state; b) ZSEO:Explicitly Opened state; c) ZSC:Closed state; and d) the ZSF:Full state, <p>shall be transitioned to the ZSE:Empty state.</p> <p>If the command completes successfully, then for each affected zone:</p> <ul style="list-style-type: none"> 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': <ul style="list-style-type: none"> a) Zone Descriptor Extension Valid; b) Finish Zone Recommended; c) Reset Zone Recommended; and d) Zone Finished by Controller.
--	--	---

Source: ZNS Specification, at 30–31.

A Zone Management Send command specifying a Reset Zone action is a maintenance request.

The Zone Management Send command addresses a namespace (block device) and a zone (segment) within the namespace. 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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Source: NVMe Base 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	<p>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.</p>

Source: ZNS Specification, at 29.

		<p>3.4.3.1.4 Reset Zone</p> <p>If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the:</p> <ul style="list-style-type: none"> 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; <div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> 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. <p>If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the:</p> <ul style="list-style-type: none"> a) ZSIO:Implicitly Opened state; b) ZSEO:Explicitly Opened state; c) ZSC:Closed state; and d) the ZSF:Full state, <p>shall be transitioned to the ZSE:Empty state.</p> <p>If the command completes successfully, then for each affected zone:</p> <ul style="list-style-type: none"> 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': <ul style="list-style-type: none"> a) Zone Descriptor Extension Valid; b) Finish Zone Recommended; c) Reset Zone Recommended; and d) Zone Finished by Controller. </div> <p>Source: ZNS Specification, at 30–31.</p>
4b	the storage device comprises circuitry to erase, in association	<p>Samsung's Accused Instrumentalities comprise a storage device comprising circuitry to erase, in association with execution of the maintenance request, one or more of the physical erase units which are mapped to the segment addressed by the maintenance request.</p>

<p>with execution of the maintenance request, one or more of the physical erase units which are mapped to the segment addressed by the maintenance request.</p>	<p>Samsung's Accused Instrumentalities are able to erase one or more of the physical blocks which are mapped to the addressed zone in association with execution of the Zone Management Send command specifying a Reset Zone action. See:</p> <div data-bbox="541 326 1761 464" style="border: 1px solid black; padding: 5px;"> <p>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.</p> </div> <p>Source: ZNS Specification, at 42.</p> <div data-bbox="541 548 1656 732" style="border: 1px solid black; padding: 5px;"> <p>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</p> </div> <p>Source: Bjørling.</p> <div data-bbox="541 813 1850 899" style="border: 1px solid black; padding: 5px;"> <p>The Zone Management Send command with a Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone.</p> </div> <p>Source: ZNS Specification, at 10.</p> <div data-bbox="541 984 1850 1089" style="border: 1px solid black; padding: 5px;"> <p>If the command completes successfully, then for each affected zone:</p> <ol style="list-style-type: none"> a) the Write Pointer zone attribute in the Zone Descriptor shall be set to the ZSLBA of the zone; </div> <p>Source: ZNS Specification, at 30–31.</p>
---	---

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).

		<p>2.2.1. Erase-before-write constraint</p> <p>Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (<i>in-place</i> update). <u>Before performing any write operation on a page which already contains data, this page has to be erased.</u> However, the erase operation affects <u>an entire block</u>, 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</p> <p>Source: Boukhobza at 24 (highlighted).</p>
5pre	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
5a	the flash memory is to have block devices, including the addressed block device; and	<p>Samsung's Accused Instrumentalities comprise flash memory that is to have block devices, including the addressed block device.</p> <p>Samsung's Accused Instrumentalities are able to have namespaces (block device) including the addressed namespace. See:</p> <div> <p>1.5.37 namespace</p> <p>A formatted quantity of non-volatile memory that may be directly accessed by a host.</p> <p>1.5.38 namespace ID (NSID)</p> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 22.</p>

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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

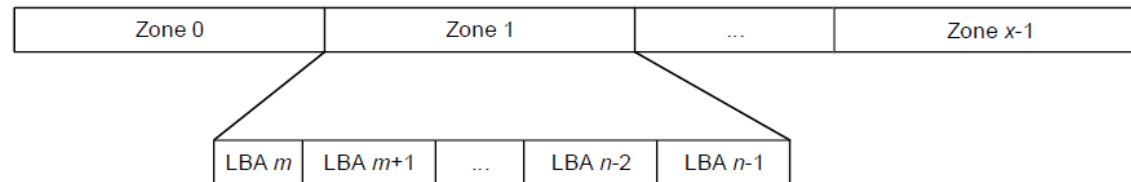
Source: NVMe Base Specification, at 107.

5b	in connection with a given one of the incoming data access requests, the	Samsung's Accused Instrumentalities comprise circuitry that is to, in connection with a given one of the incoming data access requests, identify and retrieve a look-up table specific to the addressed block device which was identified from the first address portion derived from the given one of the incoming data access requests, and identify a physical storage location in the flash memory dependent on the look-up table.
----	--	--

<p>circuitry is to: identify and retrieve a look-up table specific to the addressed block device which was identified from the first address portion derived from the given one of the incoming data access requests, and identify a physical storage location in the flash memory dependent on the look-up table.</p>	<p>Samsung's Accused Instrumentalities are able to identify a namespace (block device) from the NSID (first address portion) of an incoming read or write command (data access requests) and retrieve and use a look-up table specific to the namespace addressed by the NSID of the incoming read or write command to identify a physical storage location in the flash memory, including an erase unit, block, and/or page. See:</p> <p>See:</p> <div data-bbox="537 418 1740 997" data-label="Table"> <p style="text-align: center;">Figure 88: Common Command Format</p> <table> <tr> <th>Bytes</th><th>Description</th></tr> <tr> <td>07:04</td><td> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> </td></tr> </table> </div> <p>Source: NVMe Base Specification, at 107.</p>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Bytes	Description				
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>				

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.

Zone 0			Zone 1		...	Zone x-1	
LBA 0	LBA 1	...					LBA z-1



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.

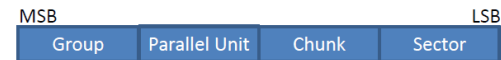
The address of a logical block, referred to commonly as LBA.

Source: NVM Command Set Specification, at 10.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

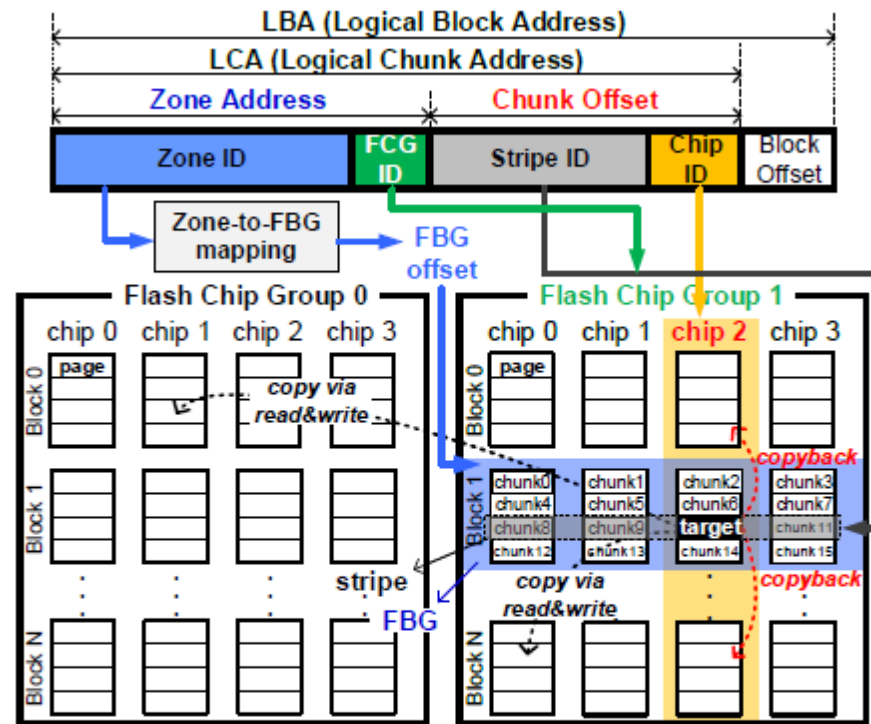


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

6pre

The storage device of claim 1 wherein:

Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.

6a	the storage device further comprises circuitry to track metadata for the addressed segment, where said metadata indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased; and	<p>Samsung's Accused Instrumentalities comprise circuitry to track metadata for the addressed segment, where said metadata indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased.</p> <p>Samsung's Accused Instrumentalities are able to maintain a write pointer (metadata) for each zone (segment). The write pointer indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased. For example, the extent to which memory can continue to receive additional write data without being erased can be calculated by subtracting the write pointer from the sum of the zone start LBA and the zone capacity (i.e., remaining space = ZSLBA + ZCAP – write pointer). See:</p> <table><tr><td>Write Pointer</td><td>The Write Pointer attribute defines the lowest numbered writeable logical block address in that zone. The validity of the write pointer is zone state specific and is defined per zone type (refer to section 2.1.1.2).</td></tr></table> <p>Source: ZNS Specification, at 9.</p> <div><p>2.1.1.2.1.1 Writing in Sequential Write Required Zones</p><p>The following commands may be used to write to logical blocks in a specific zone of zone type Sequential Write Required:</p><ul style="list-style-type: none">• Write command;• Write Zeroes command;• Write Uncorrectable command;• Copy command; and• Zone Append command.<p>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.</p></div> <p>Source: ZNS Specification, at 9.</p>	Write Pointer	The Write Pointer attribute defines the lowest numbered writeable logical block address in that zone. The validity of the write pointer is zone state specific and is defined per zone type (refer to section 2.1.1.2).
Write Pointer	The Write Pointer attribute defines the lowest numbered writeable logical block address in that zone. The validity of the write pointer is zone state specific and is defined per zone type (refer to section 2.1.1.2).			

Figure 5 shows an example of a zone in the ZSE:Empty state. LBA m is the ZSLBA attribute, the write pointer indicates ZSLBA, and $n-1$ is the highest LBA of the zone.

Figure 5: Write Pointer in an Empty Zone

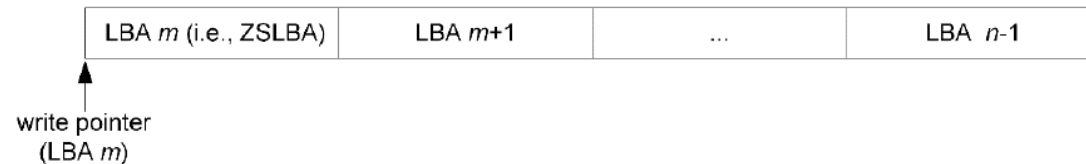
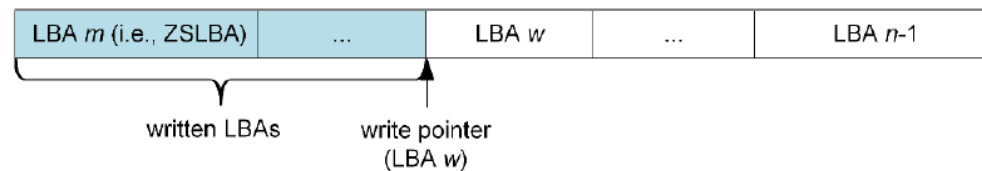


Figure 6 shows an example of a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state, that has had some logical blocks written. The write pointer, indicated by LBA w , is the lowest-numbered unwritten LBA (i.e., the next LBA to be written) and $n-1$ is the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

Figure 6: Write Pointer in a Partially Written Zone



Source: ZNS Specification, at 10.

Zone Capacity (ZCAP): This field contains the maximum number of logical blocks that are available to be written with user data when the zone is in the ZSE:Empty state. This value shall be less than or equal to the Zone Size field (refer to Figure 49).

Source: ZNS Specification, at 28.

Zone Descriptor	Zone Capacity (ZCAP)	15:8	Indicates the maximum number of logical blocks that may be written in the zone associated with the data descriptor.
-----------------	----------------------	------	---

Source: ZNS Specification, at 46.

6b	the storage device further comprises circuitry to transmit information to a host dependent on said tracked metadata.	<p>Samsung's Accused Instrumentalities comprise circuitry to transmit information to a host dependent on said tracked metadata.</p> <p>Samsung's Accused Instrumentalities are able to transmit a Report Zones data structure that includes a Zone Descriptor to the host, which includes the write pointer (metadata). See:</p> <div data-bbox="537 383 1843 777" style="border: 1px solid black; padding: 10px;"> <p>3.4.2.1 Zone Receive Actions</p> <p>The Zone Management Receive command Zone Receive Action field specifies what action to perform.</p> <p>3.4.2.1.1 Report Zones</p> <p>The Report Zones action returns the Report Zones data structure (refer to Figure 35). The Zone Descriptors of the Report Zones data structure shall:</p> <ul style="list-style-type: none"> a) report only Zone Descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the command; b) match the criteria in the Zone Receive Action Specific field; and c) be sorted in ascending order by the ZSLBA value of each zone. </div> <p>Source: ZNS Specification at 26.</p>
----	--	--

3.4.2.2.1 Report Zones Data Structure

Figure 35 defines the Report Zones Data Structure.

Figure 35: Report Zones Data Structure

Bytes	Description
07:00	Number of Zones: If the Partial Report bit (refer to Figure 34) is cleared to '0', then this field indicates the number of zones that match the criteria defined in section 3.4.2.1.1. If the Partial Report bit is set to '1', then this field indicates the number of zones for which complete Zone Descriptors were transferred to the data buffer. Refer to section 3.4.2.1.1 for the content of the data buffer.
63:08	Reserved
127:64	Zone Descriptor 0: Contains the Zone Descriptor for the first zone reported, if any (refer to Figure 37).
191:128	Zone Descriptor 1: Contains the Zone Descriptor for the second zone reported, if any.
...	...
$((n+1)*64)+63:(n+1)*64$	Zone Descriptor n: Contains the Zone Descriptor for the last zone reported, if any.

Source: ZNS Specification, at 26.

3.4.2.2.3 Zone Descriptor Data Structure

Figure 37 defines the Zone Descriptor data structure.

Figure 37: Zone Descriptor Data Structure

Bytes	Description									
00	Bits	Description								
	7:4	Reserved								
	3:0	Zone Type (ZT): This field indicates the type of the zone.								
		<table><tr><th>Value</th><th>Definition</th></tr><tr><td>2h</td><td>Sequential Write Required</td></tr><tr><td>All other values</td><td>Reserved</td></tr></table>			Value	Definition	2h	Sequential Write Required	All other values	Reserved
		Value	Definition							
		2h	Sequential Write Required							
All other values	Reserved									

		<table><tr><td>23:16</td><td>Zone Start Logical Block Address (ZSLBA): This field contains the 64-bit address of the lowest logical block for the zone.</td></tr><tr><td>31:24</td><td>Write Pointer (WP): This field is the logical block address where the next write operation for this zone should be issued. Refer to section 2.1.1.2.1 for the behavior of the write pointer.</td></tr></table> <p>Source: ZNS Specification, at 27–28.</p>	23:16	Zone Start Logical Block Address (ZSLBA): This field contains the 64-bit address of the lowest logical block for the zone.	31:24	Write Pointer (WP): This field is the logical block address where the next write operation for this zone should be issued. Refer to section 2.1.1.2.1 for the behavior of the write pointer.
23:16	Zone Start Logical Block Address (ZSLBA): This field contains the 64-bit address of the lowest logical block for the zone.					
31:24	Write Pointer (WP): This field is the logical block address where the next write operation for this zone should be issued. Refer to section 2.1.1.2.1 for the behavior of the write pointer.					
7pre	The storage device of claim 1 wherein:	Samsung’s Accused Instrumentalities comprise the storage device of claim 1. See claim 1.				
7a	the incoming data access requests are read requests received from a host;	<p>Samsung’s Accused Instrumentalities comprise the storage device of claim 1 wherein the incoming data access requests are read requests received from a host.</p> <p>Samsung’s Accused Instrumentalities are able to receive read requests as incoming data access requests from a host. See claim 1.</p>				
7b	the storage device can receive write requests from the host, and the storage device comprises circuitry to issue commands to the flash memory in fulfillment of the write requests;	<p>Samsung’s Accused Instrumentalities comprise the storage device of claim 1 wherein the storage device can receive write requests from the host, and the storage device comprises circuitry to issue commands to the flash memory in fulfillment of the write requests.</p> <p>Samsung’s Accused Instrumentalities are able to receive Zone Append commands from the host and issue commands to the flash memory in fulfillment of a Zone Append command from the host. A Zone Append command is a write request. See:</p> <div><p>3.4.1 Zone Append command</p><p>The Zone Append command writes data and metadata, if applicable, to the I/O controller for the zone indicated by the ZSLBA field. The controller assigns the data and metadata, if applicable, to a set of logical blocks within the zone. The lowest LBA of the set of logical blocks written is returned in the completion queue entry (refer to section 3.4.1.2). The host may also specify protection information to include as part of the operation.</p></div>				

		<div><div><div>Figure 22: Zone Append – Metadata Pointer</div><table><tr><th>Bits</th><th>Description</th></tr><tr><td>63:00</td><td>Metadata Pointer (MPTR): This field contains the Metadata Pointer, if applicable. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.</td></tr></table></div><div><div>Figure 23: Zone Append – Data Pointer</div><table><tr><th>Bits</th><th>Description</th></tr><tr><td>127:00</td><td>Data Pointer (DPTR): This field specifies the location of a data buffer where data is transferred from. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.</td></tr></table></div></div> <div>Source: ZNS Specification at 21.</div>	Bits	Description	63:00	Metadata Pointer (MPTR): This field contains the Metadata Pointer, if applicable. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.	Bits	Description	127:00	Data Pointer (DPTR): This field specifies the location of a data buffer where data is transferred from. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.
Bits	Description									
63:00	Metadata Pointer (MPTR): This field contains the Metadata Pointer, if applicable. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.									
Bits	Description									
127:00	Data Pointer (DPTR): This field specifies the location of a data buffer where data is transferred from. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.									
7c	the storage device is to assign an address value in association with fulfillment of each of the write requests, in order to identify a physical storage location in which associated write data has been stored,	<div><p>Samsung’s Accused Instrumentalities comprise the storage device of claim 1 wherein the storage device is to assign an address value in association with fulfillment of each of the write requests, in order to identify a physical storage location in which associated write data has been stored, and comprises circuitry to transmit information representing each assigned address value to the host.</p><p>Samsung’s Accused Instrumentalities are able to assign an address value in association with fulfillment of the Zone Append command, identify a physical storage location to store the data, and transmit the assigned address value to the host via a completion queue entry. The assigned address value is the lowest LBA of the set of LBAs assigned for storage. See:</p><div><div>3.4.1 Zone Append command</div><div>The Zone Append command writes data and metadata, if applicable, to the I/O controller for the zone indicated by the ZSLBA field. The controller assigns the data and metadata, if applicable, to a set of logical blocks within the zone. The lowest LBA of the set of logical blocks written is returned in the completion queue entry (refer to section 3.4.1.2). The host may also specify protection information to include as part of the operation.</div></div></div>								

	and comprises circuitry to transmit information representing each assigned address value to the host; and	<p>3.4.1.2 Command Completion</p> <p>When the command is completed, the controller shall post a completion queue entry (CQE) to the associated I/O Completion Queue indicating the status for the command. If the command is successfully completed, then the lowest LBA containing the data written by the command is returned in the Assigned LBA (ALBA) field of the completion queue entry. Dword 0 contains bits 31:00 of the ALBA field; Dword 1 contains bits 63:32 of the ALBA field. If the command is not successfully completed, then the contents of the ALBA field are undefined.</p> <p>Source: ZNS Specification at 21.</p>
7d	for one of the incoming data access requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, the addressed storage location identified from the third portion is dependent on the assigned address value which was transmitted to the host in association	<p>Samsung's Accused Instrumentalities comprise the storage device of claim 1 wherein for one of the incoming data access requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, the addressed storage location identified from the third portion is dependent on the assigned address value which was transmitted to the host in association with the one of the write requests.</p> <p>Samsung's Accused Instrumentalities are able to receive and execute incoming Read commands (data access requests), which are directed to data which has been written pursuant to the Zone Append command, where the third address portion is used to identify the storage location that corresponds to the assigned address value that was transmitted to the host in association with the execution of a Zone Append command (write request),. See:</p> <p>3.3.4 Read command</p> <p>The Read command reads data and metadata, if applicable, from the I/O controller for the LBAs indicated. The command may specify protection information to be checked as part of the read operation.</p> <p>Source: NVM Command Set Specification, at 48.</p>

with the one of the write requests.

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.

Figure 52: Read – 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 read 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 49.

3.4.1 Zone Append command

The Zone Append command writes data and metadata, if applicable, to the I/O controller for the zone indicated by the ZSLBA field. The controller assigns the data and metadata, if applicable, to a set of logical blocks within the zone. The lowest LBA of the set of logical blocks written is returned in the completion queue entry (refer to section 3.4.1.2). The host may also specify protection information to include as part of the operation.

Figure 22: Zone Append – Metadata Pointer

Bits	Description
63:00	Metadata Pointer (MPTR): This field contains the Metadata Pointer, if applicable. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.

Figure 23: Zone Append – Data Pointer

Bits	Description
127:00	Data Pointer (DPTR): This field specifies the location of a data buffer where data is transferred from. Refer to the Common Command Format figure in the NVM Express Base Specification for the definition of this field.

