

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc.,
Petitioner,

v.

RADIAN MEMORY SYSTEMS, LLC,
Patent Owner.

Case IPR2025-01321

U.S. Patent No. 11,709,772

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. §§ 42.100 *et seq.***

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	STANDING	2
III.	TECHNICAL BACKGROUND	2
IV.	OVERVIEW OF THE '772 PATENT	3
	A. Priority	3
	B. Prosecution History	3
	C. Alleged Invention	3
V.	FOUNDATIONS	4
VI.	LEVEL OF ORDINARY SKILL	4
VII.	CLAIM CONSTRUCTION	5
VIII.	GROUND I: CLAIMS 1-6, 8, 11-13, 15-20, 22, 25-30 ARE RENDERED OBVIOUS BY OLBRICH IN LIGHT OF YAMADA, AND NVME1.0B	5
	A. Overview of the Prior Art.....	5
	1. Olbrich (EX1004)	5
	2. Yamada (EX1005)	6
	3. NVMe1.0b (EX1006)	7
	B. Motivation to Combine	7
	C. Claim-by-Claim Analysis.....	9
	1. Claim 1[pre]: A method for providing a host with access to an SSD that contains logic and non-volatile memory comprising NAND flash memory, the logic comprising circuitry and/or a processor executing instructions, and the NAND flash memory comprising a plurality of erase units and each erase unit comprising a plurality of pages, the method comprising, at the SSD:	9
	(a) Claim 1[a]: dividing the flash memory into zones; wherein the zones are non-overlapping subdivisions of the flash memory, each zone comprises two or more erase units, the erase units in each zone are on two or more different planes,.....	10

(b)	Claim 1[b]: and each zone is associated with a zone start logical block address and a range of logical block addresses;	11
(c)	Claim 1[c]: in response to a write command from the host to write data to the range of logical block addresses of a specified zone, performing a multi-plane write of said data to erase units on different planes in the specified zone using a common page address for the respective erase units on the different planes;	13
(d)	Claim 1[d]: generating metadata based on accessing and maintenance of erase units in the specified zone; and;	13
(e)	Claim 1[e]: based on the metadata, making a notification available to the host that recommends performing maintenance on the specified zone, wherein the recommendation is associated with the zone start logical block address of the specified zone;.....	14
(f)	Claim 1[f]: wherein the SSD and the host communicate over a host interface, using commands from a Non-Volatile Memory Express (NVMe) standard.....	16
2.	Claim 2: A solid state storage drive (SSD) comprising:.....	16
(a)	2[a]: a host interface configured to communicate with a host using commands from a Non-Volatile Memory Express (NVMe) standard;	17
(b)	2[b]: non-volatile memory comprising NAND flash memory, the NAND flash memory comprising a plurality of erase units and each erase unit comprising a plurality of pages;	17
(c)	2[c]: logic coupled to the host interface and to the flash memory, wherein the logic comprises at least one of hardware logic and software logic, and communicates with the host via the host interface using NVMe commands;	17

(d)	2[d]: wherein the logic operates to: divide the flash memory into zones; wherein the zones are non-overlapping subdivisions of the flash memory, each zone comprises two or more erase units, the erase units in at least one zone are on two or more different planes, ...	17
(e)	2[e]: and each zone is associated with a zone start logical block address and a range of logical block addresses;	17
(f)	2[f]: in response to a write command from the host to write data within the range of logical block addresses of a specified zone, perform a multi-plane write of said data to erase units on different planes in the specified zone using a common page address for the respective erase units on the different planes;	18
(g)	2[g]: generate metadata based on accessing and maintenance of erase units in the specified zone; and;	18
(h)	2[h]: based on the metadata, make a notification available to the host that recommends performing maintenance on the specified zone, wherein the recommendation is associated with the zone start logical block address of the specified zone.	18
3.	Claim 3: The SSD of claim 2 further comprising: additional non-volatile memory, wherein the logic further operates to store the generated metadata in the additional non-volatile memory.	18
4.	Claim 4: The SSD of claim 2 wherein the logic further operates to compare the metadata with at least one threshold, and wherein making the notification available to the host is dependent on the comparison.	19
5.	Claim 5: The SSD of claim 2	20
(a)	wherein the logic further operates to determine, based on the metadata, that maintenance is recommended for the specified zone and to identify the zone start logical block address	

	associated with the specified zone for which maintenance is recommended;.....	20
(b)	wherein determining that maintenance is recommended for the specified zone, identifying the zone start logical block address associated with the specified zone, and making the notification available to the host are all unsolicited by the host.	21
6.	Claim 6: The SSD of claim 2 wherein the logic further operates to receive a command from the host; and, in response to said command, transmit to the host (a) information representing list of host accessible zones including a zone start logical block address for each listed zone, and (b) a host accessible capacity for each listed zone.	22
7.	Claim 8: The SSD of claim 2 wherein:.....	24
(a)	the logic further operates to receive read commands and write commands from the host, and to execute the read commands and the write commands by transferring associated data between the host and the NAND flash memory;	24
(b)	and each read command directed to the SSD is accompanied by a logical read address, each logical read address has a first address portion which identifies one of the zones according to a value of the first portion, and a second address portion which identifies a physical memory location in the SSD in a manner that depends on the first address portion.	24
8.	Claim 11: The SSD of claim 2 wherein the flash memory comprises multiple dies of flash memory and the erase units on different planes are also in different dies., ...	25
9.	Claim 12: The SSD of claim 2 wherein the logic further operates to, upon the performance of the multi-plane write, update the metadata to indicate an amount of unerasd host accessible space in the specified zone.	26
10.	Claim 13: The SSD of claim 12 wherein the logic further operates to: upon the performance of the multi-plane	