3.4.1.2 Command Completion

When the command is completed, the controller shall post a completion queue entry (CQE) to the associated I/O Completion Queue indicating the status for the command. If the command is successfully completed, then the lowest LBA containing the data written by the command is returned in the Assigned LBA (ALBA) field of the completion queue entry. Dword 0 contains bits 31:00 of the ALBA field; Dword 1 contains bits 63:32 of the ALBA field. If the command is not successfully completed, then the contents of the ALBA field are undefined.

Source: ZNS Specification at 21.

location in the flash memory and write the host-selected data to the second storage location in the flash memory without sending the data being copied to the host; and

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

Range	Bytes	Description									
Source Range 0	03:00	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>	Format 0h bytes	Format 2h bytes	Description	03:00	n/a	Reserved	n/a	03:00	Source Namespace Identifier (SNSID): Specifies the source namespace for this Source Range entry.
		Format 0h bytes	Format 2h 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)										

Range	Bytes	Description								
		Read Parameters (RPARS): This field specifies attributes as follows.								
	19:16	<table><tr><th>Bits</th><th>Description</th></tr><tr><td>31:20</td><td>Reserved</td></tr><tr><td>19:16</td><td>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.</td></tr><tr><td>15:00</td><td>Number of Logical Blocks (NLB): This field indicates the number of logical blocks to be copied. This is a 0's based value.</td></tr></table>	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.
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:									
		<table><tr><th>Format 1h bytes</th><th>Format 3h 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>	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.
		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:									
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>31:20</td><td>Reserved</td></tr><tr><td>19:16</td><td>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.</td></tr><tr><td>15:00</td><td>Number of Logical Blocks (NLB): This field indicates the number of logical blocks to be copied. This is a 0's based value.</td></tr></table>	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.	
		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.										

		<p>Figure 41: Source LBA and Destination LBA Relationship Example</p> <p>Destination LBAs: SDLBA + Sum (all NLB fields) 10001-10170</p> <p>Source LBAs:</p> <ul style="list-style-type: none"> Range 0: 101-150 Range 1: 2300-2399 Range 2: 331-340 Range 3: 215-224 <p>Source: NVM Command Set Specification at 29–37.</p>
8b	the request from the host to copy is to designate destination address information corresponding to the second storage location, wherein the storage device is to identify from the destination	<p>Samsung’s Accused Instrumentalities comprise the storage device which can receive the request from the host to copy which is to designate destination address information corresponding to the second storage location, wherein the storage device is to identify from the destination address information a destination block device and an addressed segment within the destination block device, and wherein the second storage location corresponds to the addressed segment within the destination block device.</p> <p>Samsung’s Accused Instrumentalities are able to receive and execute a Copy command. The Copy command designates destination address information including a destination namespace (block device) and a Starting Destination LBA (“SDLBA”) which indicates an addressed zone (segment) within the namespace. See:</p>

<p>address information a destination block device and an addressed segment within the destination block device, and wherein the second storage location corresponds to the addressed segment within the destination block device.</p>	<div data-bbox="541 201 1831 448"> <p>3.3.2 Copy command</p> <p>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.</p> </div> <p>Source: NVM Command Set Specification at 29.</p>
---	--

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

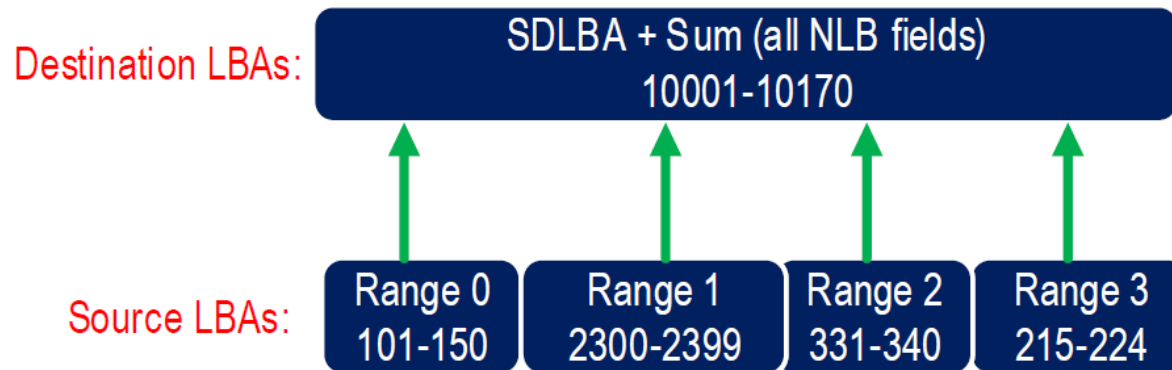
Opcode by Field			Combined Opcode ¹	Command ²	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer ³			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express 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



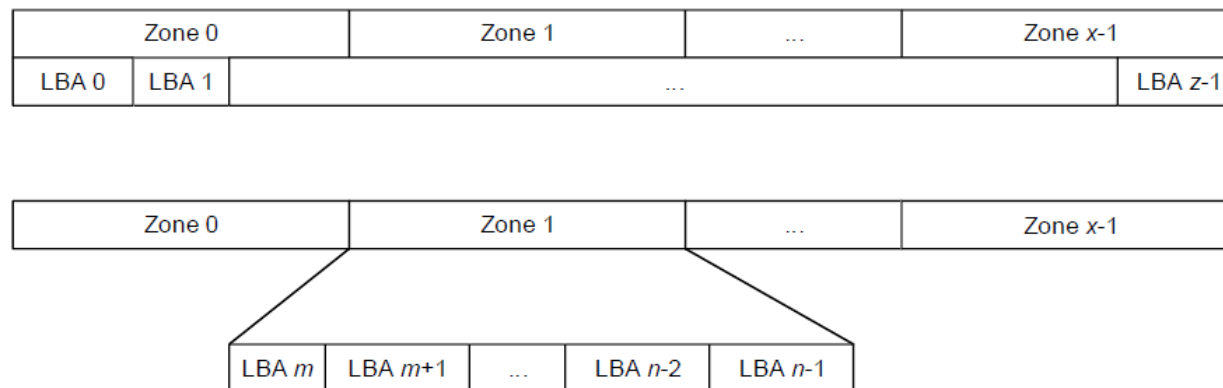
Source: NVM Command Set Specification at 29–37.

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.

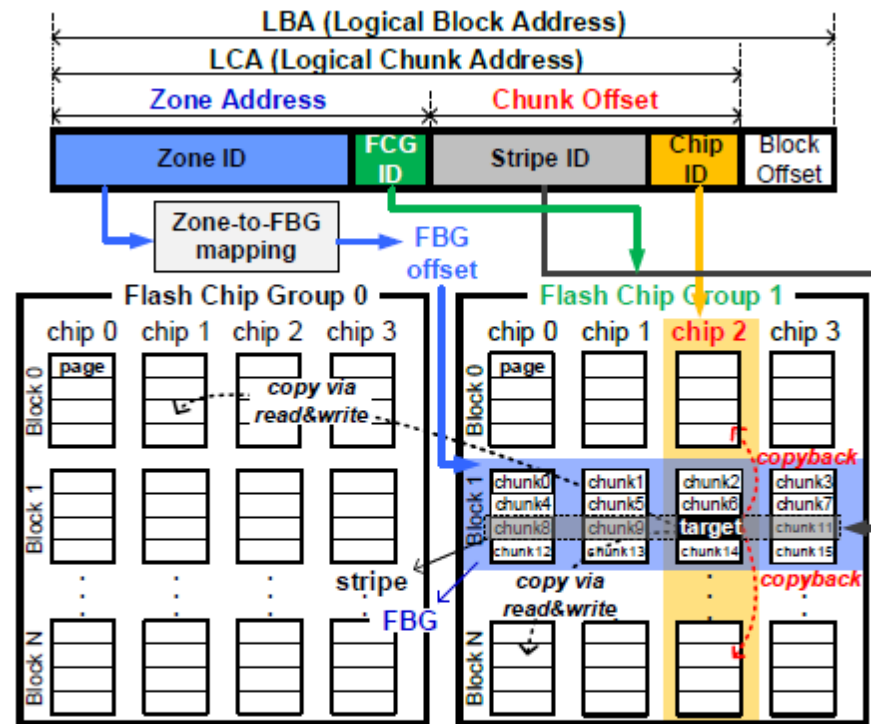


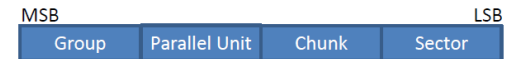
Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

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

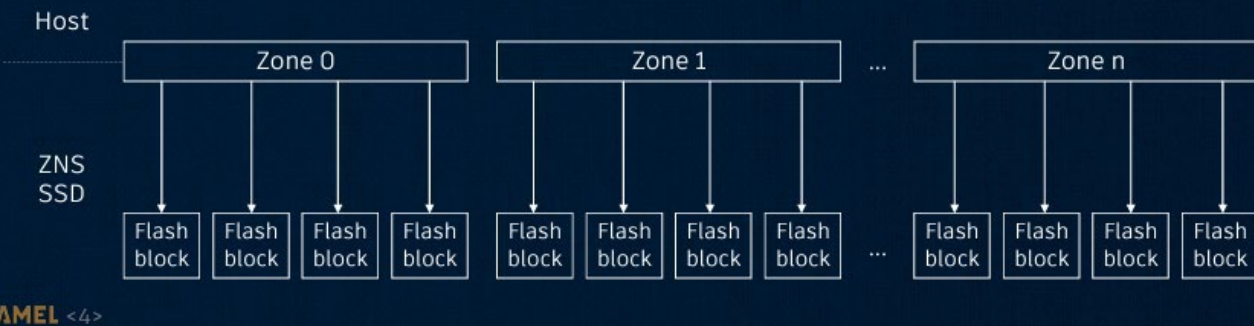
* "FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs" Huang et al, USENIX-FAST 2017

15

Source: Microsoft Denali Presentation, at 15.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



Source: Bae, at 4.

9

The storage device of claim 8 wherein the storage device is to identify from the destination address

Samsung's Accused Instrumentalities comprise the storage device of claim 8. See claim 8.

Samsung's Accused Instrumentalities comprise a storage device that is to identify from the destination address information a destination logical erase unit and is to select the second storage location in dependence on the identified destination logical erase unit.

Samsung's Accused Instrumentalities are able to receive and execute a Copy command that designates destination address information. Samsung's Accused Instrumentalities are able to identify a destination

	<p>information a destination logical erase unit and is to select the second storage location in dependence on the identified destination logical erase unit.</p>	<p>logical erase unit from the destination address information and to select the second storage location in dependence on the identified 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:</p> <div data-bbox="537 415 1845 667" style="border: 1px solid black; padding: 10px;"> <p>3.3.2 Copy command</p> <p>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.</p> </div> <p>Source: NVM Command Set Specification at 29.</p>
--	--	---

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

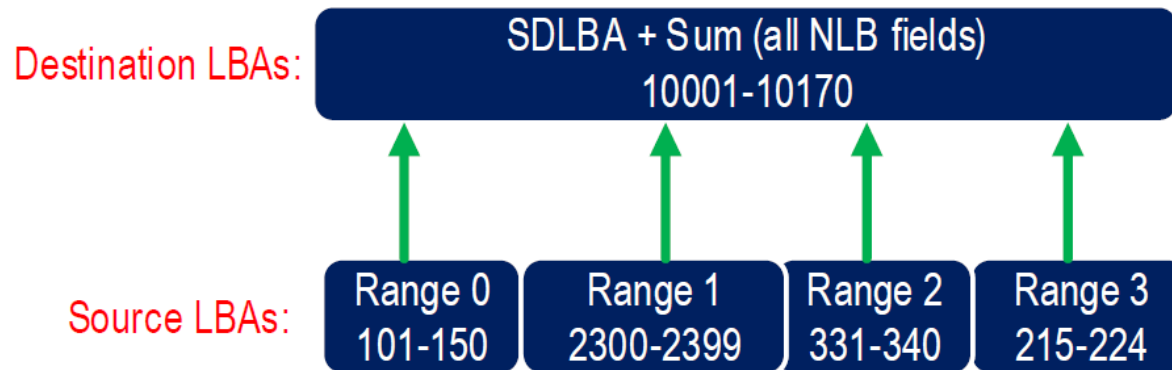
Opcode by Field			Combined Opcode ¹	Command ²	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer ³			
Refer to the NVM Express Base Specification				Reservation Acquire	NVM Express Base Specification
Refer to the NVM Express Base Specification				Reservation Release	NVM Express 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

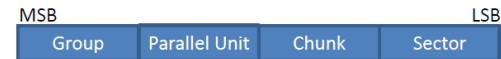


Source: NVM Command Set Specification at 29–37.

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

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

15

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.

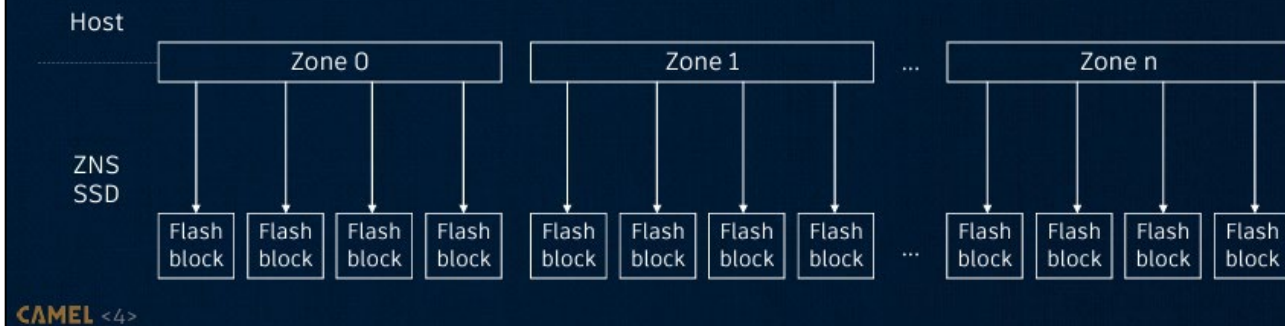
		<p>2. Why is OP important?</p> <p>OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process 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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

		<p>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.</p> <p>As an example, there is no substantial difference between (1) identifying a destination logical erase unit and selecting a storage location in dependence on the identified 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 storage location in dependence on the identified 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.</p>
12	<p>The storage device of claim 1 wherein the addressed segment corresponds to an integer number of physical erase units for which the failure condition has not been detected, and the third address portion specifies a location within a sequential range of locations that correspond to the integer number of</p>	<p>Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.</p> <p>Samsung's Accused Instrumentalities comprise the storage device of claim 1 wherein the addressed segment corresponds to an integer number of physical erase units for which the failure condition has not been detected, and the third address portion specifies a location within a sequential range of locations that correspond to the integer number of physical erase units.</p> <p>Samsung's Accused Instrumentalities are able to receive and execute incoming data access requests, which identify an addressed zone (segment) that corresponds to an integer number of physical blocks (erase units) for which a failure condition has not been detected, and an LBA and/or zone offset (third address portion) identifies 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 integer number of physical blocks of the zone. See:</p>

physical erase units.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



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.

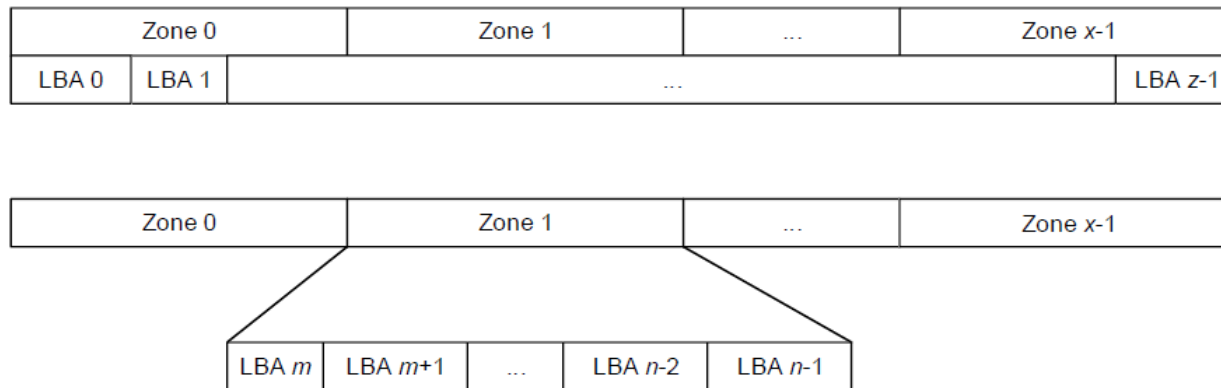
Source: Zoned Storage Introduction.

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.

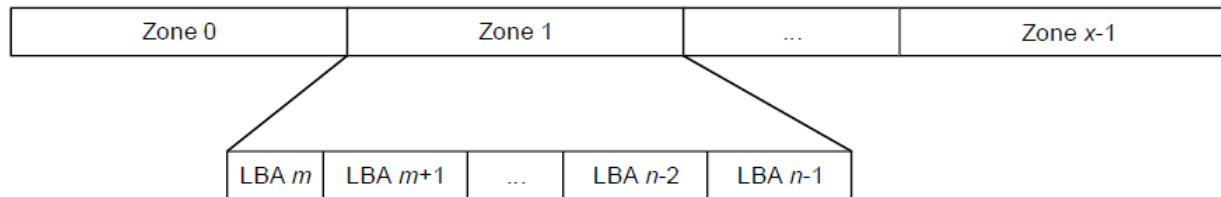
The third address portion (the LBA and/or the zone offset) specifies a location within a sequential range of locations that correspond to the integer number of physical erase units. See:

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.

13pr
e

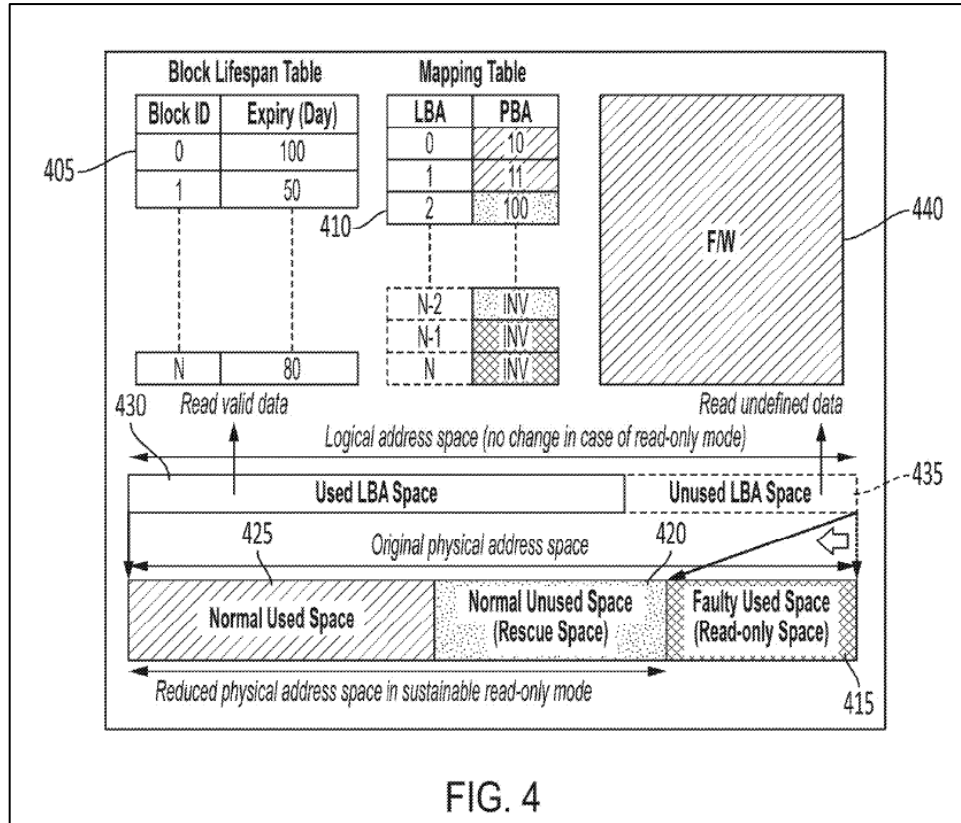
The storage device of claim 1 wherein:

Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.

13a	the flash memory is to have block devices, including the addressed block device; and	<p>Samsung's Accused Instrumentalities comprise flash memory that has block devices, including the addressed block device.</p> <p>Samsung's Accused Instrumentalities have namespaces (block devices) including the addressed namespace (the addressed block device). See:</p> <div data-bbox="537 383 1837 669" style="border: 1px solid black; padding: 5px;"> <p>1.5.37 namespace</p> <p>A formatted quantity of non-volatile memory that may be directly accessed by a host.</p> <p>1.5.38 namespace ID (NSID)</p> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 22.</p> <div data-bbox="537 753 1818 951" style="border: 1px solid black; padding: 5px;"> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 28.</p>
-----	--	---

representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,

locations associated with the namespace, including metadata such as age of data or data retention period. See:

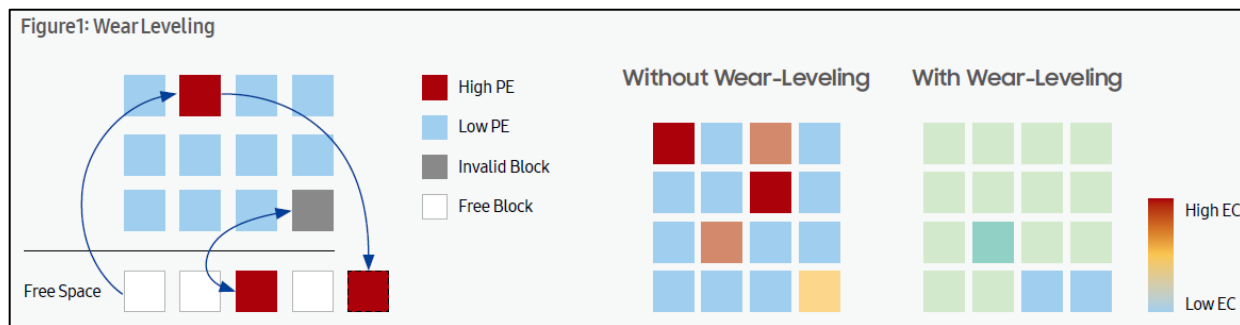


		<p>[0067] The storage media 125 may have a retention period (which may depend on the usage history of the storage media 125, and, as such, may vary within the storage media 125); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become 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.</p>	
--	--	--	--

		<p>[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.</p>	
--	--	---	--

[0090] FIG. 5A 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. 5B 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.