	write, detect that the specified zone contains no erased host accessible space; and, in response to said detection, transition the specified zone to a full state.....	26
11.	Claim 15: The SSD of claim 2 wherein the logic further operates to receive a maintenance command directed to the specified zone and, in response to the maintenance command, erase host accessible space only within the specified zone and transition the specified zone to an empty state.	27
12.	Claim 16: The SSD of claim 2 wherein the metadata is dependent on an erase count for the specified zone; and the logic further operates to compare the metadata with at least one threshold for erase count, wherein making the notification available to the host is dependent on the comparison.	28
13.	Claim 17: The SSD of claim 2 wherein the logic further operates to: in response to a maintenance command from the host directed to the zone start logical block address of the specified zone, perform an erase of erase units in the specified zone, and transition a state of the specified zone to an empty state whereby all host accessible pages of the specified zone are available for writing by the host.	29
14.	Claim 18: The SSD of claim 17 wherein the maintenance command from the host is in response to the notification that recommends performing maintenance on the specified zone.....	31
15.	Claim 19: The SSD of claim 17 wherein the metadata is dependent on erase counts for erase units in the specified zone; and the logic further operates to update the metadata responsive to performance of the erase.	32
16.	Claim 20: The SSD of claim 17 wherein performing the erase of erase units in the specified zone comprises performing the erase on erase units of different planes in the specified zone.....	32
17.	Claim 22: The SSD of claim 2.....	33

	(a)	wherein the logic further operates to, in response to a write command from the host to the zone start logical block address of the specified zone:	33
	(b)	assign a logical block address within the range of logical block addresses of the specified zone, transfer data from the host to the specified zone at the assigned logical block address,.....	33
	(c)	and return the assigned logical block address to the host.....	34
18.		Claim 25: The SSD of claim 2.....	35
	(a)	wherein the metadata is dependent on a time period since data was written to the specified zone;	35
	(b)	and the logic further operates to compare the metadata with at least one threshold, wherein making the notification available to the host is dependent on the comparison	35
19.		Claim 26: The SSD of claim 25 wherein comparing the metadata with at least one threshold and making the notification available to the host are unsolicited by the host	36
20.		Claim 27: The SSD of claim 2 wherein logic further operates to, in response to a command from the host, transmit to the host information dependent on a time period since data was written to the specified zone.....	36
21.		Claim 28: The SSD of claim 2 wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and transmit to the host the zone start logical block address and a state of the specified zone.....	37
22.		Claim 29: The SSD of claim 2 wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and, in response to a write command from the host within the range of logical block addresses of the specified zone, transmit to the host a notification of a write failure.	39

23.	Claim 30: The SSD of claim 2 wherein the metadata is dependent on read errors detected for the specified zone.....	40
IX.	GROUND II: CLAIMS 1-6 ARE RENDERED OBVIOUS BY GONZALEZ IN LIGHT OF GOSS AND NVME1.0B	41
A.	Overview of the Prior Art.....	41
1.	Gonzalez (EX1007).....	41
2.	Goss (EX1008).....	41
B.	Motivation to Combine	41
C.	Claim-by-Claim Analysis.....	43
1.	Claim 1[pre]	43
(a)	Claim 1[a]	43
(b)	Claim 1[b].....	43
(c)	Claim 1[c]	44
(d)	Claim 1[d].....	44
(e)	Claim 1[e]	45
(f)	Claim 1[f].....	47
2.	Claim 2	47
(a)	2[a]	47
(b)	2[b].....	47
(c)	2[c]	48
(d)	2[d].....	48
(e)	2[e].....	48
(f)	2[f]	48
(g)	2[g].....	48
(h)	2[h].....	48
3.	Claim 3.....	48
4.	Claim 4.....	49
5.	Claim 5.....	50
(a)	5[a].....	50
(b)	5[b].....	50

6.	Claim 6.....	51
X.	GROUND III – CLAIM 8 IS RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, NVME1.0B, AND BAN	51
A.	References	51
1.	Ban (EX1010)	51
B.	Motivation to Combine	52
C.	Claim-by-claim Analysis.....	52
1.	Claim 8.....	52
(a)	8[a].....	53
(b)	8[b].....	53
XI.	GROUND IV - CLAIM 11 IS RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, NVME1.0B, AND SPROUSE	55
A.	References	55
1.	Sprouse (EX1011).....	55
B.	Motivation to Combine	55
C.	Claim-by-claim Analysis.....	56
1.	Claim 11	56
XII.	GROUND V – CLAIMS 12, 13, 15-20, 22, 25-27 ARE RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, NVME1.0B, AND SON	57
A.	References	57
1.	SON (EX1009)	57
B.	Motivation to Combine	57
C.	Claim-by-claim Analysis.....	59
1.	Claim 12	59
2.	Claim 13.....	60
3.	Claim 15.....	60
4.	Claim 16.....	61
5.	Claim 17.....	62
6.	Claim 18.....	63

7.	Claim 19.....	64
8.	Claim 20.....	65
9.	Claim 22.....	65
	(a) 22[a].....	65
	(b) 22[b].....	65
	(c) 22[c].....	66
10.	Claim 25.....	67
	(a) 25[a].....	67
	(b) 25[b].....	67
11.	Claim 26.....	68
12.	Claim 27.....	68
XIII.	GROUND VI - CLAIMS 28-30 ARE RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, SON, NVME1.0B, AND CORNWELL	69
A.	References	69
	1. Cornwell (EX1012).....	69
B.	Motivation to Combine	69
C.	Claim-by-claim Analysis.....	70
	1. Claim 28.....	70
	2. Claim 29.....	71
	3. Claim 30.....	72
XIV.	MANDATORY NOTICES AND FEES	73
A.	Real Party-In-Interest	73
B.	Related Matters.....	73
C.	Counsel and Service Information.....	73
D.	Payment of Fees	74
XV.	CONCLUSION.....	74

EXHIBIT LIST

Ex.	Description
1001	U.S. Patent No. 11,709,772 (“’772 Patent”)
1002	Prosecution History of U.S. Patent No. 11,709,772
1003	Expert Declaration of Dr. Heiner Litz
1004	U.S. Patent No. 8,386,700 (“Olbrich”)
1005	U.S. Pat. Pub. No. 2011/0271032 (“Yamada”)
1006	NVMe Base Specification Rev. 1.0b (“NVMe1.0b”)
1007	U.S. Pat. Pub. No. 2010/0049908 (“Gonzalez”)
1008	U.S. Pat. Pub. No. 2011/0302477 (“Goss”)
1009	U.S. Pat. Pub. No. 2012/0096217 (“Son”)
1010	U.S. Patent No. 5,404,485 (“Ban”)
1011	U.S. Patent Pub. No. 2011/0153911 (“Sprouse”)
1012	U.S. Patent Pub. No. 2012/0072807 (“Cornwell”)

I. INTRODUCTION

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc., (collectively, “Samsung” or “Petitioner”) respectfully requests *inter partes* review of claims 1-6, 8, 11-13, 15-20, 22, and 25-30 of U.S. Patent No. 11,709,772 (the “’772 Patent”). A reasonable likelihood exists that Petitioner will prevail on at least one challenged claim.

The ’772 Patent purports to describe “improvements in managing multi-drive, multi-die or multi-plane NAND flash memory,” based on sharing information and responsibility between the controller and host, termed “cooperative or shared management.” EX1001, Abstract, 18:26-45. The ’772 Patent relates to a flash memory controller where memory is organized into units termed subdivisions or zones, where metadata is provided to the host for use in memory management.

The ’772 Patent represents common techniques applied to flash memory management in a system with distributed functions between the host and controller. The claimed novelty, shared management responsibility for memory management, was known in the art prior to the effective filing date of the ’772 Patent. As one example, Yamada teaches cooperative interaction between a controller and a host, where metadata is sent to the host for memory management. EX1005, [0014], [0109]-[0111].

The '772 Patent lacks any inventive concept and for the reasons herein is rendered obvious by Grounds I-VI.

II. STANDING

Petitioner certifies the '772 Patent is available for *inter partes* review and is not estopped under 35 U.S.C. § 325.

III. TECHNICAL BACKGROUND

Flash memory is organized into discrete units called erase units (also referred to as erase blocks or blocks). EX1001, 1:35-56. To increase the efficiency, SSDs may employ a larger unit of memory—organizing blocks across planes that are operated as a single unit, *e.g.*, “SuperBlocks,” EX1004, 16:12-16, 19:11-15; EX1003, ¶29.

To increase the longevity of flash memory, SSDs perform management operations including wear-leveling and other maintenance where data may be copied, moved or erased to prolong data storage life. EX1004, Abstract; EX1005, [0007]; EX1003, ¶¶34-37. To identify data for maintenance, metadata is tracked describing memory history. EX1004, 48:5-7; EX1005, [0081], [0208]-[0209].

Memory management functions may be carried out by the controller and/or the host. EX1005, [0014], [0109]-[0111]; EX1003, ¶¶40-41.

IV. OVERVIEW OF THE '772 PATENT

A. Priority

The '772 patent issued on January 3, 2023, and claims priority to Provisional Application No. 61/757,464, filed on January 28, 2013.

The prior art references relied upon in this Petition predate January 28, 2013, the earliest possible critical date.

B. Prosecution History

The nonprovisional application that became the '772 Patent was filed on July 29, 2022. *See* EX1002. On October 7, 2022, the examiner issued a non-final rejection on grounds other than 35 U.S.C. 102 and 103. On February 8, 2023, the examiner issued a Notice of Allowance.

C. Alleged Invention

The '772 Patent describes a memory system where responsibility for memory management is shared between the controller and host. EX1001, Abstract, 18:26-45.

The claims are directed to memory units termed, “zones” which are “two or more erase units ...on two or more different planes.” EX1001, Claim 1. The specification does not use the term “zones,” but describes “subdivisions,” where “each subdivision can include one or more erase blocks or units (EUs), with each EU having a minimum number of memory cells that must be erased at once.” *Id.*, 13:22-26.

Memory management functions, *e.g.*, “host-initiated maintenance operations,” are based on metadata sent “to the host” such as “age” or “erase count,” *id.*, 24:36-25:64, 53:15-36.