		<table> <tr> <th>ID</th><th>Attribute name</th><th>Status Flag</th><th>Threshold (%)</th></tr> <tr> <td>5</td><td>Reallocated Sector Count</td><td>110011</td><td>10</td></tr> <tr> <td>9</td><td>Power-on Hours</td><td>110010</td><td>-</td></tr> <tr> <td>12</td><td>Power-on Count</td><td>110010</td><td>-</td></tr> <tr> <td>177</td><td>Wear Leveling Count</td><td>010011</td><td>5</td></tr> <tr> <td>179</td><td>Used Reserved Block Count (total)</td><td>010011</td><td>10</td></tr> <tr> <td>180</td><td>Unused Reserved Block Count (total)</td><td>010011</td><td>10</td></tr> <tr> <td>181</td><td>Program Fail Count (total)</td><td>110010</td><td>10</td></tr> <tr> <td>182</td><td>Erase Fail Count (total)</td><td>110010</td><td>10</td></tr> <tr> <td>183</td><td>Runtime Bad Count (total)</td><td>010011</td><td>10</td></tr> <tr> <td>184</td><td>End to End Error data path Error count</td><td>110011</td><td>97</td></tr> <tr> <td>187</td><td>Uncorrectable Error Count</td><td>110010</td><td>-</td></tr> <tr> <td>190</td><td>Airflow Temperature</td><td>110010</td><td>-</td></tr> </table>	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	-	
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: Samsung Over-Provisioning White Paper, 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.

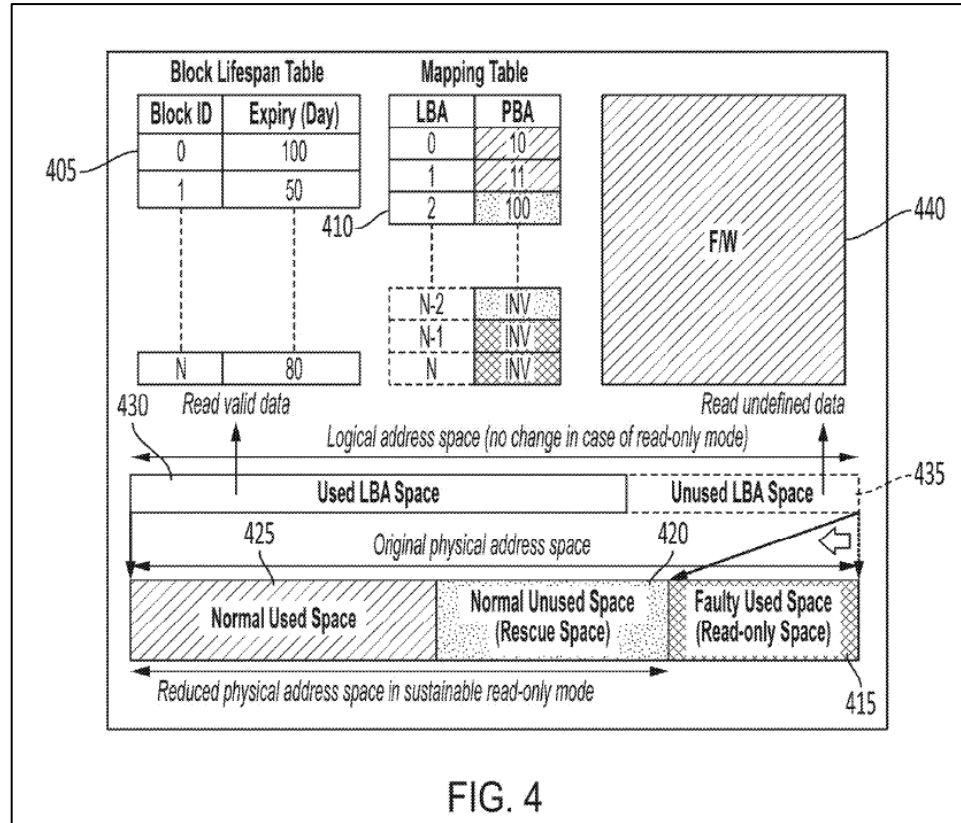
		<div data-bbox="554 207 768 245" data-label="Section-Header"> <h3>Wear Leveling</h3> </div> <div data-bbox="554 272 1766 505" data-label="Text"> <p>NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called 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.</p> </div> <div data-bbox="554 574 682 613" data-label="Section-Header"> <h3>Wrap Up</h3> </div> <div data-bbox="554 641 1766 771" data-label="Text"> <p>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.</p> </div> <div data-bbox="554 828 793 867" data-label="Section-Header"> <h3>Why Samsung?</h3> </div> <div data-bbox="554 894 1766 1159" data-label="Text"> <p>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.</p> </div> <div data-bbox="533 1208 1073 1245" data-label="Text"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p> <p>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.</p> <p>Source: Samsung Tech Blog.</p>	
--	--	---	--

		<p>2.2.3. Reliability limitation</p> <p>A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (<i>bitflips</i>) 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].</p> <p>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].</p> <p>Source: Boukhobza, at 26.</p>	
13d	compare the metadata to at	<p>Samsung's Accused Instrumentalities comprise circuitry to compare the metadata to at least one threshold.</p> <p>Samsung's Accused Instrumentalities are able to compare the metadata to at least one threshold. The metadata includes metadata representing a time since data was programmed in one or more physical</p>	

least one threshold,

storage locations associated with the namespace, including metadata such as age of data or data retention period. See:

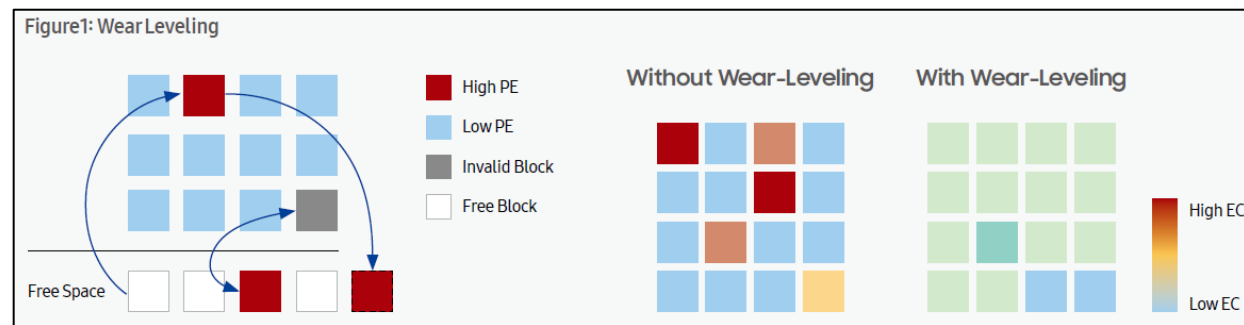


		<p>[0067] The storage media 125 may have a retention period (which may depend on the usage history of the storage media 125, and, as such, may vary within the storage media 125); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become 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.</p>	
--	--	--	--

		<p>[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.</p>	
--	--	---	--

[0090] FIG. 5A 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. 5B 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: Samsung Over-Provisioning White Paper, 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.

		<div data-bbox="554 207 768 245" data-label="Section-Header"> <h3>Wear Leveling</h3> </div> <div data-bbox="554 272 1766 505" data-label="Text"> <p>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.</p> </div> <div data-bbox="554 574 682 612" data-label="Section-Header"> <h3>Wrap Up</h3> </div> <div data-bbox="554 639 1766 769" data-label="Text"> <p>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.</p> </div> <div data-bbox="554 829 793 867" data-label="Section-Header"> <h3>Why Samsung?</h3> </div> <div data-bbox="554 894 1766 1159" data-label="Text"> <p>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.</p> </div> <div data-bbox="535 1208 1073 1245" data-label="Text"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p> <p>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.</p> <p>Source: Samsung Tech Blog.</p>	
--	--	---	--

		<p>2.2.3. Reliability limitation</p> <p>A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (<i>bitflips</i>) 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].</p> <p>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].</p> <p>Source: Boukhobza, at 26.</p>	
13e	upon detection that specific metadata has satisfied the	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 one of the physical erase units for which the failure	

	<p>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 one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and</p>	<p>condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device.</p> <p>Samsung's Accused Instrumentalities are able to perform internal operations to copy data from one physical storage location at one physical erase unit to another physical erase unit, where a failure condition has not been detected for either physical erase unit. Samsung's Accused Instrumentalities are able to perform such internal operations upon detection that the time since last write operation has satisfied the one or more thresholds. See:</p> <div data-bbox="537 493 1818 1344" style="border: 1px solid black; padding: 10px;"> <p>5.4 Reset Zone Recommended</p> <p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:</p> <ul style="list-style-type: none"> 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. <p>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'.</p> </div> <p>Source: ZNS Specification, at 41–42.</p>
--	--	--

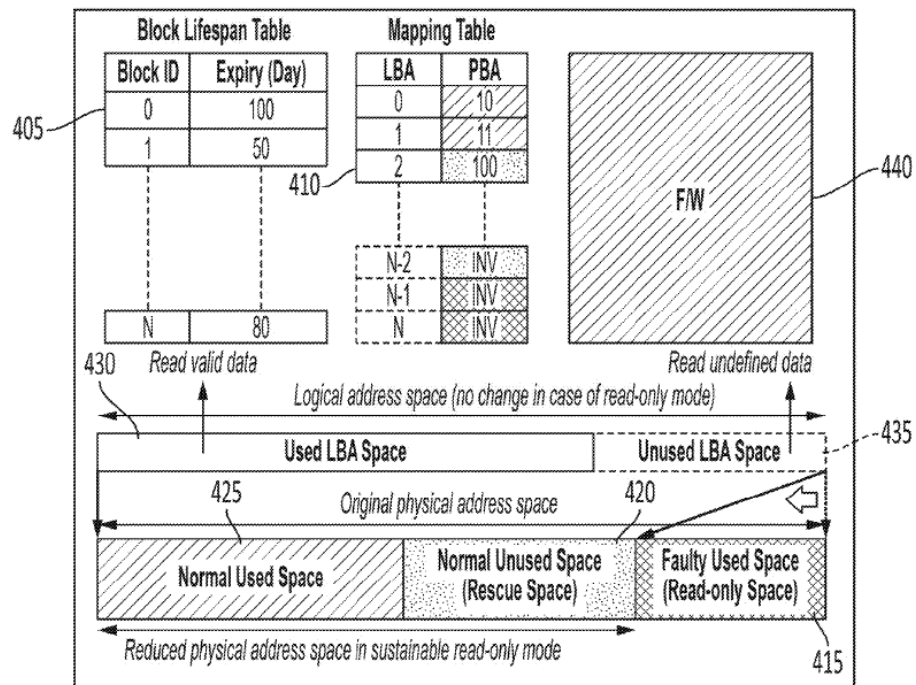


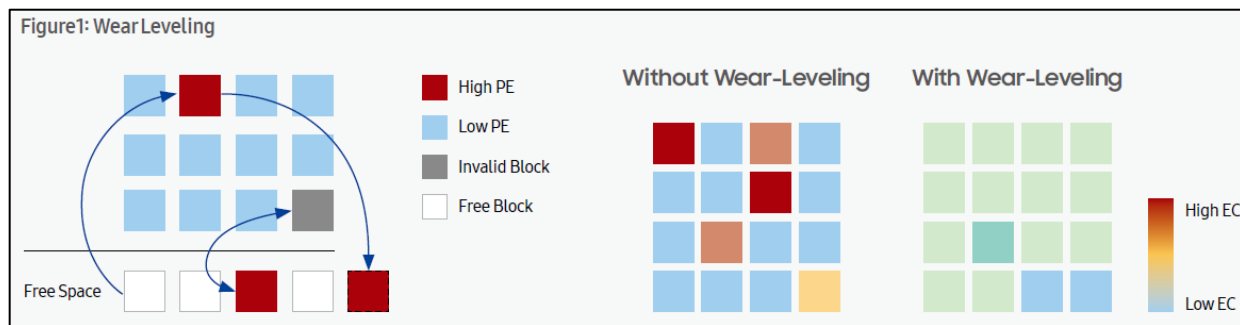
FIG. 4

		<p>[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.</p>	
--	--	--	--

	<p>[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.</p>	
--	---	--

[0090] FIG. 5A 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. 5B 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: Samsung Over-Provisioning White Paper, 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.

		<div data-bbox="552 207 768 248"> <h3>Wear Leveling</h3> </div> <div data-bbox="552 272 1766 505"> <p>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.</p> </div> <div data-bbox="552 574 682 615"> <h3>Wrap Up</h3> </div> <div data-bbox="552 639 1766 773"> <p>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.</p> </div> <div data-bbox="552 828 793 868"> <h3>Why Samsung?</h3> </div> <div data-bbox="552 893 1766 1159"> <p>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.</p> </div> <div data-bbox="533 1208 1073 1243"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p> <p>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.</p> <p>Source: Samsung Tech Blog.</p>	
--	--	---	--

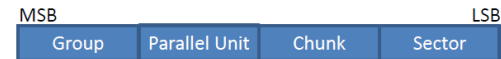
		<p>2.2.3. Reliability limitation</p> <p>A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (<i>bitflips</i>) may happen in the stored data. These errors are due to the NAND flash memory cells' density inside the chip, and to the high voltages 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].</p> <p>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].</p> <p>Source: Boukhobza, at 26.</p>	
13f	remap the logical erase unit corresponding	Samsung's Accused Instrumentalities comprise circuitry to remap the logical erase unit corresponding to the first one of the physical erase units for which the failure condition has not been detected by said	

<p>to the first one of the physical erase units for which the failure condition has not been detected by said storage device to the second one of the physical erase units for which the failure condition has not been detected by said storage device.</p>	<p>storage device to the second one of the physical erase units for which the failure condition has not been detected by said storage device.</p> <p>Samsung's Accused Instrumentalities are able to remap the correspondence from the logical erase unit from a first physical erase unit for which the failure condition has not been detected to a second physical erase unit for which the failure condition has not been detected in association with the internal operations so that the location of data is properly tracked after the copy. See:</p> <div data-bbox="537 456 1818 609" style="border: 1px solid black; padding: 10px;"> <p>5.4 Reset Zone Recommended</p> <p>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</p> </div> <div data-bbox="537 638 1818 1308" style="border: 1px solid black; padding: 10px;"> <p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:</p> <ul style="list-style-type: none"> 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. <p>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'.</p> </div> <p>Source: ZNS Specification, at 41–42.</p>
--	--

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

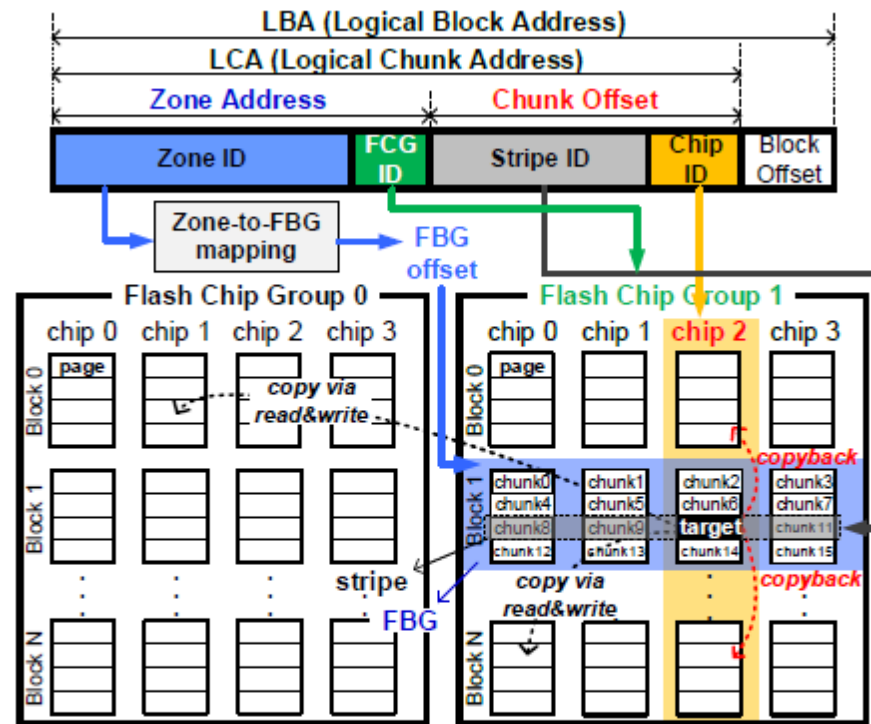


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

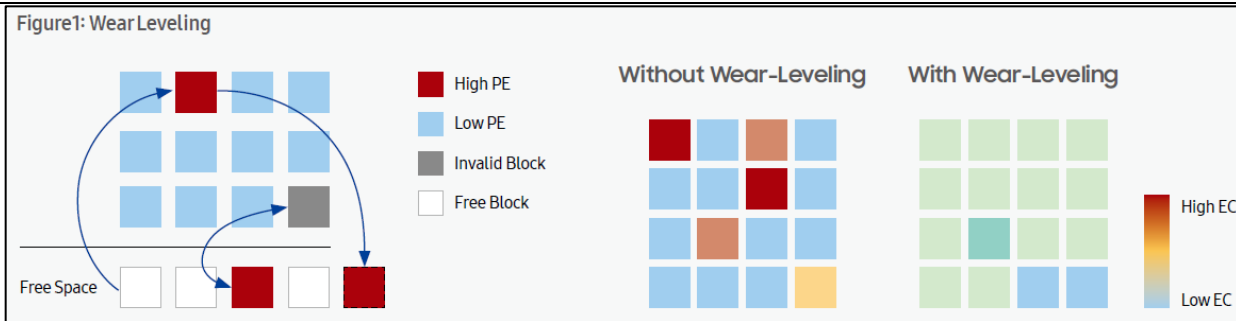
Source: Han, at 149.

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

		<p>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.</p> <p>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.</p>
<i>14pre</i>	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
<i>14a</i>	the flash memory is to have block devices, including the addressed block device; and	<p>Samsung's Accused Instrumentalities comprise flash memory that has block devices, including the addressed block device.</p> <p>Samsung's Accused Instrumentalities have namespaces (block devices) including the addressed namespace (the addressed block device). See:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>1.5.37 namespace</p> <p>A formatted quantity of non-volatile memory that may be directly accessed by a host.</p> <p>1.5.38 namespace ID (NSID)</p> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 22.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>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.</p> </div>

		<div>Source: NVMe Base Specification, at 28.</div> <div><div>Figure 88: Common Command Format</div><table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table></div> <div>Source: NVMe Base Specification, at 107.</div>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Bytes	Description					
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>					
14b	the storage device further comprises circuitry to:	Samsung’s Accused Instrumentalities comprise a storage device that further comprises circuitry to perform what is described in elements [14c]-[14f]. See claim elements [14c]-[14f].				
14c	store metadata on an independent basis for flash memory associated with each one of the	<p>Samsung’s Accused Instrumentalities comprise circuitry to store metadata on an independent basis for 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.</p> <p>Samsung’s Accused Instrumentalities are able to store metadata for each namespace. The metadata includes metadata dependent on an erase count of physical memory space associated with the respective namespace. See:</p>				

	<p>block devices, wherein the stored metadata is dependent on an erase count of associated physical memory space,</p>	<p>After a sub-block erase operation, the erase manage module 43a 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 43a may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module 43a may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module 43a may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module 43a may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module 43a 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 43a 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 43a.</p> <p>Source: Samsung's '250 Patent at 5:16–38.</p>	
--	---	---	--



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: Samsung Over-Provisioning White Paper, at 6.

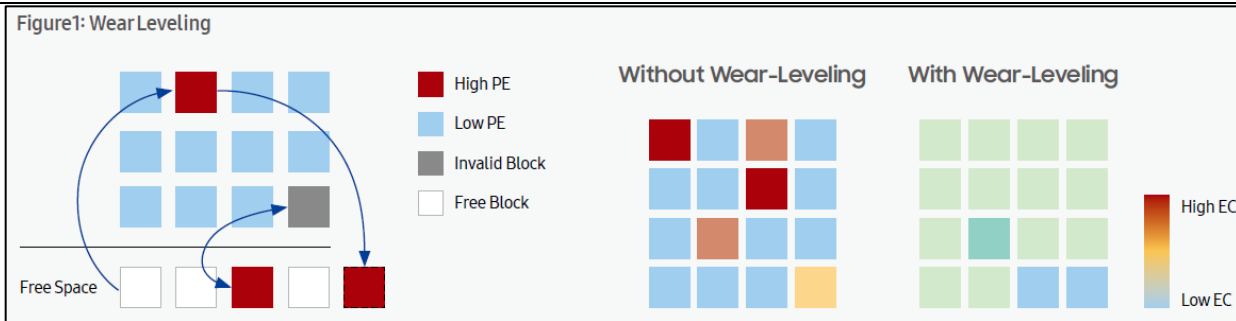
Bad Block Management & Error Correcting Code (ECC)

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.