V. GROUNDS

Ground	Basis	Reference(s)	Claims
I	§ 103	Olbrich, Yamada, NVMe1.0b	1-6, 8, 11-13, 15-20, 22, 25-30
II	§ 103	Gonzalez, Goss, NVMe1.0b	1-6
III	§ 103	Gonzalez, Goss, Ban, NVMe1.0b	8
IV	§ 103	Gonzalez, Goss, Sprouse, NVMe1.0b	11
V	§ 103	Gonzalez, Goss, Son, NVMe1.0b	12-13, 15-20, 22, 25-27
VI	§ 103	Gonzalez, Goss, Son, Cornwell, NVMe1.0b	28-30

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSA”) at the time of the ’772 Patent would have had at least a Bachelor of Science degree in electrical engineering, or an equivalent field, and at least three years of experience with SSDs or other flash memory devices. EX1003, ¶47. More education can supplement practical experience and vice versa. *Id.* Petitioner’s expert, Dr. Litz, exceeds this skill level and has thus the qualification and experience necessary to assess the ’772 Patent based on the perspective of a POSA. *Id.*, ¶¶4-11, Ex. 1.

VII. CLAIM CONSTRUCTION

Petitioner does not believe any terms require construction at this stage for the Board to find all challenged claims unpatentable and should thus be given their plain and ordinary meaning.

VIII. GROUND I: CLAIMS 1-6, 8, 11-13, 15-20, 22, 25-30 ARE RENDERED OBVIOUS BY OLBRICH IN LIGHT OF YAMADA, AND NVME1.0B

A. Overview of the Prior Art

1. Olbrich (EX1004)

U.S. Patent 8,386,700 (“Olbrich”) was filed on November 29, 2011, was published on March 22, 2012, and issued on February 26, 2013. It is prior art under AIA 35 U.S.C. § 102.

Olbrich discloses a storage device with SuperBlocks where metadata for the SuperBlocks is tracked in a “super block metadata table,” as illustrated below:

3502		3505 SUPER PAGE TIME STAMP	3510 DEFECT	3511 LBA
3503 BANK 0	SUPER PAGE 0			
	SUPER PAGE 1			
	⋮			
	SUPER PAGE 63			
3504 BANK 1	SUPER PAGE 0			
	SUPER PAGE 1			
	⋮			
	SUPER PAGE 63			
⋮				
BANK n	SUPER PAGE 0			
	SUPER PAGE 1			
	⋮			
	SUPER PAGE 63			
3506 — SUPER BLOCK TIME STAMP _____				
3507 — ERASE COUNT _____				
3508 — OPEN FLAG _____				
3509 — CLOSED FLAG _____				

SUPER BLOCK METADATA TABLE 3501

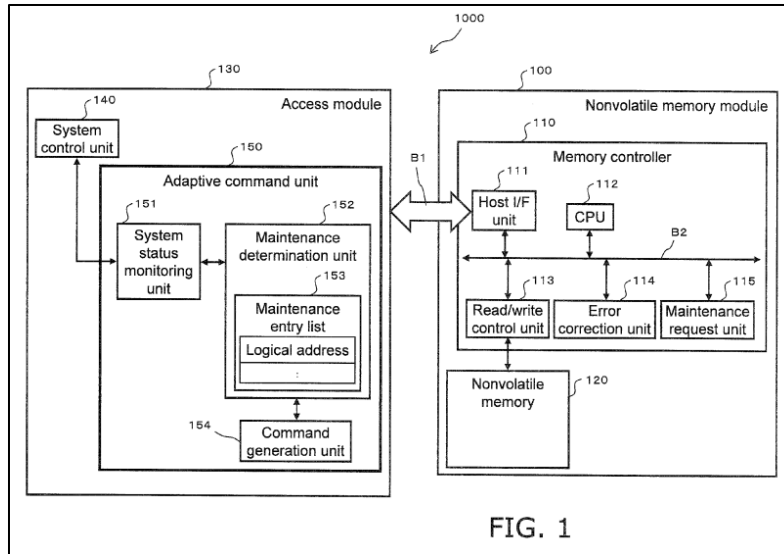
EX1004, Fig. 35, 48:5-44; EX1003, ¶¶51-52.

2. Yamada (EX1005)

U.S. Pat. Pub. No. 2011/0271032A1 (“Yamada”) was filed on July 27, 2010, and was published November 3, 2011. It is prior art under AIA 35 U.S.C. § 102.

Yamada teaches an SSD including a controller with a “maintenance request unit,” and the access device (host) with a “maintenance determination unit.”

EX1006, [0012]; EX1003, ¶55.



The controller identifies an LBA(s) for maintenance metadata threshold-analysis. EX1006, [0013]-[0014], [0032], [0080]-[0081], [0109]-[0111]; EX1003, ¶56.

3. NVMe1.0b (EX1006)

The NVMe Express Base Specification Revision 1.0b (“NVMe1.0b”) was published on July 12, 2011. It is prior art under AIA 35 U.S.C. § 102.

The NVMe1.0b represents a widely adopted standard promulgated by the industry to promote interoperability and compatibility across systems. EX1003, ¶58. The NVMe includes a unit of memory termed a namespace, a grouping of logical blocks. *Id.*, ¶ 59.

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Yamada and NVMe1.0b with Olbrich. Specifically, a POSA would have been motivated to (i) incorporate Yamada’s teaching regarding controller maintenance requests and host-

initiated maintenance commands and (ii) conform to the NVMe1.0b standard. EX1003, ¶¶75-83.

First, Olbrich and Yamada involve similar techniques to solve the same problem, *i.e.*, collecting, storing and using metadata for maintenance activities to improve the efficiency and performance of NAND flash memory systems. *E.g.*, EX1004, 1:45-3:17, Fig. 1; EX1006, [0005]-[0011], [0029], [0207]-[0211]; EX1003, ¶78.

Second, Olbrich teaches various means to “maximize[] use of the host and flash interfaces” through the “use of multiple microprocessors in the controller design, thereby creating multiple parallel independent pipelines.” EX1004, 2:64-3:1. Due to this architecture, the system of Olbrich is capable of ready modification to include Yamada’s discrete maintenance modules. A POSA would have a reasonable expectation of success in incorporating this shared host functionality—adding a maintenance request module to the controller and translocating the maintenance command responsibility to host-side software—and it would have been straightforward and routine. *Id.*; EX1003, ¶79.

A POSA would be motivated to incorporate Yamada’s teaching into the system in Olbrich due to the benefits taught regarding host-initiated maintenance. EX1003, ¶80. Yamada describes a litany of advantages regarding data retention,

longevity and performance of flash memory based on shared control with the host with respect to maintenance activities. *See, e.g.*, EX1005, [0013], [0036].

Thus, a POSA would be motivated to incorporate Yamada’s teaching into the system in Olbrich regarding host-initiated maintenance commands, along with the metadata variants triggering maintenance. EX1003, ¶81.

Third, Olbrich teaches that the claimed systems and methods are flexible and are “designed to operate with various enterprise-level mass storage protocols” and “may be adapted for use with other similar protocols, either currently in use or not yet developed.” EX1004, 4:35-55. A POSA would therefore be motivated to incorporate the teachings of a then-operative NVMe express standard and could do so with a reasonable expectation of success. EX1003, ¶82.

Petitioners are unaware of any secondary considerations of non-obviousness.

C. Claim-by-Claim Analysis

- 1. Claim 1[pre]: A method for providing a host with access to an SSD that contains logic and non-volatile memory comprising NAND flash memory, the logic comprising circuitry and/or a processor executing instructions, and the NAND flash memory comprising a plurality of erase units and each erase unit comprising a plurality of pages, the method comprising, at the SSD:**

To the extent the preamble is limiting, it is disclosed by Olbrich. EX1003, ¶¶84-85.

Olbrich teaches a “controller” that provides a host access to an SSD through a “processor” [*a method for providing a host with access to an SSD that contains logic and non-volatile memory ...the logic comprising circuitry and/or a processor executing instructions*], EX1004, 1:45-52, 2:59-3:21, Fig. 1, Claim 1, where the SSD comprises “solid state nonvolatile memory,” e.g., a “NAND flash interface” made up of erase “blocks, each made up of 64 pages,” [*comprising NAND flash memory ... and the NAND flash memory comprising a plurality of erase units and each erase unit comprising a plurality of pages*], *id.*, 9:43-48, 11:18-28.

- (a) **Claim 1[a]: dividing the flash memory into zones; wherein the zones are non-overlapping subdivisions of the flash memory, each zone comprises two or more erase units, the erase units in each zone are on two or more different planes,**

Olbrich teaches and/or renders this limitation obvious. EX1003, ¶¶86-88.

Olbrich teaches *dividing the flash memory into “SuperBlocks,”* where each SuperBlock includes 2+ blocks from multiple different memory banks [*dividing the flash memory into zones; wherein the zones are non-overlapping subdivisions of the flash memory, each zone comprises two or more erase units, the erase units in each zone are on two or more different planes*]. EX1004, 16:12-16. A POSA would understand that blocks on different banks and dies represent non-overlapping blocks on different planes. EX1003, ¶87.

BANK	FLASH GROUP 0		FLASH GROUP 1		FLASH GROUP 2	
	SPO	SP1	SPO	SP1	SPO	SP1
0	SMT	285-299	SMT	375-389	SMT	465-479
1	0-14	300-314	90-104	390-404	180-194	480-494
2	15-29	315-329	105-119	405-419	195-209	495-509
3	30-44	330-344	120-134	420-434	210-224	510-524
4	45-59	345-359	135-149	435-449	225-239	525-539
5	60-74	540-554	150-164	585-599	240-254	630-644
6	75-89	555-569	165-179	600-614	255-269	645-659
7	270-284	570-584	360-374	615-629	450-464	660-674

1601

1602

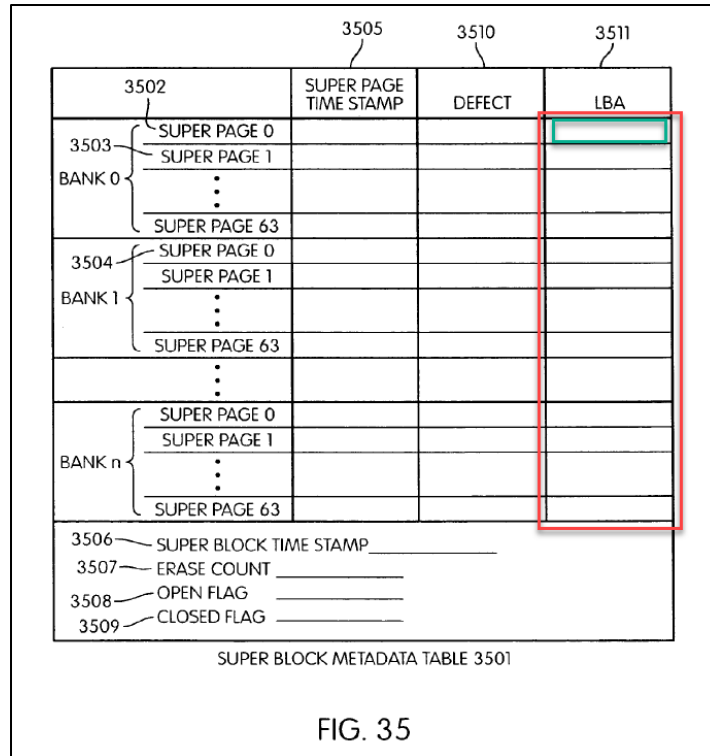
FIG. 16

EX1004, Fig. 16. A SuperBlock includes SuperPages where each “consists of four flash Pages, each of which is stored on a different Flash Die” and thus a different plane, *id.*, 6:23-29.