		<div data-bbox="552 207 768 248"> <h3>Wear Leveling</h3> </div> <div data-bbox="552 272 1766 505"> <p>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.</p> </div> <div data-bbox="552 574 682 615"> <h3>Wrap Up</h3> </div> <div data-bbox="552 639 1766 769"> <p>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.</p> </div> <div data-bbox="552 828 793 868"> <h3>Why Samsung?</h3> </div> <div data-bbox="552 893 1766 1159"> <p>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.</p> </div> <div data-bbox="533 1208 1073 1243"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>
<i>14d</i>	compare the metadata to at least one threshold,	<p>Samsung's Accused Instrumentalities comprise circuitry to compare the metadata to at least one threshold.</p> <p>Samsung's Accused Instrumentalities are able to compare the metadata to at least one threshold. The metadata includes metadata dependent on an erase count of physical memory space associated with the respective namespace. See:</p>

		<p>After a sub-block erase operation, the erase manage module 43a 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 43a may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module 43a may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module 43a may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module 43a may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module 43a 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 43a 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 43a.</p>	
		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: Samsung Over-Provisioning White Paper, 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.

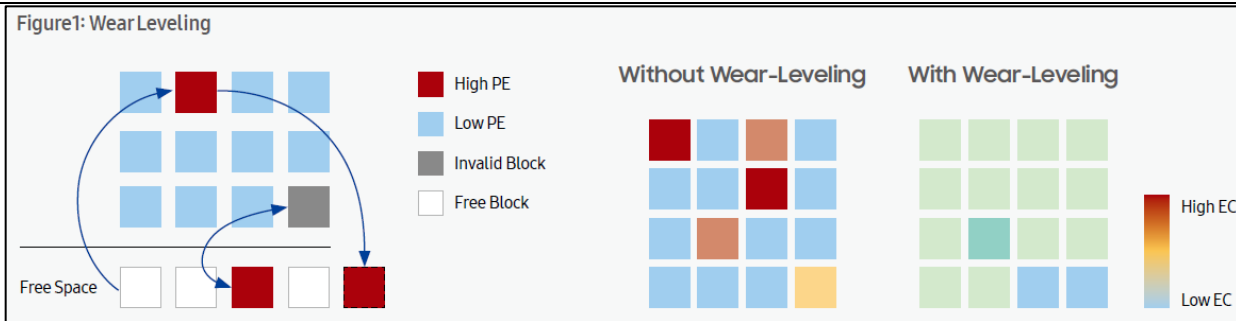
		<div> <h3>Wear Leveling</h3> <p>NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles depends on the process technology (e.g. 27nm, 21nm, 19 nm, etc.) 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.</p> </div> <div> <h3>Wrap Up</h3> <p>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.</p> </div> <div> <h3>Why Samsung?</h3> <p>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.</p> </div> <p>Source: Samsung SSD White Paper, at 19.</p>
--	--	--

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>
14e	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 one of the physical erase units for which the failure condition has not been detected by said storage	<p>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 one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device.</p> <p>Samsung's Accused Instrumentalities are able to perform internal operations to copy data from one physical storage location at one physical erase unit to another physical erase unit, where a failure condition has not been detected for either physical erase unit. 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:</p> <div> <p>5.4 Reset Zone Recommended</p> <p>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</p> </div>

<p>device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and</p>	<p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:</p> <ul style="list-style-type: none"> 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. <p>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'.</p> <p>Source: ZNS Specification, at 41–42.</p>
---	--

	<p>After a sub-block erase operation, the erase manage module 43a 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 43a may sense memory cells of the erased sub-block to determine whether specific parameters exceed a reference cycle value. The erase manage module 43a may read data of sub-block(s) adjacent to the erased sub-block to detect erase-inhibition efficiency. For example, the erase manage module 43a may detect bit error rate (BER) based on data read from an erased sub-block. The erase manage module 43a may acquire and monitor wear-leveling information (e.g., erase count) on the erased sub-block. In addition, the erase manage module 43a 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 43a 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 43a.</p>	
--	--	--

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: Samsung Over-Provisioning White Paper, 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.

		<div data-bbox="552 207 768 248"> <h3>Wear Leveling</h3> </div> <div data-bbox="552 272 1766 505"> <p>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.</p> </div> <div data-bbox="552 574 682 615"> <h3>Wrap Up</h3> </div> <div data-bbox="552 639 1766 769"> <p>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.</p> </div> <div data-bbox="552 828 793 868"> <h3>Why Samsung?</h3> </div> <div data-bbox="552 893 1766 1159"> <p>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.</p> </div> <div data-bbox="533 1208 1073 1243"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

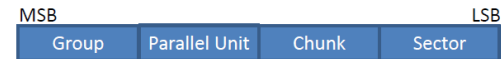
		<p>2. Why is OP important?</p> <p>OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process 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.</p> <p>Source: Samsung Application Note, at 2.</p>
14f	<p>remap the logical erase unit corresponding to the first physical erase unit for which the failure condition has not been detected by said storage device to the second physical erase unit for which the failure condition has not been detected by said storage device.</p>	<p>Samsung's Accused Instrumentalities comprise circuitry to remap the logical erase unit corresponding to the first physical erase unit for which the failure condition has not been detected by said storage device to the second physical erase unit for which the failure condition has not been detected by said storage device.</p> <p>Samsung's Accused Instrumentalities are able to remap the correspondence from the logical erase unit from a first physical erase unit for which the failure condition has not been detected to a second physical erase unit for which the failure condition has not been detected in association with the internal operations so that the location of data is properly tracked after the copy. See:</p> <div> <p>5.4 Reset Zone Recommended</p> <p>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</p> </div>

		<p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:</p> <ul style="list-style-type: none"> 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. <p>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'.</p>	
		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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

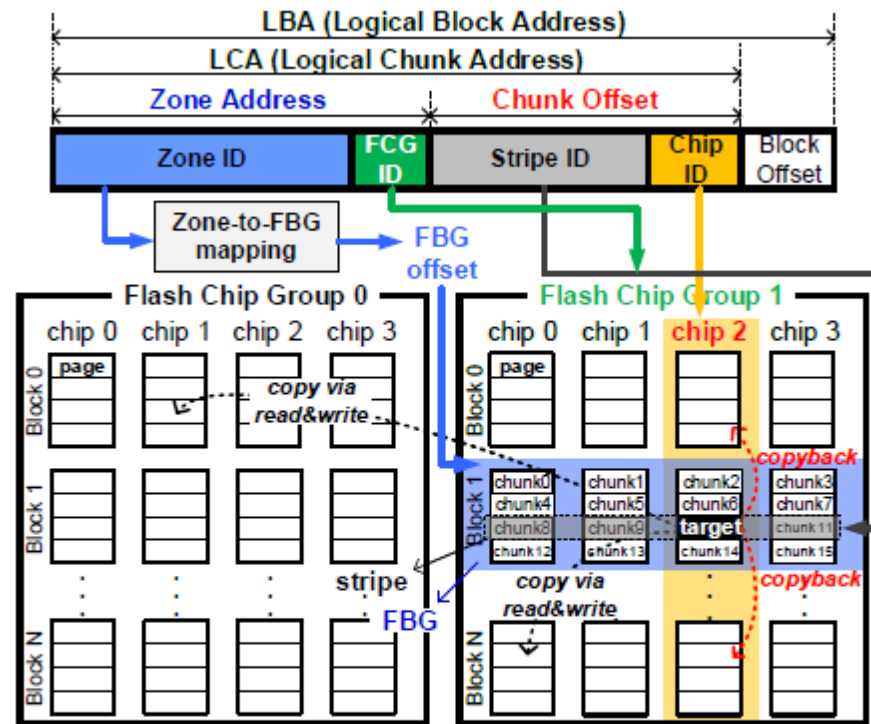


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

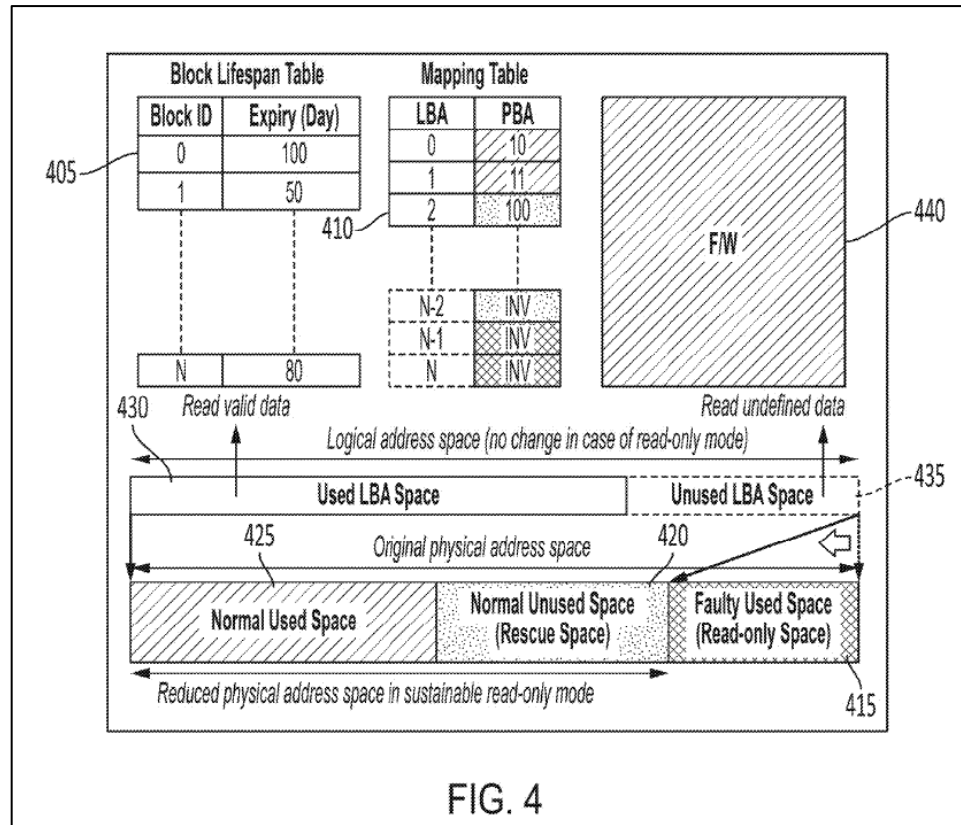
To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein

		<p>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.</p> <p>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.</p>
15 <i>pr</i> <i>e</i>	The storage device of claim 1 wherein:	Samsung's Accused Instrumentalities comprise the storage device of claim 1. See claim 1.
15 <i>a</i>	the flash memory is to have block devices, including the addressed block device; and	<p>Samsung's Accused Instrumentalities comprise flash memory that has block devices, including the addressed block device.</p> <p>Samsung's Accused Instrumentalities have namespaces (block devices) including the addressed namespace (the addressed block device). See:</p> <div data-bbox="533 784 1837 1068" data-label="Text" style="border: 1px solid black; padding: 10px;"> <p>1.5.37 namespace</p> <p>A formatted quantity of non-volatile memory that may be directly accessed by a host.</p> <p>1.5.38 namespace ID (NSID)</p> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 22.</p> <div data-bbox="533 1154 1818 1352" data-label="Text" style="border: 1px solid black; padding: 10px;"> <p>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.</p> </div>

		<div>Source: NVMe Base Specification, at 28.</div> <div><div>Figure 88: Common Command Format</div><table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table></div> <div>Source: NVMe Base Specification, at 107.</div>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Bytes	Description					
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>					
15b	the storage device further comprises circuitry to:	Samsung’s Accused Instrumentalities comprise a storage device that further comprises circuitry to perform what is described in elements [15c]-[15e]. See claim elements [15c]-[15e].				
15c	store metadata on an independent basis for flash memory associated with each one of the	<p>Samsung’s Accused Instrumentalities comprise circuitry to store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation.</p> <p>Samsung’s Accused Instrumentalities are able to store metadata for each namespace. The metadata includes metadata representing a time since data was programmed in one or more physical storage</p>				

block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,

locations associated with the namespace, including metadata such as age of data or data retention period. See:

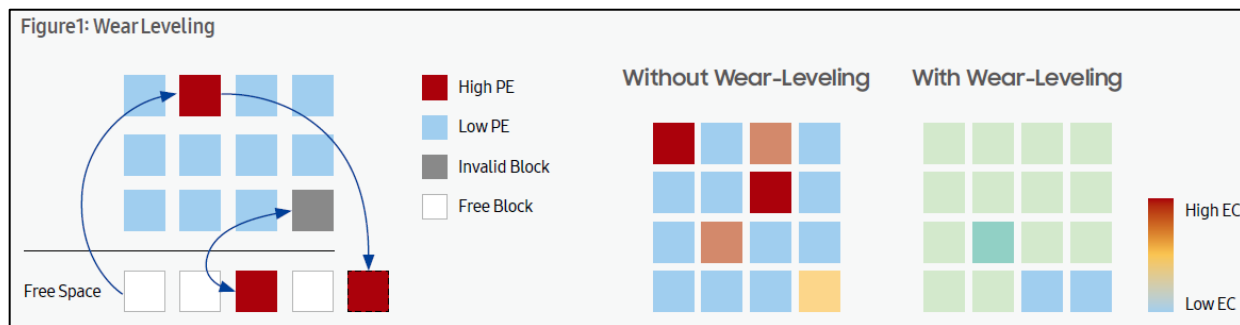


		<p>[0067] The storage media 125 may have a retention period (which may depend on the usage history of the storage media 125, and, as such, may vary within the storage media 125); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become 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.</p>	
--	--	--	--

		<p>[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.</p>	
--	--	---	--

[0090] FIG. 5A 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. 5B 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: Samsung Over-Provisioning White Paper, 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.

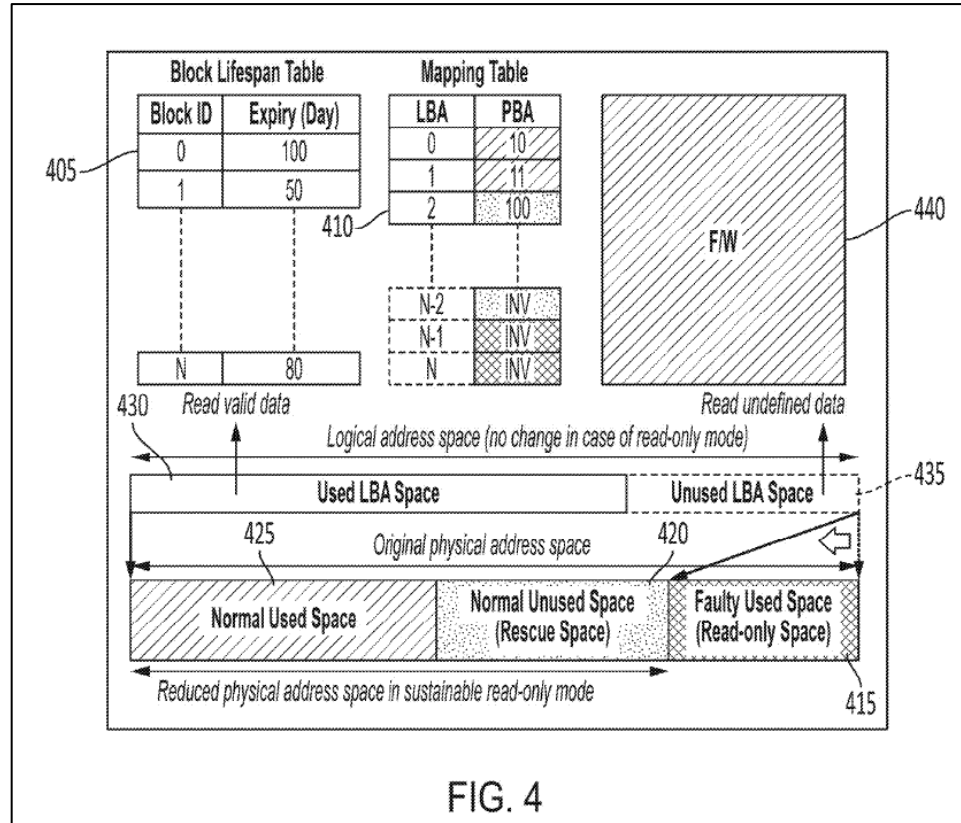
		<div data-bbox="552 207 768 248"> <h3>Wear Leveling</h3> </div> <div data-bbox="552 272 1766 505"> <p>NAND flash memory suffers from one final limitation: each cell has a finite lifespan and can only withstand a limited number of program/erase cycles (called P/E cycles). The specific amount of P/E cycles 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.</p> </div> <div data-bbox="552 574 682 615"> <h3>Wrap Up</h3> </div> <div data-bbox="552 639 1766 773"> <p>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.</p> </div> <div data-bbox="552 828 793 868"> <h3>Why Samsung?</h3> </div> <div data-bbox="552 893 1766 1159"> <p>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.</p> </div> <div data-bbox="533 1208 1073 1243"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p> <p>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.</p> <p>Source: Samsung Tech Blog.</p>	
--	--	---	--

		<p>2.2.3. Reliability limitation</p> <p>A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (<i>bitflips</i>) 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].</p> <p>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].</p> <p>Source: Boukhobza, at 26.</p>	
15d	compare the metadata to at	<p>Samsung's Accused Instrumentalities comprise circuitry to compare the metadata to at least one threshold.</p> <p>Samsung's Accused Instrumentalities are able to compare the metadata to at least one threshold. The metadata includes metadata representing a time since data was programmed in one or more physical</p>	

least one threshold,

storage locations associated with the namespace, including metadata such as age of data or data retention period. See:

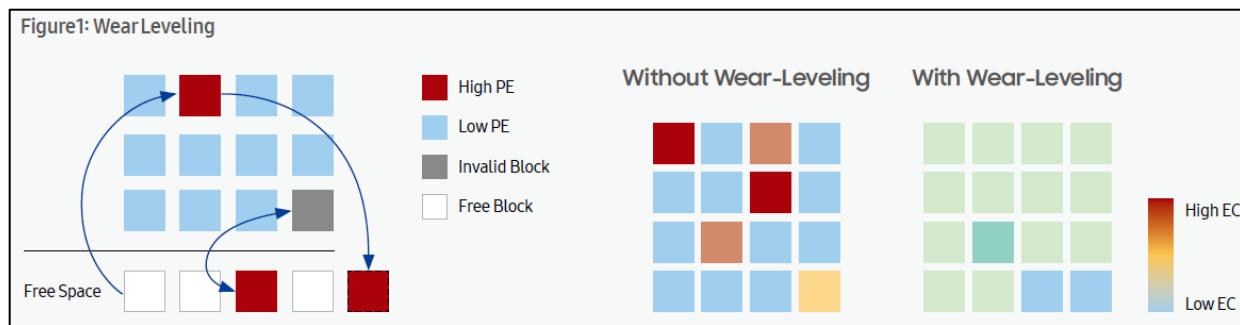


		<p>[0067] The storage media 125 may have a retention period (which may depend on the usage history of the storage media 125, and, as such, may vary within the storage media 125); data that has been stored longer than the retention period (i.e., data having an age exceeding the retention period) may become 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.</p>	
--	--	--	--

		<p>[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.</p>	
--	--	---	--

[0090] FIG. 5A 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. 5B 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: Samsung Over-Provisioning White Paper, 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.

		<div data-bbox="554 207 768 245" data-label="Section-Header"> <h3>Wear Leveling</h3> </div> <div data-bbox="554 272 1766 505" data-label="Text"> <p>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.</p> </div> <div data-bbox="554 574 682 612" data-label="Section-Header"> <h3>Wrap Up</h3> </div> <div data-bbox="554 639 1766 771" data-label="Text"> <p>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.</p> </div> <div data-bbox="554 829 793 867" data-label="Section-Header"> <h3>Why Samsung?</h3> </div> <div data-bbox="554 894 1766 1159" data-label="Text"> <p>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.</p> </div> <div data-bbox="535 1208 1073 1245" data-label="Text"> <p>Source: Samsung SSD White Paper, at 19.</p> </div>
--	--	---

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p> <p>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.</p> <p>Source: Samsung Tech Blog.</p>	
--	--	---	--

		<p>2.2.3. Reliability limitation</p> <p>A flash memory chip is unreliable in itself. Actually, during read and write operations, phenomena of bit inversion (<i>bitflips</i>) 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].</p> <p>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].</p> <p>Source: Boukhobza, at 26.</p>	
15e	responsive to detection that specific metadata has	Samsung's Accused Instrumentalities comprise circuitry to, responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted	

<p>satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.</p>	<p>information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.</p> <p>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.</p> <p>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. The data is stored at one of the physical storage locations within the zone associated with the specific metadata.</p> <p>See:</p> <div data-bbox="537 711 1885 850"> <p>5.4 Reset Zone Recommended</p> <p>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</p> </div>
---	---

		<p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:</p> <ul style="list-style-type: none"> 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. <p>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'.</p>
		Source: ZNS Specification, at 41–42.

Figure 37: Zone Descriptor Data Structure

Bytes	Description												
02	Zone Attributes (ZA): Indicates attributes for the Zone:												
	<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7</td><td>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.</td></tr><tr><td>6:3</td><td>Reserved</td></tr><tr><td>2</td><td>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.</td></tr><tr><td>1</td><td>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.</td></tr><tr><td>0</td><td>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.</td></tr></table>	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.
	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: ZNS Specification, 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.

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

Value	Description
EFh	<p>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'.</p> <p>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:</p> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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: ZNS Specification, 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: ZNS Specification, at 27–28.

		<p>Memory density and die floor plan</p> <p>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 mm² while the 48L V-NAND die measures 99.8 mm², 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/mm². The memory density of leading-edge 2D planar NAND devices such as Toshiba 15 nm TLC NAND is 1.28 Gb/mm², (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 mm² to 68.7 mm² 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.</p> <p>Source: EETimes V-NAND Analysis (highlighted).</p>	
--	--	--	--

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).

		<p>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 <i>zone interleaving degree</i> 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.</p> <p>Source: Han, at 149.</p>
19b	the storage device is to, for each one of a set of the incoming data access requests having sequential addressing, identify a commonly addressed page and write data accompanying	<p>Samsung's Accused Instrumentalities comprise the storage device that is to, for each one of a set of the incoming data access requests having sequential addressing, identify a commonly addressed page and write data accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes.</p> <p>Samsung's Accused Instrumentalities are able to receive and execute Write commands, which are incoming data access requests. See claim 1.</p> <p>A Write command has sequential addressing. See:</p> <div> <p>3.3.7 Write command</p> <p>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).</p> </div>

	<p>the set of incoming data access requests concurrently into respective ones of the multiple planes.</p>	<p>Source: ZNS Specification, at 20.</p> <div data-bbox="537 256 1843 902" style="border: 1px solid black; padding: 10px;"> <p>2.1.1.2 Zone Types</p> <p>2.1.1.2.1 Sequential Write Required Zones</p> <p>A zone type of Sequential Write Required requires the set of logical block addresses of a zone to be written sequentially.</p> <p>2.1.1.2.1.1 Writing in Sequential Write Required Zones</p> <p>The following commands may be used to write to logical blocks in a specific zone of zone type Sequential Write Required:</p> <ul style="list-style-type: none"> • Write command; • Write Zeroes command; • Write Uncorrectable command; • Copy command; and • Zone Append command. <p>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.</p> </div> <p>Source: ZNS Specification, at 9.</p> <p>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. See:</p>
--	---	--

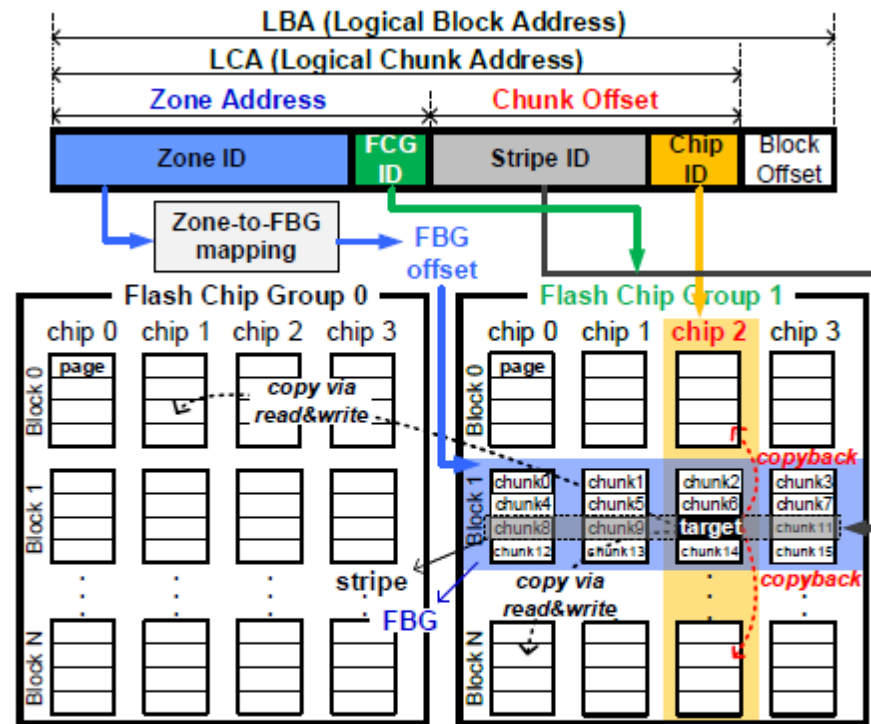
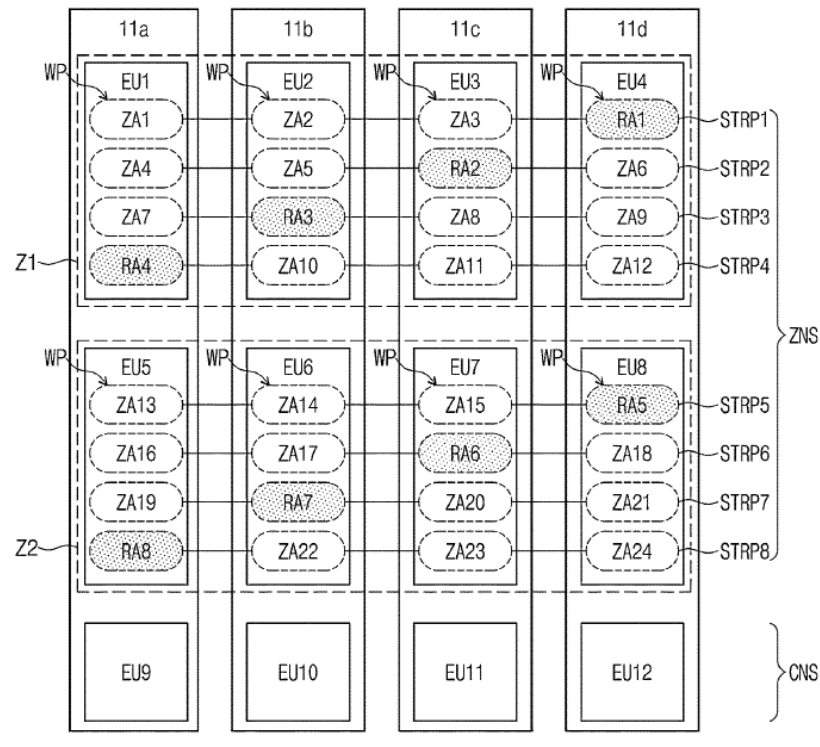


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

		<p>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 <i>zone interleaving degree</i> 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.</p>	
--	--	--	--

Source: Han, at 149.

FIG. 7

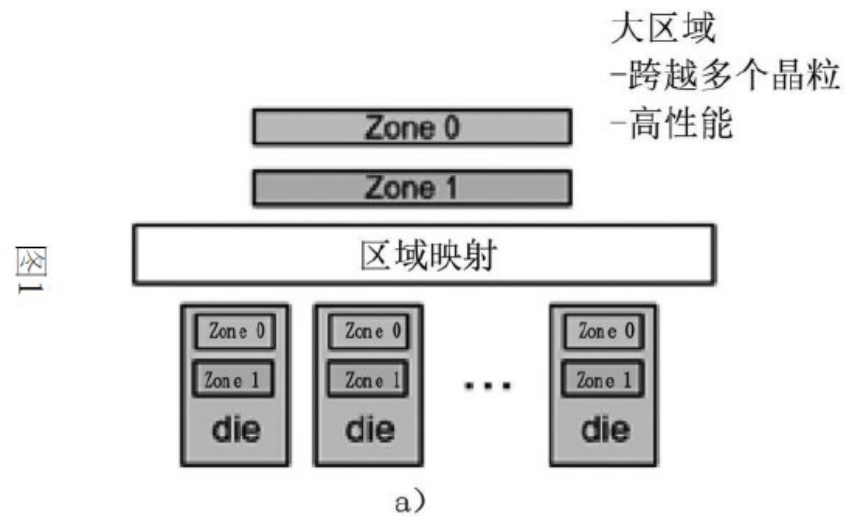


		<p>[0027] Each of the plurality of storage devices 11a, 11b, 11c, and 11d may include a nonvolatile memory device configured to retain data stored therein even when a power is turned off. For example, each of the plurality of storage devices 11a, 11b, 11c, and 11d may be (or include) a solid state drive. 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 11a, 11b, 11c, and 11d may include a first storage area ZNS and a second storage area CNS.</p> <p>[0028] The first storage area ZNS may be used as a zoned namespace. Each of the plurality of storage devices 11a, 11b, 11c, and 11d 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., RAID5 and/or RAID 4) standard.</p>	
--	--	---	--

		<p>[0113] FIG. 7 illustrates an example in which a first zone Z1 and a second zone Z2 are generated in the storage devices 11a, 11b, 11c, and 11d. Referring to FIGS. 1, 2, and 7, in at least one embodiment, one zone may correspond to four erase units.</p> <p>[0114] The storage devices 11a, 11b, 11c, and 11d may assign the first to fourth erase units EU1 to EU4 respectively included in the storage devices 11a, 11b, 11c, and 11d to the first zone Z1. Continuous logical addresses may be assigned to the first to 12th zone areas ZA1 to ZA12 that are included in the first to fourth RAID stripes STRP1 to STRP4 defined in the first to fourth erase units EU1 to EU4.</p> <p>[0115] Continuous logical addresses may be assigned to each of the first to fourth RAID areas RA1 to RA4 that are included in the first to fourth RAID stripes STRP1 to STRP4 defined in the first to fourth erase units EU1 to EU4. 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.</p>
--	--	--

		<p>[0116] A write pointer WP of the first storage device 11a 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 ZA1. Data may be written in a storage space of the first storage device 11a, 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 11a may update the write pointer WP to point out the first (or least significant bit (LSB)) logical address belonging to the first storage device 11a from among the logical addresses of the first zone Z1.</p> <p>[0117] Likewise, the write pointer WP of the second storage device 11b may point out the second zone area ZA2, and the write pointer WP of the third storage device 11c may point out the third zone area ZA3.</p> <p>[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 RA1 are continuous to the logical addresses of the third zone area ZA3, the write pointer WP of the fourth storage device 11d may point out the first RAID area RA1.</p> <p>[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</p>	
--	--	---	--

		<p>11d may skip the first RAID area RA1 and may point out the sixth zone area ZA6 (or may jump to the sixth zone area ZA6).</p> <p>[0120] The storage devices 11a, 11b, 11c, and 11d may assign the fifth to eighth erase units EU5 to EU8 respectively included in the storage devices 11a, 11b, 11c, and 11d to the second zone Z2. Continuous logical addresses may be assigned to the 13th to 24th zone areas ZA13 to ZA24 that are included in the fifth to eighth RAID stripes STRP5 to STRP8 defined in the fifth to eighth erase units EU5 to EU8.</p> <p>[0121] Continuous logical addresses may be assigned to each of the fifth to eighth RAID areas RA5 to RA8 that are included in the fifth to eighth RAID stripes STRP5 to STRP8 defined in the fifth to eighth erase units EU5 to EU8. 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.</p>	
		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”

		<div data-bbox="537 370 1751 600"> <p>[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。</p> </div> <div data-bbox="617 626 787 660"> <p><u>Translation:</u></p> </div> <div data-bbox="617 698 1671 987"> <p>“[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.”</p> </div> <div data-bbox="537 1065 1751 1295"> <p>[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。</p> <p>[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。</p> </div> <div data-bbox="617 1321 787 1356"> <p><u>Translation:</u></p> </div>
--	--	--

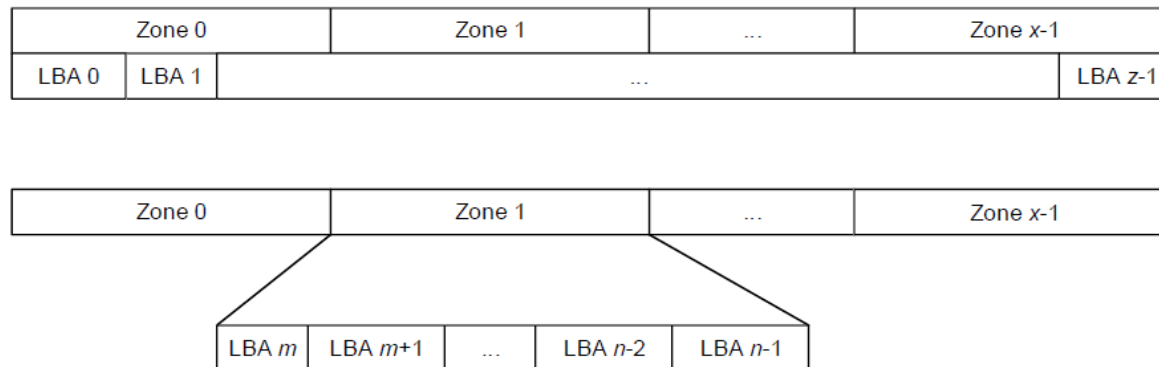
		<p>“[0042] Fig. 1 shows a schematic diagram of the relationship between the data transmission bandwidth of a zone and the size of the zone for a ZNS SSD.</p> <p>[0043] Referring to a) in Fig. 1, when the size of a ZNS SSD zone is large, it can span multiple dies, which can ensure a higher data transmission 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.”</p> <p>Source: Samsung’s ’816 Patent Application at Fig. 1a, [0003], [0042]–[0043].</p>
20	The storage device of claim 1 wherein the division operation is applied to subdivide the third address portion into respective first and second bit fields.	<p>Samsung’s Accused Instrumentalities comprise the storage device of claim 1. See claim 1.</p> <p>Samsung’s Accused Instrumentalities comprise the storage device of claim 1 wherein the division operation is applied to subdivide the third address portion into respective first and second bit fields.</p> <p>Samsung’s Accused Instrumentalities are able to subdivide the third address portion into respective first and second bit fields to identify a logical block, erase unit, page, and/or logical erase unit within the zone (addressed segment). See:</p>

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



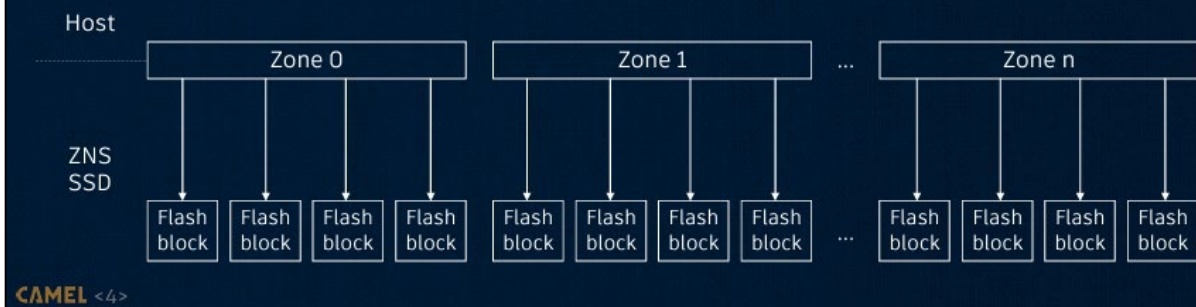
The Zoned Namespace Command Set is based on the NVM Command Set (refer to the NVM Command Set Specification).

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.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

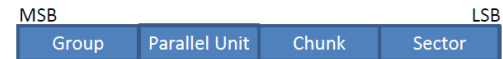


Source: Bae, at 4.

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

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

15

Source: Microsoft Denali Presentation, at 15.

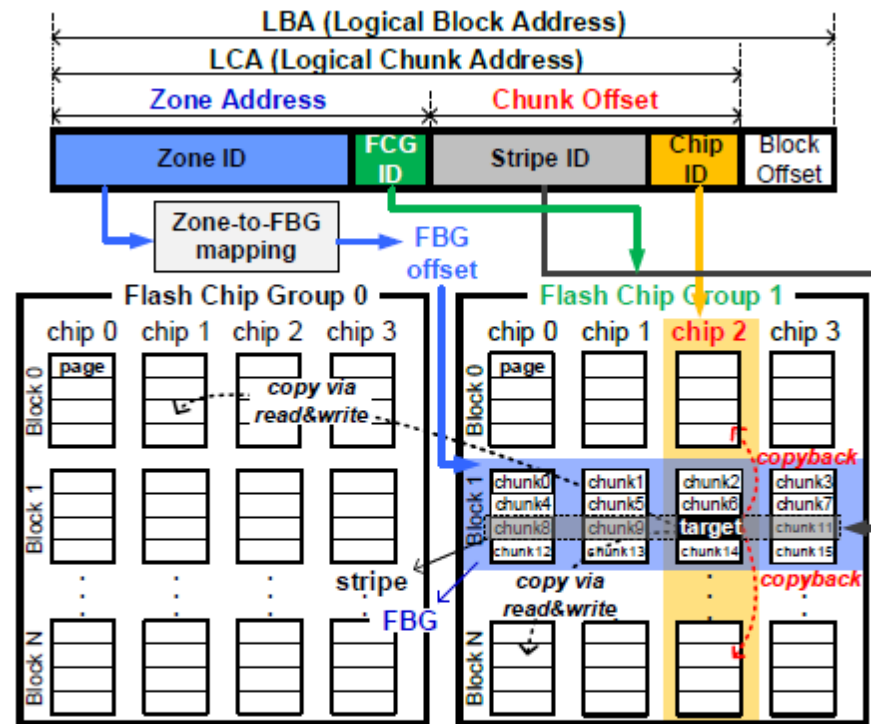


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

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 level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment including a mutually exclusive set of one or more logical erase units that each	<p>Samsung’s Accused Instrumentalities comprise a 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 including a mutually exclusive set of one or more logical erase units that each map to a physical erase unit, each segment being associated with only one of the block devices, each logical erase unit being associated with only one of the segments.</p> <p>Samsung’s Accused Instrumentalities have NAND flash memory.</p> <p>For example, Samsung’s PM1733 contains NAND flash memory. See:</p> <table><tr><th colspan="2">Samsung PM1733 specifications</th></tr><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> <p>Source: Samsung PM1733 Product Brief (annotated).</p> <p>Samsung’s Accused Instrumentalities comprise a flash memory that has one or more namespaces (block devices) at a hierarchically-superior level. See:</p>	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®
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®													

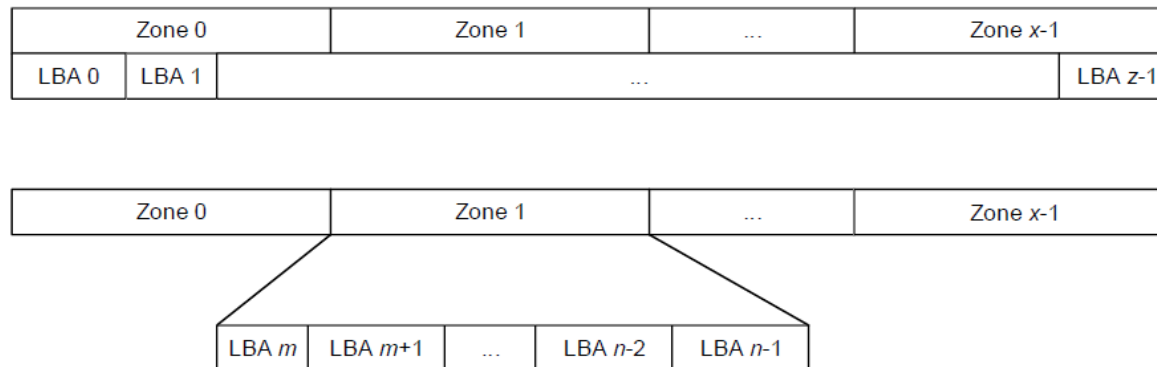
	<p>map to a physical erase unit, each segment being associated with only one of the block devices, each logical erase unit being associated with only one of the segments; and</p>	<div data-bbox="541 203 1837 480"> <p>1.5.37 namespace</p> <p>A formatted quantity of non-volatile memory that may be directly accessed by a host.</p> <p>1.5.38 namespace ID (NSID)</p> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 22.</p> <div data-bbox="541 565 1818 761"> <p>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.</p> </div> <p>Source: NVMe Base Specification, at 28.</p>
--	--	--

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.

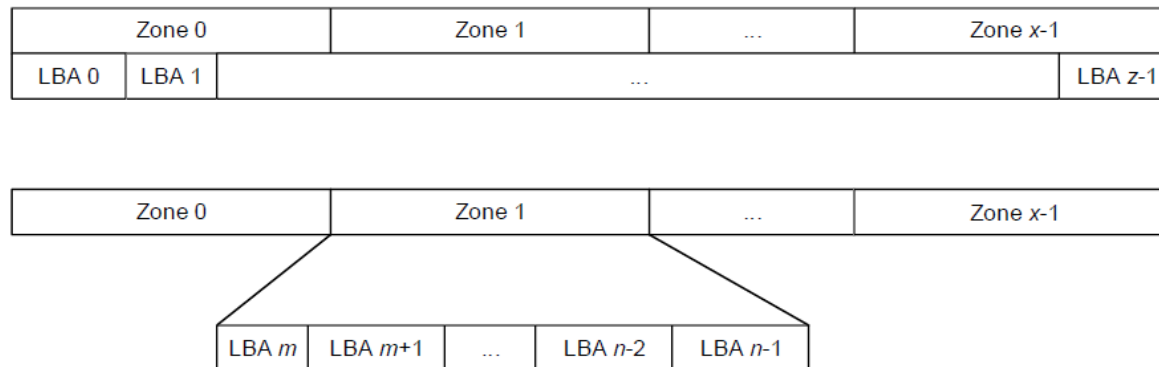
Samsung's Accused Instrumentalities comprise a flash memory that has addressable storage locations (including a logical block, page, and/or erase unit) within each of the segments (zones) at a hierarchically-inferior level. See:

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).