(b) Claim 1[b]: and each zone is associated with a zone start logical block address and a range of logical block addresses;

Olbrich teaches this limitation. EX1003, ¶¶89-91.

Olbrich teaches that each SuperBlock is comprised of a range of LBAs (Fig. 35, annotated red), and a starting LBA (annotated green) [*and each zone is associated with a zone start logical block address and a range of logical block addresses*].



EX1004, Fig. 35.

A POSA would understand that a SuperBlock comprised of multiple SuperPages includes a ZSLBA, the starting address of the first SuperPage, and a range of LBAs (sum of consecutive LBAs for each SuperPage) ending with the “highest LBA.” *Id.*, 15:37-49 (“SSD Controller 106 responds to ***an LBA range that begins with LBA 0*** and ends with the highest LBA addressable by the Controller.”); EX1003, ¶91. *See also* EX1004, 15:37-49, 72:33-38 (“all four Dies receive the same address”).

- (c) **Claim 1[c]: in response to a write command from the host to write data to the range of logical block addresses of a specified zone, performing a multi-plane write of said data to erase units on different planes in the specified zone using a common page address for the respective erase units on the different planes;**

Olbrich teaches and/or renders this limitation obvious. EX1003, ¶¶92-95.

Olbrich teaches that “SSD Controller 106 [may] receive a write command for a particular block of LBAs,” or a single LBA that is directed to a SuperBlock [*in response to a write command from the host to write data to the range of logical block addresses of a specified zone*]. EX1004, 15:27-34, 25:21-41, 41:23-25.

Olbrich teaches that the write command may designate a multi-plane write across pages in blocks on different dies—a POSA would understand that a die is composed of planes, meaning a multi-die write necessarily is a multi-plane write—through the use of “the same address location” [*using a common page address for the respective erase units on the different planes*]. *Id.* 15:60-67; EX1003, ¶94.

- (d) **Claim 1[d]: generating metadata based on accessing and maintenance of erase units in the specified zone; and;**

Olbrich teaches this limitation. EX1003, ¶¶96-98.

Olbrich teaches the generation of “Time Stamp” SuperBlock metadata “indicating the time when each SuperBlock was originally closed for writing.” EX1004, 47:45-49. Olbrich further teaches generating metadata showing “the

number of times the associated LBAs ***have been accessed*** (read or written).” *Id.*, 46:19-29. A POSA would understand that this metadata describes access to SuperBlocks and all constituent blocks and pages [***generating metadata based on accessing ... erase units in the specified zone***]. EX1003, ¶96.

Olbrich also teaches generating “Erase Count” SuperBlock metadata, representing “the number of times the SuperBlock has been erased.” EX1004, 48:39-41. Olbrich states this metadata is used “for wear leveling purposes” [***metadata based on ... maintenance of erase units in the specified zone***]. *Id.*

- (e) **Claim 1[e]: based on the metadata, making a notification available to the host that recommends performing maintenance on the specified zone, wherein the recommendation is associated with the zone start logical block address of the specified zone;**

The combined system of Olbrich and Yamada renders this limitation obvious. EX1003, ¶¶99-104.

Olbrich teaches the initiation of maintenance—the plain of ordinary meaning of which includes operations “such as wear leveling and garbage collection,” EX1001, 9:48-52—based on metadata. For example, Olbrich teaches that a SuperBlock may be selected for garbage collection based on SuperBlock erase count metadata. EX1004, Claims 1, 24, 92:18-22. Thus, Olbrich teaches SuperBlock maintenance [***performing maintenance on the specified zone***], depending on SuperBlock erase count metadata [***based on the metadata***].

As an additional disclosure teaching this limitation, Olbrich teaches that based on capacity metadata, maintenance may be initiated—where a “Counter value” (SuperBlock capacity) “is at or below the Critical Threshold, a Critical Flag is set and the [device] shifts into critical garbage collection mode.” EX1004, 80:16-24, 81:22-38.

To the extent Olbrich does not teach this limitation, a POSA would be motivated to incorporate Yamada’s teaching into Olbrich’s system—given that the “SSD Controller 106 could use explicit wear leveling techniques.” *Id.*, 91:64-65; EX1003, ¶¶101-103. Yamada teaches that the controller’s “maintenance request unit determines whether an area (in units of blocks) in the nonvolatile memory from which data has been read needs maintenance based on [metadata, *e.g.*,] the number of errors determined by the error correction unit” [***based on the metadata***]. EX1005, [0081], [0026]. Upon “determining that the area needs maintenance, the maintenance request unit transmits a maintenance request to the [host]” [***making a notification available to the host***]. *Id.*, [0081]-[0082]. The maintenance request may be directed “to a plurality of addresses of a plurality of blocks” identified by, *e.g.*, the “logical address of the start of the maintenance-needed block” [***that recommends performing maintenance on the specified zone***]. *Id.*, [0101], [0116]-[0118].