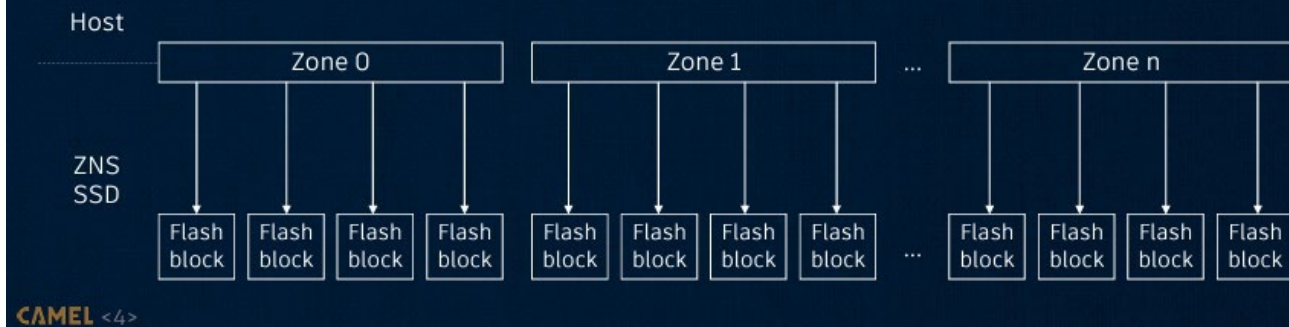
Each zone has an associated Zone Descriptor that contains a set of attributes. A Zone Management Receive command may be used to retrieve one or more Zone Descriptors.

Source: ZNS Specification, at 8.

		<p>1.7.1 logical block</p> <p>The smallest addressable data unit for Read and Write commands.</p> <p>1.7.2 logical block address (LBA)</p> <p>The address of a logical block, referred to commonly as LBA.</p>
		<p>Source: NVMe Base Specification, at 24.</p> <p>Each zone includes a mutually exclusive set of one or more logical erase units that each map to a physical erase unit. Each logical erase unit is associated with only one of the zones (segments). 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:</p>

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

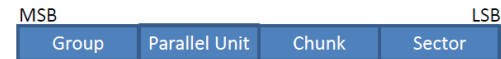


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

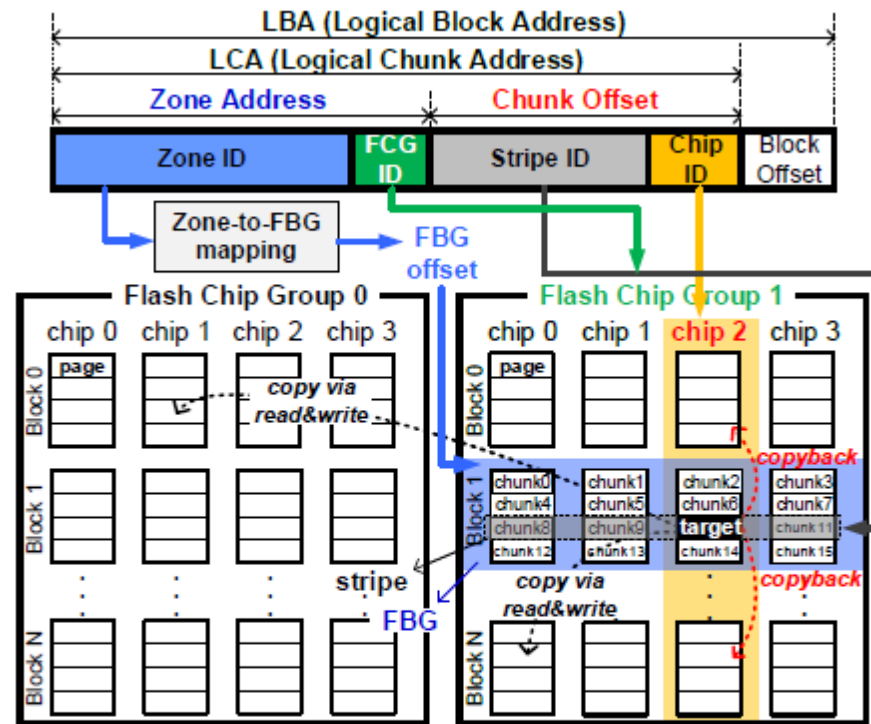


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

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

		<p>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.</p> <p>As an example, there is no substantial difference between (1) having a physical erase unit which maps to a logical erase unit and (2) having a physical erase unit which maps to the combination of a logical erase unit group and an offset within the erase unit group. The two ways of mapping perform substantially the same function in substantially the same way to achieve substantially the same result.</p>
21b	circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:	<p>Samsung's Accused Instrumentalities contain circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information.</p> <p>Samsung's Accused Instrumentalities are able to receive incoming Read commands from a host.</p> <p>A Read command is a data access request.</p> <p>A Read command reads data and metadata, if applicable, from the I/O controller for the logical block addresses ("LBAs") indicated. See:</p> <div data-bbox="535 784 1795 930" style="border: 1px solid black; padding: 5px;"> <p>3.3.4 Read command</p> <p>The Read command reads data and metadata, if applicable, from the I/O controller for the LBAs indicated. The command may specify protection information to be checked as part of the read operation.</p> </div> <p>Source: NVM Command Set Specification, at 48.</p> <p>The Read 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 from which data is to be read. The NSID and SLBA together are accompanying address information. See:</p>

		<p align="center">Figure 88: Common Command Format</p> <table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Bytes	Description					
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>					
		<p>Source: NVMe Base Specification, at 107.</p>				

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 52: Read – 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 read 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 49.

Samsung's Accused Instrumentalities are able to receive incoming Write commands from a host.

		<p>A Write command is a data access request.</p> <p>A Write command writes data and metadata, if applicable, to the I/O controller for the logical blocks indicated. See:</p> <div data-bbox="533 354 1864 506" style="border: 1px solid black; padding: 5px;"> <p>3.3.6 Write command</p> <p>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.</p> </div> <p>Source: NVM Command Set Specification at 53.</p> <p>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 accompanying address information. See:</p>
--	--	---

		Figure 88: Common Command Format					
		<table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>	
Bytes	Description						
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>						
		Source: NVMe Base Specification, at 107.					

		<table><tr><th colspan="6">Figure 12: Opcodes for Zoned Namespace Command Set I/O Commands</th></tr><tr><th colspan="3">Opcode by Field</th><th rowspan="3">Combined Opcode¹</th><th rowspan="3">Command²</th><th rowspan="3">Reference</th></tr><tr><th>(07)</th><th>(06:02)</th><th>(01:00)</th></tr><tr><th>Standard Command</th><th>Function</th><th>Data Transfer³</th></tr><tr><td colspan="3">Refer to the NVM Express Base Specification</td><td></td><td>Reservation Acquire</td><td>NVM Express Base Specification</td></tr><tr><td colspan="3">Refer to the NVM Express Base Specification</td><td></td><td>Reservation Release</td><td>NVM Express Base Specification</td></tr><tr><td colspan="6">NVM Command Set commands implemented by this specification</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Dataset Management</td><td>NVM Command Set Specification</td></tr><tr><td colspan="6">NVM Command Set commands modified by this specification</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Write</td><td>3.3.7</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Read</td><td>3.3.3</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Write Uncorrectable</td><td>3.3.8</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Compare</td><td>3.3.1</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Write Zeroes</td><td>3.3.9</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Verify</td><td>3.3.6</td></tr><tr><td colspan="3">Refer to the NVM Command Set Specification</td><td></td><td>Copy</td><td>3.3.2</td></tr></table>	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
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																																																																																							
		<p>Source: ZNS Specification, at 18 (annotated).</p> <table><tr><th colspan="2">Figure 69: Write – Command Dword 10 and Command Dword 11</th></tr><tr><th>Bits</th><th>Description</th></tr><tr><td>63:00</td><td>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.</td></tr></table> <p>Source: NVM Command Set Specification at 54.</p>	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.																																																																																				
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.																																																																																											
21c	derive, from the accompanying	Samsung’s Accused Instrumentalities comprise circuitry to derive, from the accompanying address information, a first address portion, a second address portion and a third address portion.																																																																																										

	address information, a first address portion, a second address portion and a third address portion,	<p>Samsung's Accused Instrumentalities are able to derive, from the accompanying address information, a Namespace Identifier or NSID (a first address portion), a zone identifier information (a second address portion), and an LBA and/or a zone offset (a third address portion). See:</p> <div data-bbox="535 334 1713 717"> <p>Additional Available Features</p> <ul style="list-style-type: none"> ▪ Fail-In-Place technology: Ensures the SSD operates normally even when errors occur at the chip level. ▪ Zoned Namespace: PM1733 is capable of supporting ZNS implementations ▪ FIPS compliance: PM1733 can be FIPS certified to satisfy government requirements. </div> <p>Source: Samsung PM1733 Product Brief (annotated).</p>
--	---	--

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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Source: NVMe Base Specification, at 107.

Figure 52: Read – Command Dword 10 and Command Dword 11

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

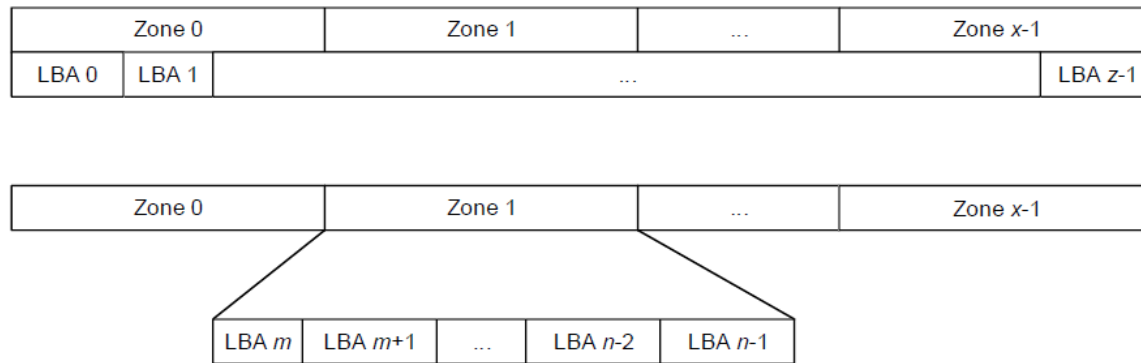
Source: NVM Command Set Specification, at 49.

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.

3.3.5 Read command

The Read 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).

3.3.5.1 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 17.

Figure 17: Read – Command Specific Status Values

Value	Description
B8h	Zone Boundary Error: The command specifies logical blocks in more than one zone.
BBh	Zone Is Offline: The accessed zone is in the ZSO:Offline state.

Source: ZNS Specification, at 20.

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.

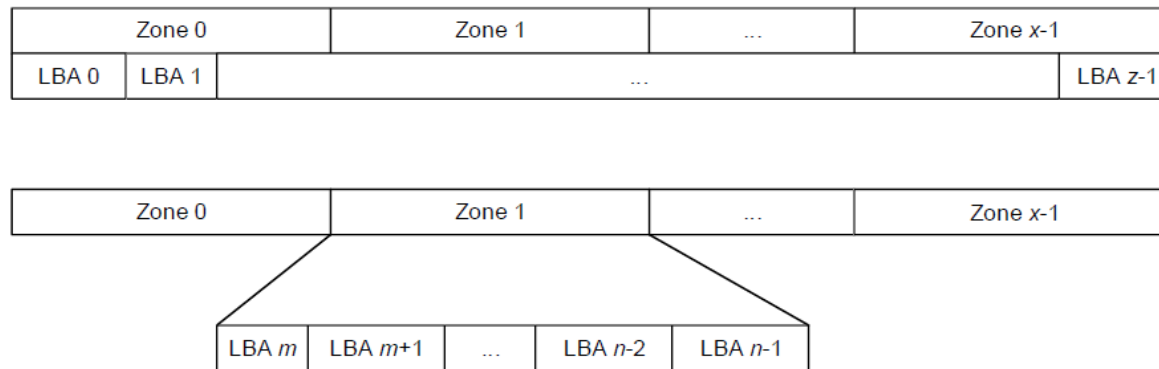
21d	identify an addressed block device from the first address portion,	<p>Samsung's Accused Instrumentalities contain said circuitry to identify an addressed block device from the first address portion.</p> <p>Samsung's Accused Instrumentalities are able to identify an addressed namespace (block device) from the Namespace Identifier or NSID (the first address portion). See:</p> <div><p style="text-align: center;">Figure 88: Common Command Format</p><table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table></div> <p>Source: NVMe Base Specification, at 107.</p>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Bytes	Description					
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>					
21e	identify an addressed segment within the addressed block device from the second address portion, and	<p>Samsung's Accused Instrumentalities contain said circuitry to identify an addressed segment within the addressed block device from the second address portion.</p> <p>Samsung's Accused Instrumentalities are able to identify an addressed zone (segment) within the addressed namespace from the second address portion. See:</p>				

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.

3.3.5 Read command

The Read 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).

3.3.5.1 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 17.

Figure 17: Read – Command Specific Status Values

Value	Description
B8h	Zone Boundary Error: The command specifies logical blocks in more than one zone.
BBh	Zone Is Offline: The accessed zone is in the ZSO:Offline state.

Source: ZNS Specification, at 20.

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.

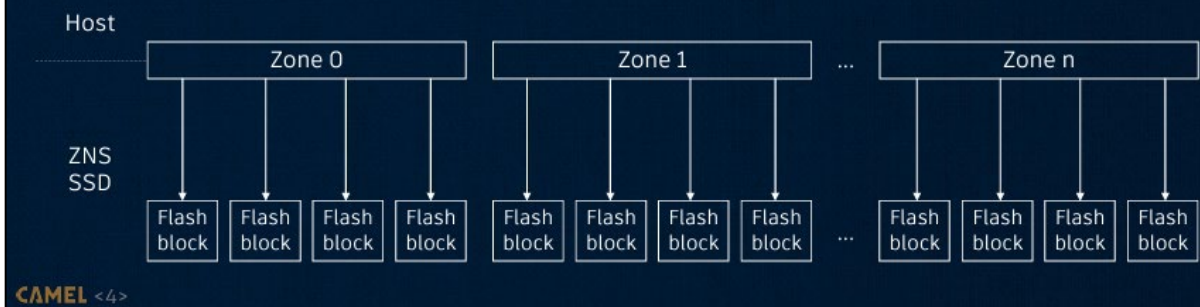
Figure 19: Write – Command Specific Status Values

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.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



Source: Bae, at 4.

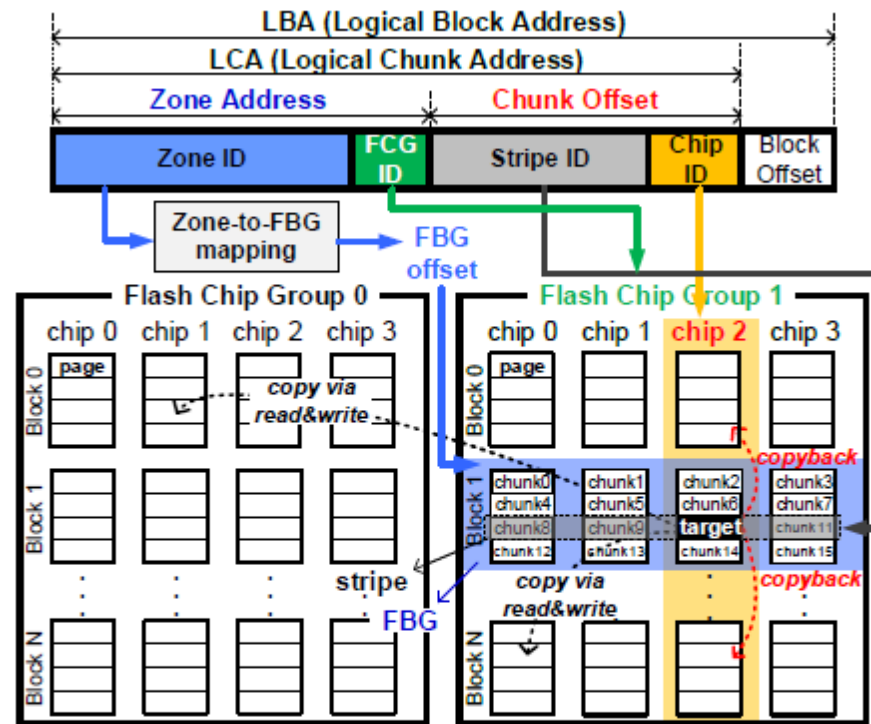


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

21f

identify an addressed storage

Samsung's Accused Instrumentalities contain said circuitry to identify an addressed storage location within the addressed segment from the third address portion.

location within the addressed segment from the third address portion,

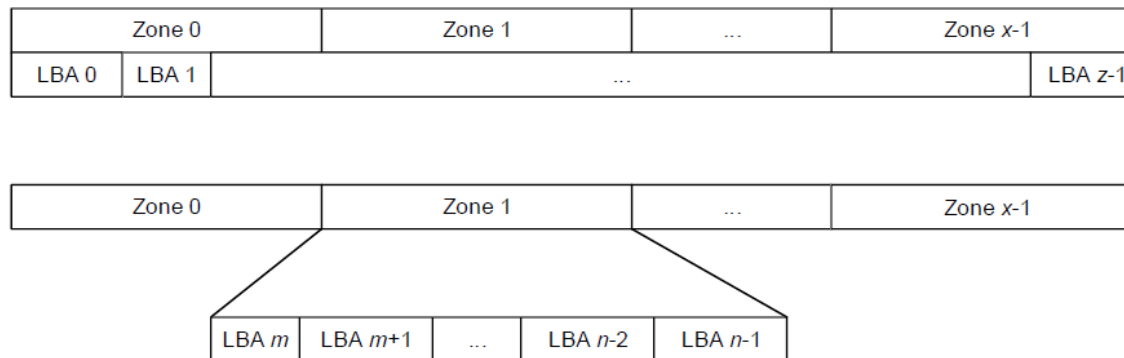
Samsung's Accused Instrumentalities are able to identify an addressed storage location, including a logical block, page, and/or erase unit, (addressed storage location) within the addressed zone (addressed segment) from the LBA and/or the zone offset (the third address portion). See:

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.

		<p>1.4.3.3 logical block</p> <p>The smallest addressable data unit for Read and Write commands.</p> <p>1.4.3.4 logical block address (LBA)</p> <p>The address of a logical block, referred to commonly as LBA.</p> <p>Source: NVM Command Set Specification, at 10.</p>	
--	--	---	--

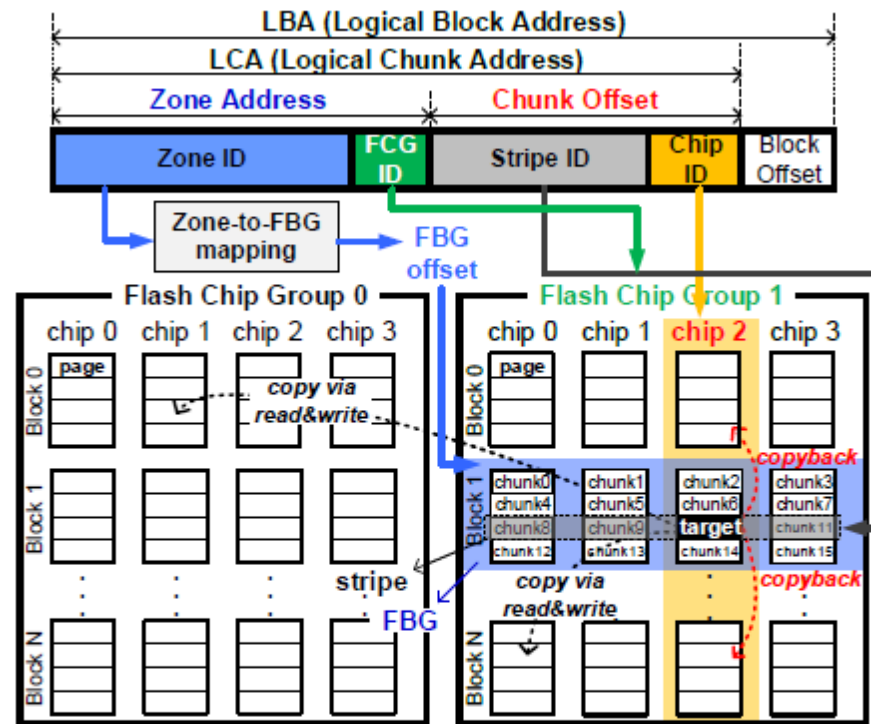
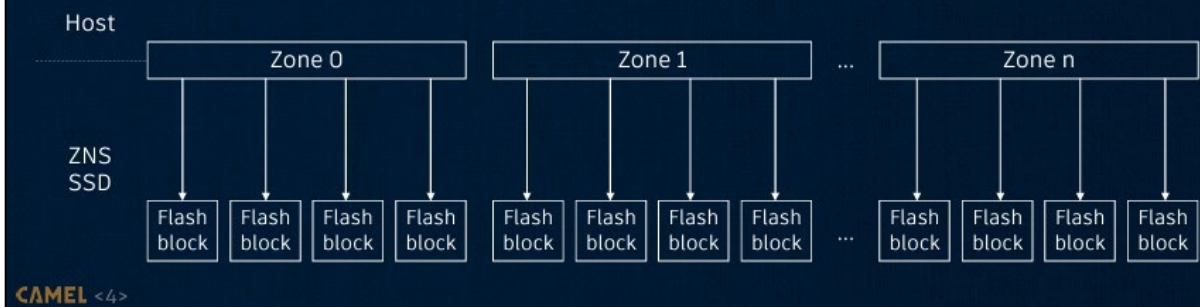


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

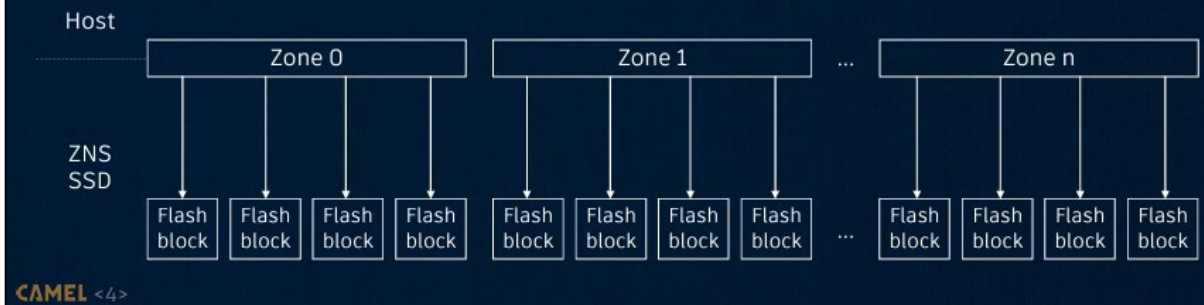


Source: Bae, at 4.