This is a clear teaching that, when incorporating Yamada’s teaching into the system of Olbrich, a POSA would understand that the maintenance request as

applied to a SuperBlock would include a starting LBA or “logical address of the start of the maintenance-needed block” in the corresponding SuperBlock [*is associated with the zone start logical block address of the specified zone*]. EX1003, ¶103.

(f) Claim 1[f]: wherein the SSD and the host communicate over a host interface, using commands from a Non-Volatile Memory Express (NVMe) standard

The combined system of Olbrich, Yamada and NVMe1.0b renders this limitation obvious. EX1003, ¶¶104-105.

Olbrich teaches that its systems are “designed to operate with various enterprise-level mass storage protocols” and “*may be adapted for use with other similar protocols, either currently in use or not yet developed.*” EX1004, 4:35-55. A POSA would therefore be motivated to incorporate an NVMe standard, such as NVMe1.0b, to facilitate host commands and communications via the host interface, EX1004, 7:26-47; EX1003, ¶105, [*wherein the SSD and the host communicate over a host interface, using commands from a Non-Volatile Memory Express (NVMe) standard*].

2. Claim 2: A solid state storage drive (SSD) comprising:

As described in Claim 1[pre], Olbrich teaches an SSD. EX1003, ¶106.

- (a) 2[a]: a host interface configured to communicate with a host using commands from a Non-Volatile Memory Express (NVMe) standard;**

As described in Claim 1[f], the combined system of Olbrich, Yamada and NVMe1.0b renders this limitation obvious. EX1003, ¶107.

- (b) 2[b]: non-volatile memory comprising NAND flash memory, the NAND flash memory comprising a plurality of erase units and each erase unit comprising a plurality of pages;**

As described in Claim 1[pre], Olbrich teaches this limitation. EX1003, ¶108.

- (c) 2[c]: logic coupled to the host interface and to the flash memory, wherein the logic comprises at least one of hardware logic and software logic, and communicates with the host via the host interface using NVMe commands;**

As described in Claim 1[pre], 1[f], the combined system of Olbrich, Yamada and NVMe1.0b renders this limitation obvious. EX1003, ¶¶109-110. *See also* EX1004, 33:47-61.

- (d) 2[d]: wherein the logic operates to: divide the flash memory into zones; wherein the zones are non-overlapping subdivisions of the flash memory, each zone comprises two or more erase units, the erase units in at least one zone are on two or more different planes,**

As described in Claim 1[a], Olbrich teaches this limitation. EX1003, ¶111.

- (e) 2[e]: and each zone is associated with a zone start logical block address and a range of logical block addresses;**

As described in Claim 1[b], Olbrich teaches this limitation. EX1003, ¶112.

- (f) **2[f]: in response to a write command from the host to write data within the range of logical block addresses of a specified zone, perform a multi-plane write of said data to erase units on different planes in the specified zone using a common page address for the respective erase units on the different planes;**

As described in Claim 1[c], Olbrich teaches this limitation. EX1003, ¶113.

- (g) **2[g]: generate metadata based on accessing and maintenance of erase units in the specified zone; and;**

As described in Claim 1[d], Olbrich teaches this limitation. EX1003, ¶114.

- (h) **2[h]: based on the metadata, make a notification available to the host that recommends performing maintenance on the specified zone, wherein the recommendation is associated with the zone start logical block address of the specified zone.**

As described in Claim 1[e], the combined system of Olbrich and Yamada renders this limitation obvious. EX1003, ¶115.

- 3. Claim 3: The SSD of claim 2 further comprising: additional non-volatile memory, wherein the logic further operates to store the generated metadata in the additional non-volatile memory.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶116-118. *See supra* Claim 2.

Olbrich teaches that the controller has logic to make use of “extra bytes,” *additional non-volatile memory*, to “store metadata,” *wherein the logic further operates to store the generated metadata in the additional non-volatile memory*.

EX1004, 17:55-18:2, 16:8-36. Additionally, Olbrich teaches SuperBlock metadata may be stored in a separate SuperPage. *Id.*, 19:16-19.

4. Claim 4: The SSD of claim 2 wherein the logic further operates to compare the metadata with at least one threshold, and wherein making the notification available to the host is dependent on the comparison.

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶119-124. *See supra* Claim 2.

As shown in Claim 1[d]-[e], Olbrich teaches logic to compare SuperBlock metadata against a threshold, *e.g.*, “**Garbage Collection Thresholds**,” *where based on comparison of SuperBlock metadata to those thresholds*, “garbage collection operations” are initiated. EX1004, 80:1-24.

Olbrich additionally teaches the use of a “**threshold**” for “erase count” to select SuperBlocks for maintenance. *Id.*, 91:64-92:22, 41:13-18, 86:24-35, 47:50-54.

Olbrich thus teaches *the logic further operates to compare the metadata with at least one threshold*. EX1003, ¶¶121-123.

A POSA would be motivated to incorporate, into Olbrich’s controller, Yamada’s teaching regarding notifying the host of the need for maintenance. EX1003, ¶124. *See supra* Claim 1[e]. Yamada teaches that the controller’s “maintenance request unit determines whether an area (in units of blocks) ... needs maintenance based on the number of errors” EX1005, [0081]. The controller

compares that metadata to a “criterion value” or “threshold” [***logic further operates to compare the metadata with at least one threshold***], and “transmits a maintenance request to the [host]” [***wherein making a notification available to the host is dependent on the comparison***]. *Id.*, [0081], [0189].

5. Claim 5: The SSD of claim 2

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶125-132. *See supra* Claim 2.