		<div><div><h2>Logical Hierarchical Addressing</h2><div><div><ul style="list-style-type: none">Each field maps to logical part of architecture<ul style="list-style-type: none">Flexibility in HW to manage NAND (such as mapping out bad blocks)System can implement 2-part wear leveling*Overheads significantly lower than conventional SSDsHost IO Requirements<ul style="list-style-type: none">Allocate a fresh a chunk before writing any sectorsWrite sectors within the chunk sequentiallySome new elements to abstract NAND management, for example, the cache minimum write size</div><div><div>Address Format:</div><div><div>MSB</div><div>Group</div><div>Parallel Unit</div><div>Chunk</div><div>Sector</div><div>LSB</div></div><div><div>Group: SSD Channel</div><div>Parallel Unit (PU): NAND Die</div><div>Chunk: multi-plane block</div><div>Sector: 512B or 4k region of NAND page</div></div></div></div><div><div>* "FlashBlox: Achieving Both Performance Isolation and Uniform Lifetime for Virtualized SSDs" Huang et al, USENIX-FAST 2017</div><div>15</div></div></div><div>Source: Microsoft Denali Presentation, at 15.</div></div>
21g	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,	<div><div>Samsung’s Accused Instrumentalities contain said circuitry to use a division operation to identify a logical erase unit within the addressed segment.</div><div>Samsung’s Accused Instrumentalities are able to use a division operation on at least one of the incoming address information or information derived therefrom to identify a logical erase unit within the addressed zone. Samsung’s Accused 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.</div><div>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:</div></div>

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

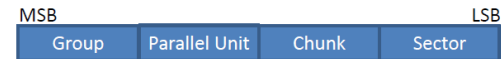


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

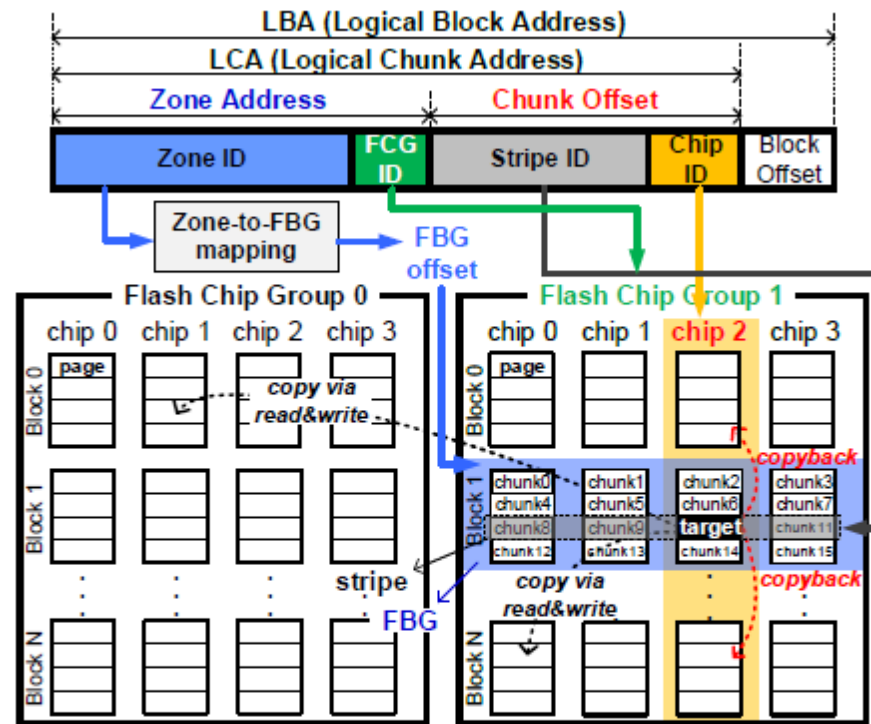


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

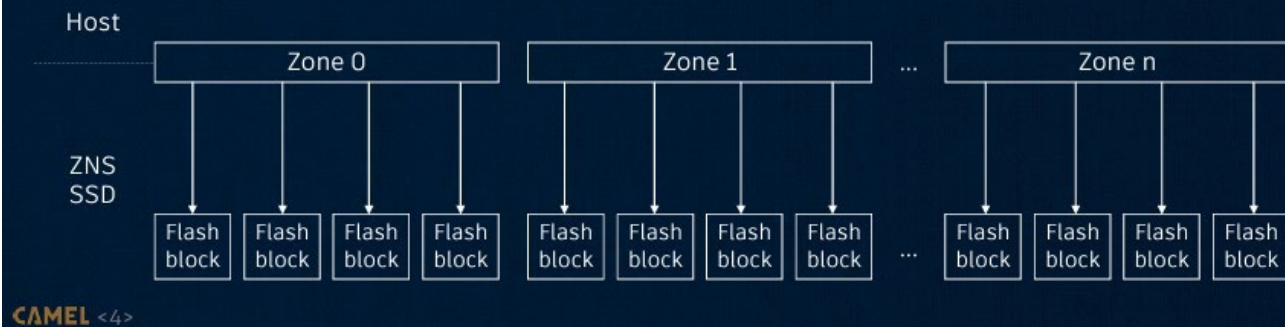
Source: Han, at 149.

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

		<p>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.</p> <p>As an example, there is no substantial difference between (1) identifying a logical erase unit within the addressed segment and (2) identifying a logical erase unit group within the addressed segment 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.</p> <p>As another example, there is no substantial difference between (1) identifying a logical erase unit within the addressed segment and (2) having the addressed segment as a logical erase unit group and 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.</p>
21h	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;	<p>Samsung's Accused Instrumentalities contain said circuitry to identify one of the physical erase units that corresponds to the identified logical erase unit.</p> <p>Samsung's Accused Instrumentalities are able to identify one of the physical erase units that corresponds to the identified logical erase unit. 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:</p>

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

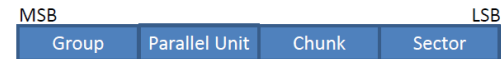


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

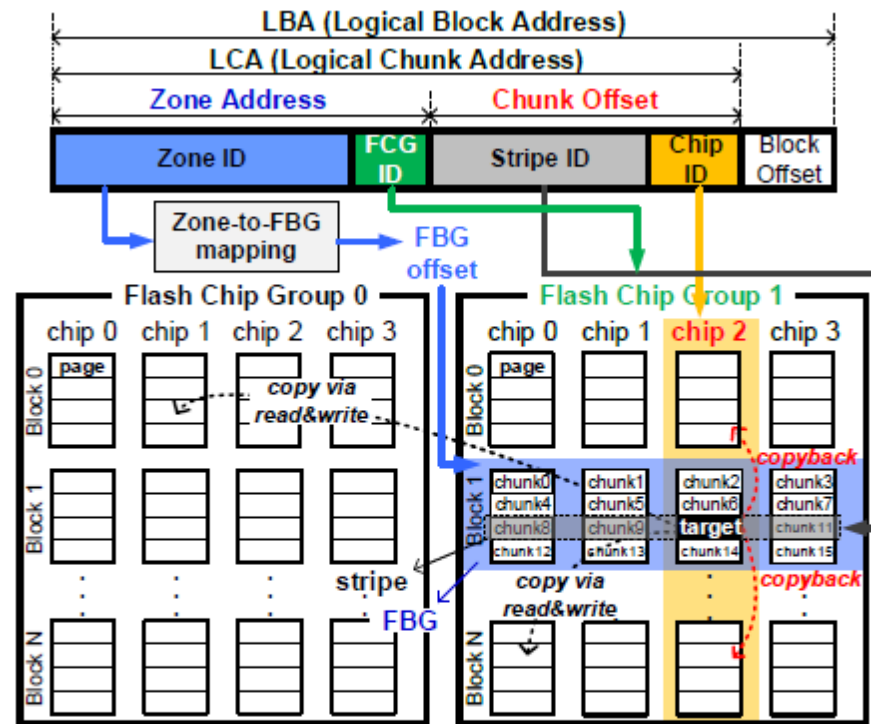


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

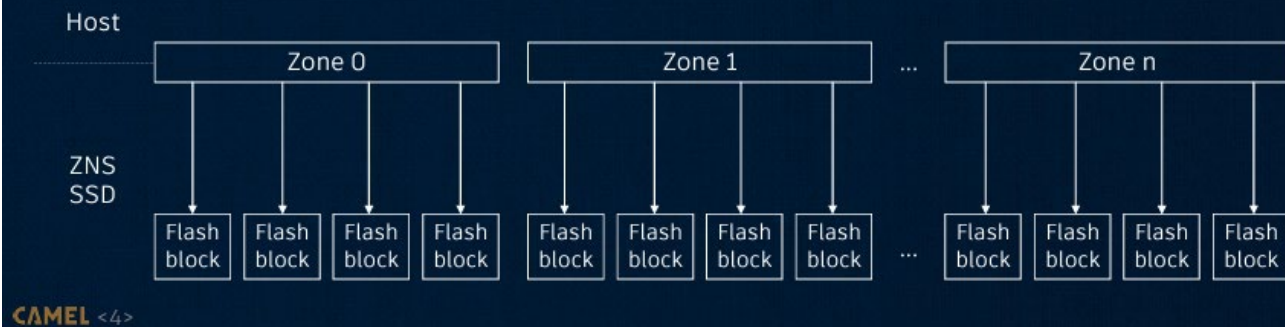
To the extent Samsung argues or the Court finds that this element is not literally satisfied by the disclosures cited herein, it is satisfied under the doctrine of equivalents by the disclosures cited herein

		<p>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.</p> <p>As an example, there is no substantial difference between (1) identifying a physical erase unit that corresponds to the identified logical erase unit and (2) identifying a physical erase unit group that corresponds to the identified 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.</p> <p>As another example, there is no substantial difference between (1) identifying a physical erase unit that corresponds to the identified logical erase unit and (2) having the addressed segment as a logical erase unit group which corresponds to a physical erase unit group and 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.</p>
21i	<p>wherein the storage device is to detect a failure 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</p>	<p>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 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.</p> <p>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 to physical correspondence accordingly and to utilize the remapped correspondence to service one or more of the incoming data access requests. See:</p>

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.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

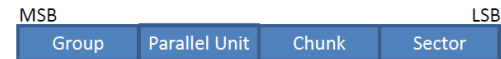


Source: Bae, at 4.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p>	
		<p>Source: Samsung Application Note, at 2.</p>	

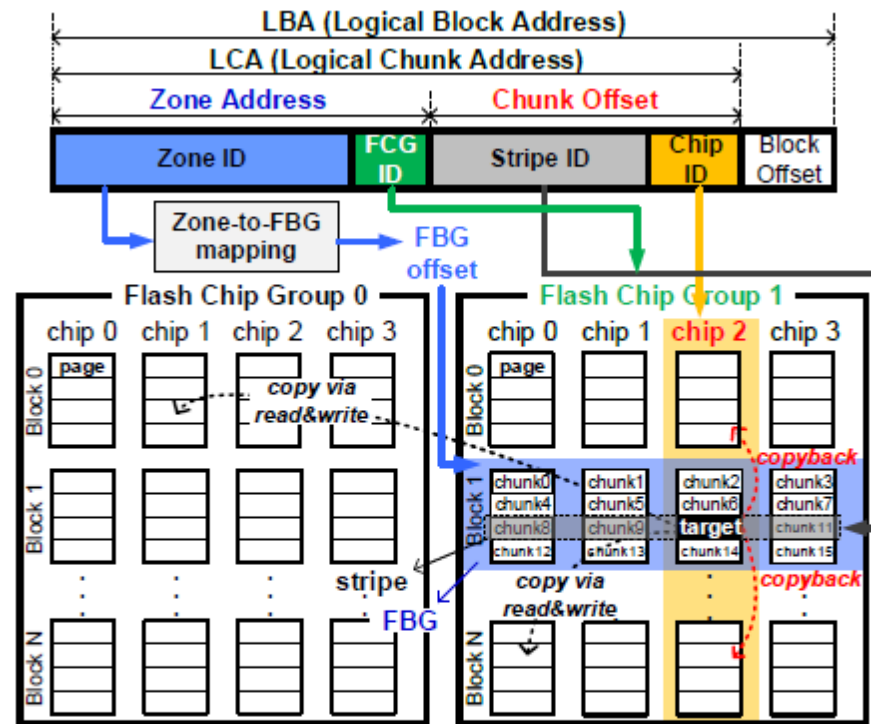


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

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

		<p>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.</p> <p>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.</p>												
23pre	A method of operation in a storage device having flash memory, the flash memory having physical erase units, and circuitry, the method comprising:	<p>To the extent the preamble is considered a limitation, Samsung’s Accused Instrumentalities include the method of operation in a storage device having flash memory, the flash memory having physical erase units, and circuitry.</p> <p>Samsung’s Accused Instrumentalities contain NAND flash memory having physical blocks (erase units) and a controller and firmware (circuitry).</p> <p>For example, Samsung’s PM1733 contains NAND flash memory. See:</p> <table><tr><th colspan="2">Samsung PM1733 specifications</th></tr><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> <p>Source: Samsung PM1733 Product Brief (annotated).</p>	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®
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®													

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.

The flash memory in Samsung's Accused Instrumentalities has 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).

		<p>2.2.1. Erase-before-write constraint</p> <p>Because of the internal architecture of a NAND chip, it is impossible to write data in a page that already contains data (<i>in-place</i> update). <u>Before performing any write operation on a page which already contains data, this page has to be erased.</u> However, the erase operation affects <u>an entire block</u>, 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</p> <p>Source: Boukhobza, at 24.</p>
23a	receiving incoming data access requests with the circuitry, wherein each of the incoming data access requests is accompanied by address information;	<p>Samsung's Accused Instrumentalities perform the step of receiving incoming data access requests with the circuitry, wherein each of the incoming data access requests is accompanied by address information.</p> <p>Samsung's Accused Instrumentalities are able to receive incoming Read commands from a host.</p> <p>A Read command is a data access request.</p> <p>A Read command reads data and metadata, if applicable, from the I/O controller for the logical block addresses ("LBAs") indicated. See:</p> <div data-bbox="537 987 1797 1133"> <p>3.3.4 Read command</p> <p>The Read command reads data and metadata, if applicable, from the I/O controller for the LBAs indicated. The command may specify protection information to be checked as part of the read operation.</p> </div> <p>Source: NVM Command Set Specification, at 48.</p> <p>The Read 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 from which data is to be read. The NSID and SLBA together are accompanying address information. See:</p>

		Figure 88: Common Command Format					
		<table><tr><th>Bytes</th><th>Description</th></tr><tr><td>07:04</td><td><p>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.</p><p>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.</p><p>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.</p><p>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.</p></td></tr></table>	Bytes	Description	07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>	
Bytes	Description						
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>						
		Source: NVMe Base 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 52: Read – 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 read 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 49.

Samsung's Accused Instrumentalities are able to receive incoming Write commands from a host.

A Write command is a data access 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 accompanying address information. See:

Figure 88: Common Command Format

Bytes	Description
07:04	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Source: NVMe Base Specification, at 107.

The Write command is accompanied by a logical block address (“LBA”). 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

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.

23b

with the circuitry, processing the incoming

Samsung’s Accused Instrumentalities perform the step of of processing, with the circuitry, the incoming address information to, for each of the incoming data access requests, perform what is described in elements [23c]–[23h]. See claim elements [23c]–[23h].

	address information to, for each of the incoming data access requests:	
23c	derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion,	<p>Samsung's Accused Instrumentalities perform the step of deriving a first address portion, a second address portion and a third address portion, and identifying an addressed block device from the first address portion.</p> <p>Samsung's Accused Instrumentalities are able to derive, from the accompanying address information, a Namespace Identifier or NSID (a first address portion), a zone identifier information (a second address portion), and an LBA and/or a zone offset (a third 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:</p> <div data-bbox="533 784 1703 1167" data-label="Complex-Block"> <p>Additional Available Features</p> <ul style="list-style-type: none"> • Fail-In-Place technology: Ensures the SSD operates normally even when errors occur at the chip level. • Zoned Namespace: PM1733 is capable of supporting ZNS implementations • FIPS compliance: PM1733 can be FIPS certified to satisfy government requirements. </div> <p>Source: Samsung PM1733 Product Brief (annotated).</p>

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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>

Source: NVMe Base Specification, at 107.

Figure 52: Read – Command Dword 10 and Command Dword 11

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

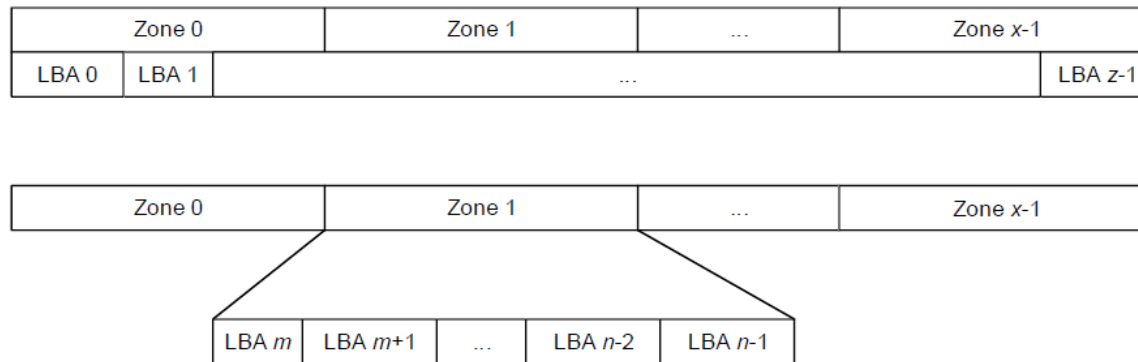
Source: NVM Command Set Specification, at 49.

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.

3.3.5 Read command

The Read 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).

3.3.5.1 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 17.

Figure 17: Read – Command Specific Status Values

Value	Description
B8h	Zone Boundary Error: The command specifies logical blocks in more than one zone.
BBh	Zone Is Offline: The accessed zone is in the ZSO:Offline state.

Source: ZNS Specification, at 20.

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.

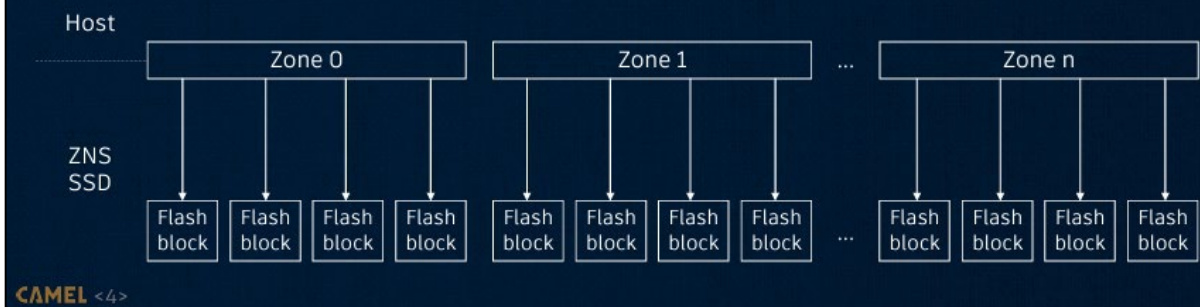
Figure 19: Write – Command Specific Status Values

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.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

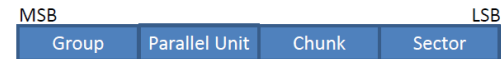


Source: Bae, at 4.

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

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

15

Source: Microsoft Denali Presentation, at 15.

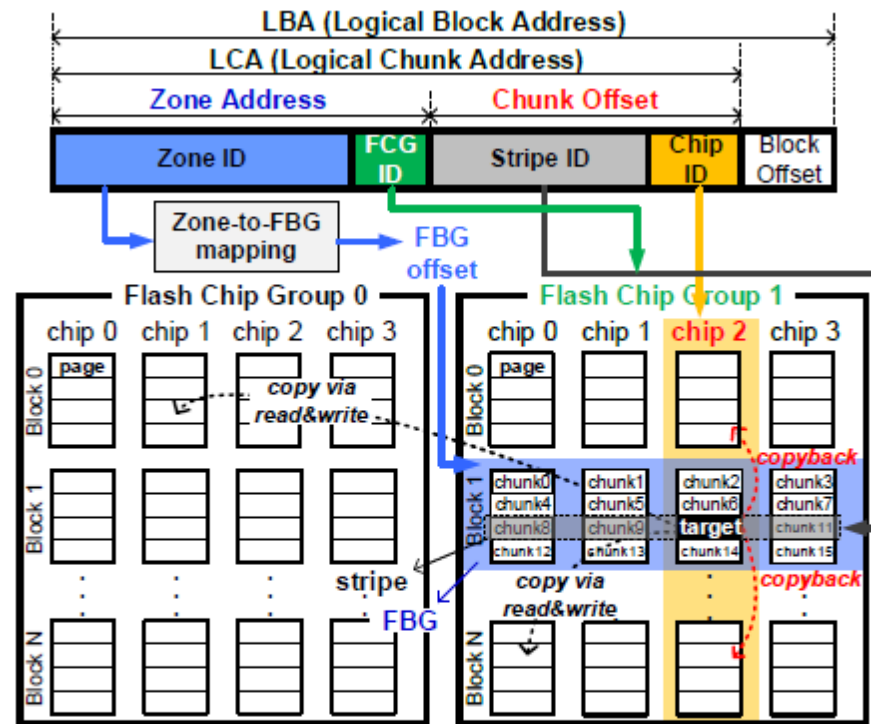


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

23d	identify an addressed segment within	Samsung's Accused Instrumentalities perform the step of identifying an addressed segment within the addressed block device from the second address portion.
-----	--------------------------------------	---

the addressed block device from the second address portion,

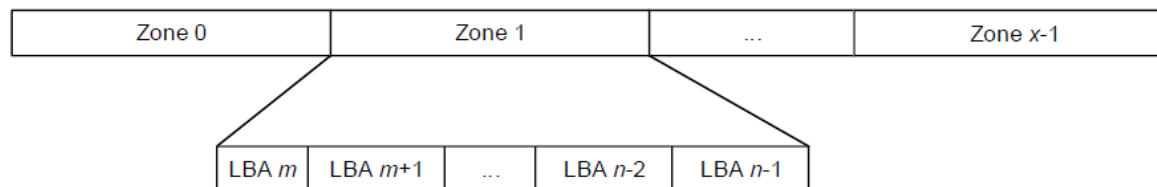
Samsung's Accused Instrumentalities are able to identify an addressed zone (segment) within the addressed namespace (block device) from the zone identifier information (second address portion). See:

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.

3.3.5 Read command

The Read 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).

3.3.5.1 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 17.

Figure 17: Read – Command Specific Status Values

Value	Description
B8h	Zone Boundary Error: The command specifies logical blocks in more than one zone.
BBh	Zone Is Offline: The accessed zone is in the ZSO:Offline state.

Source: ZNS Specification, at 20.

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.

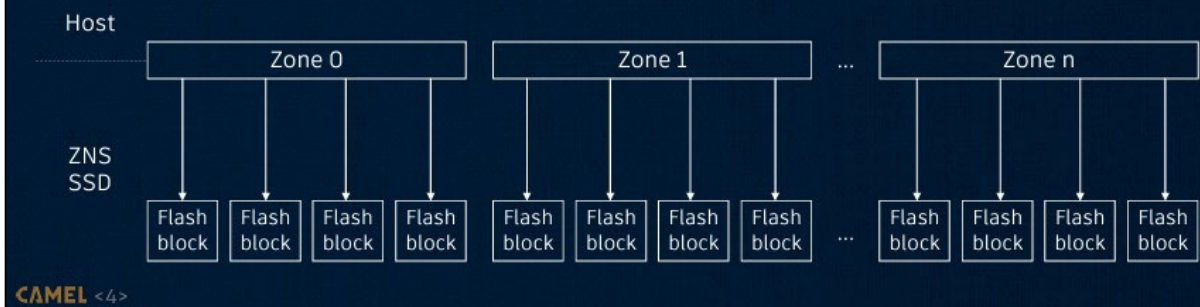
Figure 19: Write – Command Specific Status Values

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.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**

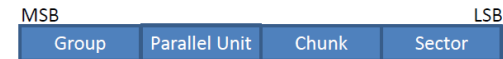


Source: Bae, at 4.

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

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

15

Source: Microsoft Denali Presentation, at 15.

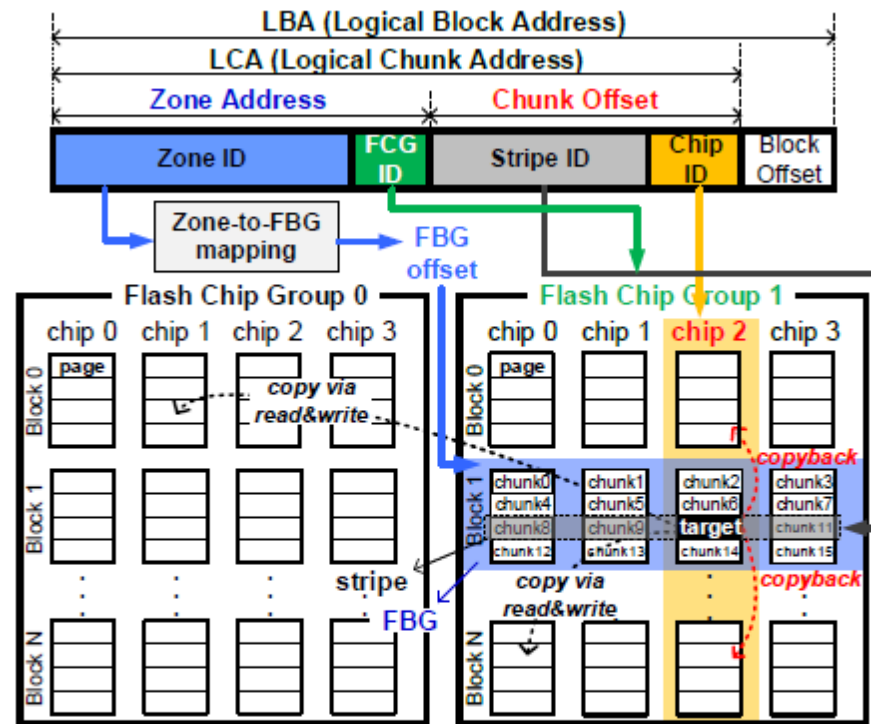


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

23e

identify an addressed storage

Samsung's Accused Instrumentalities perform the step of identifying an addressed storage location within the addressed segment from the third address portion.

location within the addressed segment from the third address portion,

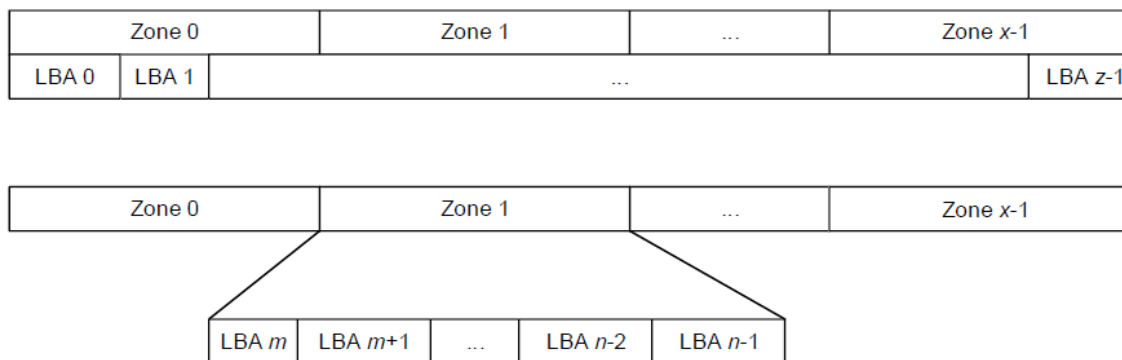
Samsung's Accused Instrumentalities are able to identify an addressed storage location, including a logical block, page, and/or erase unit, (addressed storage location) within the addressed zone (addressed segment) from the LBA and/or the zone offset (the third address portion). See:

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.

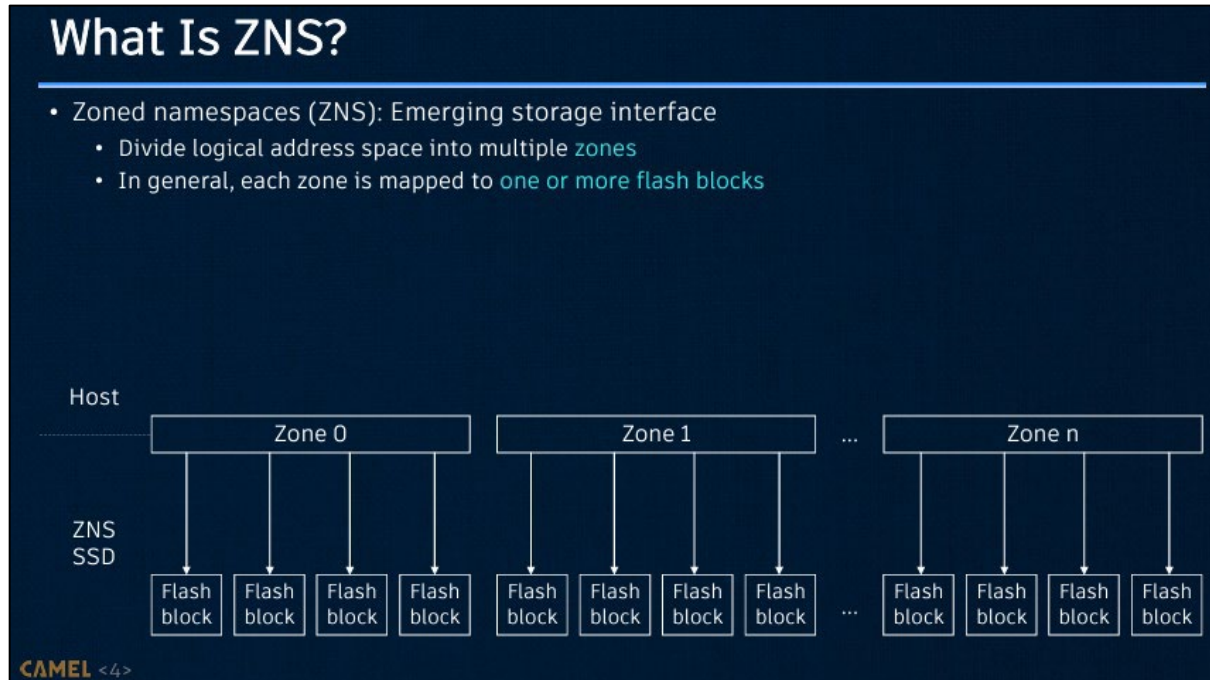
1.4.3.3 logical block

The smallest addressable data unit for Read and Write commands.

1.4.3.4 logical block address (LBA)

The address of a logical block, referred to commonly as LBA.

Source: NVM Command Set Specification, at 10.

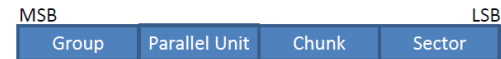


Source: Bae, at 4.

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

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

15

Source: Microsoft Denali Presentation, at 15.

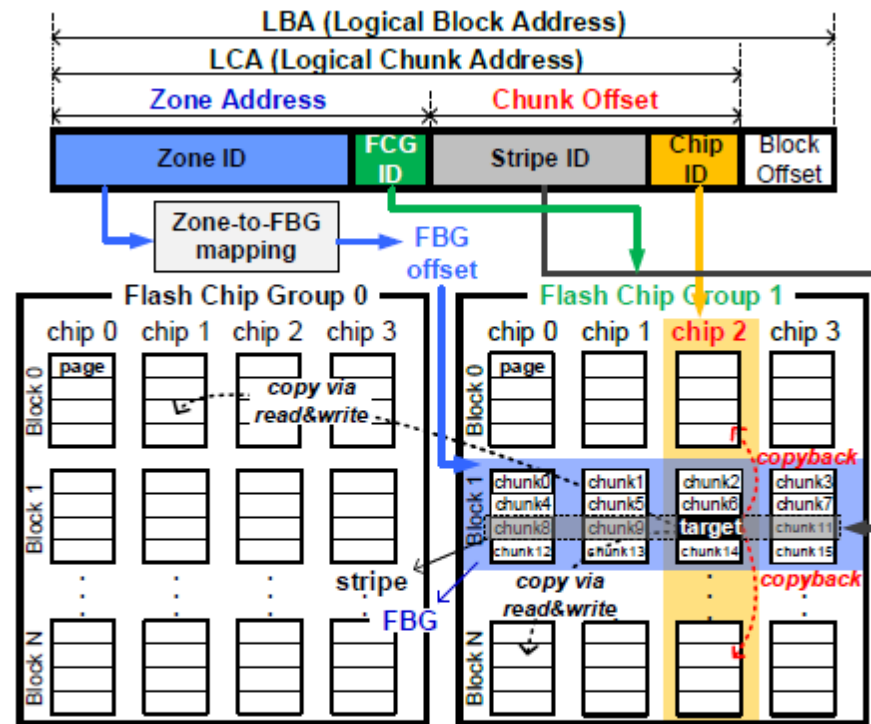


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

23f

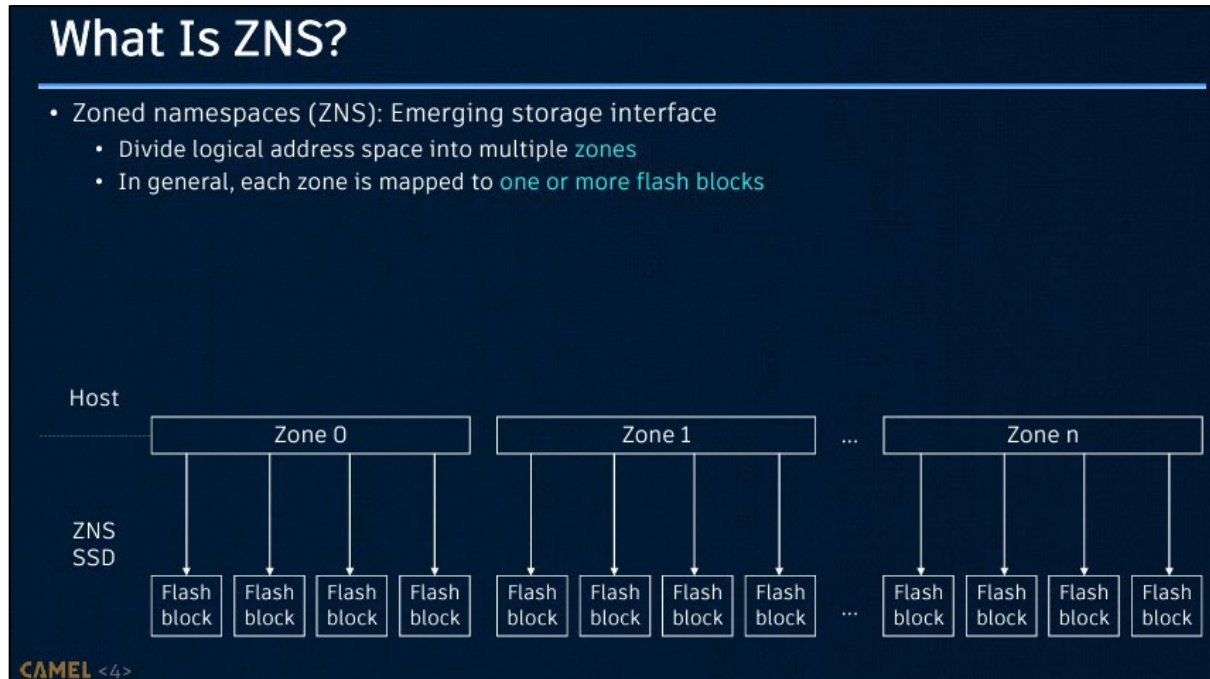
perform a
division
operation to

Samsung's Accused Instrumentalities perform the step of performing a division operation to identify a logical erase unit within the addressed segment.

identify a logical erase unit within the addressed segment, and

Samsung's Accused Instrumentalities are able to use a division operation on at least one of the incoming address information or information derived therefrom to identify a logical erase unit within the addressed zone. Samsung's Accused 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.

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:

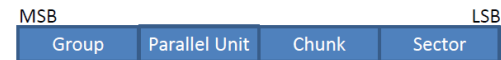


Source: Bae, at 4.

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

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

15

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.

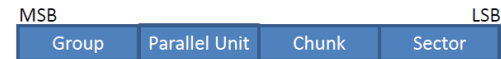
		<p>2. Why is OP important?</p> <p>OP has a direct effect on the SSD's random performance as the drive is filled with data. Guaranteeing free space to accomplish the NAND management tasks (GC, wear-leveling, bad block management) means the SSD does not have to waste time preparing space on demand, a process 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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

		<p>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.</p> <p>As an example, there is no substantial difference between (1) identifying a logical erase unit within the addressed segment and (2) identifying a logical erase unit group within the addressed segment 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.</p> <p>As another example, there is no substantial difference between (1) identifying a logical erase unit within the addressed segment and (2) having the addressed segment as a logical erase unit group and 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.</p>
23g	identify one of the physical erase units that corresponds to the identified logical erase unit;	<p>Samsung's Accused Instrumentalities perform the step of identifying one of the physical erase units that corresponds to the identified logical erase unit.</p> <p>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:</p>

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p> <p>Source: Samsung Application Note, at 2.</p>	
--	--	--	--

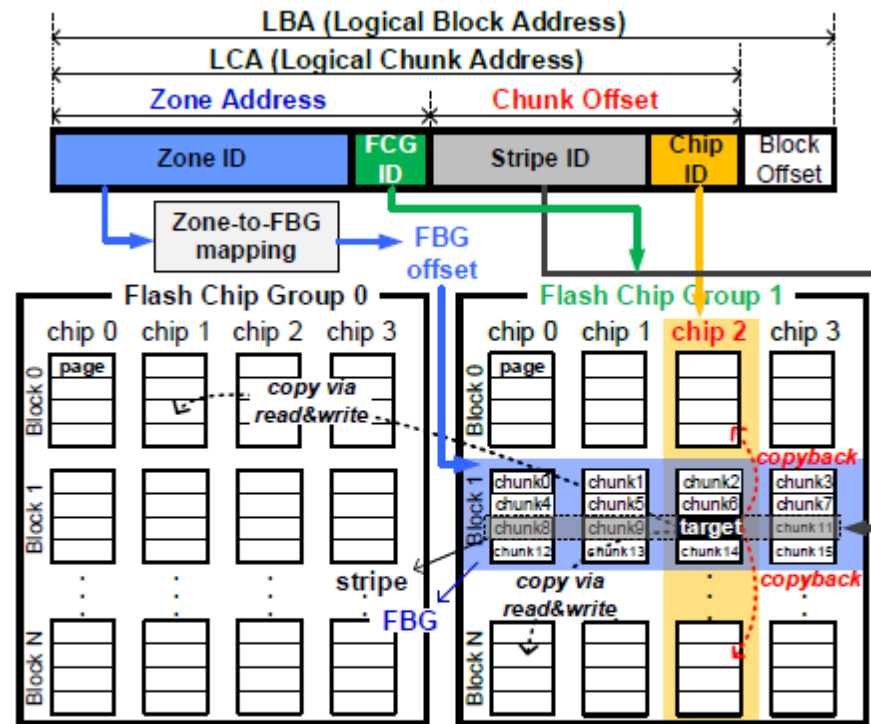
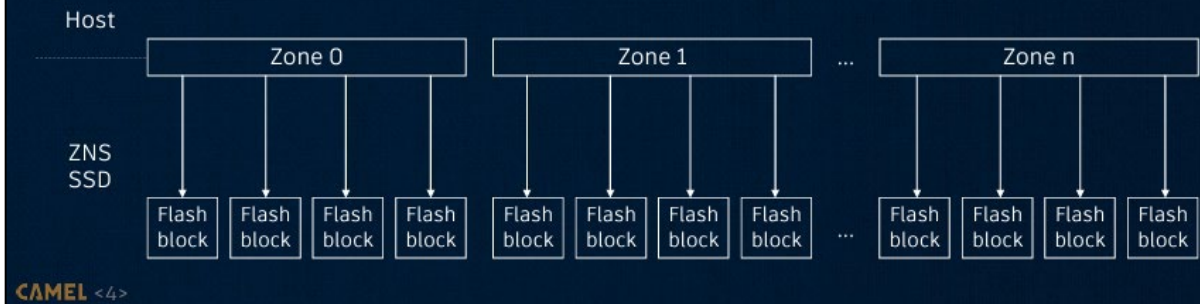


Figure 1: An example of zone and chunk mapping. With the zone address, the second FBG in the FCG 1 is selected. With the chunk offset, the third stripe in the selected FBG and the third flash page (chip 2) in the stripe are targeted.

Source: Han, at 149.

What Is ZNS?

- Zoned namespaces (ZNS): Emerging storage interface
 - Divide logical address space into multiple **zones**
 - In general, each zone is mapped to **one or more flash blocks**



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) identifying a physical erase unit that corresponds to the identified logical erase unit and (2) identifying a physical erase unit group that corresponds to the identified 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.

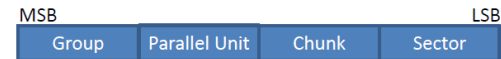
		<p>As another example, there is no substantial difference between (1) identifying a physical erase unit that corresponds to the identified logical erase unit and (2) having the addressed segment as a logical erase unit group which corresponds to a physical erase unit group and identifying an offset within 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.</p>
23h	<p>wherein the method further comprises causing the storage device to detect a failure condition of the physical erase units, 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</p>	<p>Samsung's Accused Instrumentalities cause the storage device to detect a failure condition of the physical erase units, 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 to utilize the remapped correspondence to service one or more of the incoming data access requests.</p> <p>Samsung's Accused Instrumentalities are able to detect a failure condition of physical blocks, retire the bad blocks, and substitute different blocks for the bad ones. Samsung's Accused Instrumentalities are able to remap each logical erase unit that had corresponded to one of the bad physical blocks (with the detected failure condition) to instead correspond to the replacement physical blocks for which there was not a failure condition and then utilize the remapped correspondence to service one or more of the incoming data access requests. See:</p>

detected and to utilize the remapped correspondence to service one or more of the incoming data access requests.

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

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

15

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.

		<p>2. Why is OP important?</p> <p>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.</p>	
		<p>Source: Samsung Application Note, at 2.</p>	