- (a) **wherein the logic further operates to determine, based on the metadata, that maintenance is recommended for the specified zone and to identify the zone start logical block address associated with the specified zone for which maintenance is recommended;**

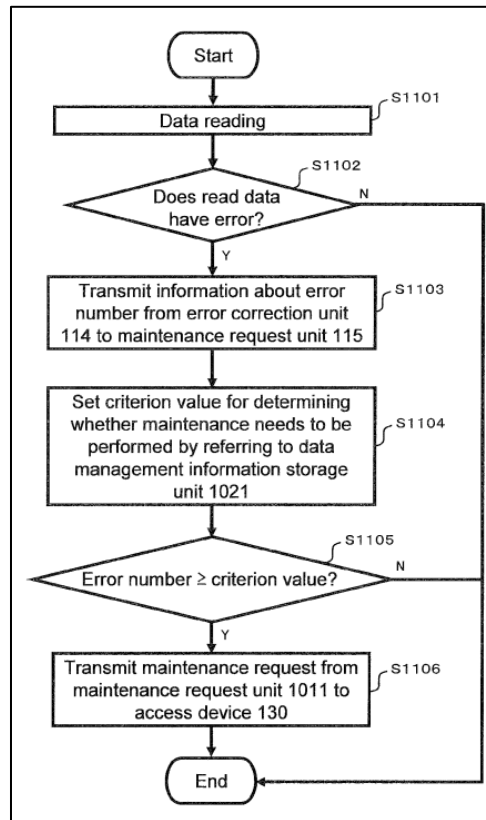
As described for Claim 4, the combined system of Olbrich, Yamada and NVMe1.0b, discloses an SSD with logic ***to determine, based on the metadata, that maintenance is recommended for the specified zone***. EX1003, ¶127.

As described for Claim 1[e], the combined system of Olbrich, Yamada and NVMe1.0b, discloses an SSD with logic ***to identify the zone start logical block address associated with the specified zone for which maintenance is recommended***. EX1003, ¶128.

- (b) wherein determining that maintenance is recommended for the specified zone, identifying the zone start logical block address associated with the specified zone, and making the notification available to the host are all unsolicited by the host.

As shown in Claim 1[e], Yamada teaches that the maintenance request unit notifies the host that maintenance is recommended. EX1003, ¶¶129-132.

Yamada teaches that the maintenance request process, including the metadata analysis and host notification, is unsolicited by the host. Figure 11 illustrates that the process of identifying data for maintenance and recommending maintenance may be *performed entirely without the host*:



EX1005, Fig. 11, [0185]-[0185], [0048], [0178].

Incorporating this teaching from Yamada would be obvious to try in the combination system. EX1003, ¶131. For example, NVMe1.0b—to which the combination system complies pursuant to Claim 1—already describes the utility of unsolicited host notification, describes “an asynchronous event ... issued to the host,” based on comparing block capacity to a threshold. EX1006, § 5.2, Figs. 31, 59.

6. **Claim 6: The SSD of claim 2 wherein the logic further operates to receive a command from the host; and, in response to said command, transmit to the host (a) information representing list of host accessible zones including a zone start logical block address for each listed zone, and (b) a host accessible capacity for each listed zone.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶133-140. *See supra* Claim 2.

Olbrich further teaches maintaining a “SuperBlock Freelist ... contain[ing] identifiers for the SuperBlocks... that are free, and therefore available to be written” [**information representing list of host accessible zones**]. EX1004, 41:3-6, 79:46-59. Olbrich discloses that in the “SuperBlock Metadata Table,” each SuperBlock is marked with an open, closed, free and/or defect flag—which a POSA would understand to collectively represent host accessibility. *Id.*, 48:30-46; EX1003, ¶135. The SuperBlock Metadata table thus describes the total amount of SuperPages (and LBAs) accessible to the host [**a host accessible capacity for each listed zone**].

Olbrich teaches that the “SuperBlock Metadata Table LBA field” includes the LBA for SuperPage 0 in Bank 0 [*a zone start logical block address for each listed zone*]. EX1004, Fig. 35, 88:44-49.

To the extent Olbrich does not teach this limitation, a POSA would be motivated to incorporate the teaching of the NVMe1.0b regarding the utility and value of maintaining zone (Superblock) metadata in list form and sharing that information with the host—as the combined system already complies with that standard, *supra* Claim 1[f]. EX1003, ¶139. Specifically, NVMe1.0b teaches that the host requests, and the controller provides, metadata about namespaces—defined as “a collection of logical blocks”—including the “Number of Namespaces,” [list of host accessible zones] “Namespace Utilization,” and “Namespace Capacity” [collectively representing host accessible capacity for each listed zone]. EX1006, Fig. 67.

A POSA would understand that in the combined system, to incorporate the list metadata functionality, the metadata would be adapted to SuperBlocks and when the controller *transmits it to host* the metadata describes SuperBlocks (as corroborated by the combined system’s compliance with NVMe1.0b, the existing SuperBlock Freelist (*a list of host accessible zones*) and SuperBlock Metadata Table (*host accessible capacity*), and host-transmission functionality). EX1003, ¶140.

7. Claim 8: The SSD of claim 2 wherein:

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶141-145. *See supra* Claim 2.

- (a) the logic further operates to receive read commands and write commands from the host, and to execute the read commands and the write commands by transferring associated data between the host and the NAND flash memory;**

Olbrich teaches that the logic of the controller is configured to “accept[] commands from a host” including “a Host-initiated read or write command” [*the logic further operates to receive read commands and write commands from the host*]. EX1004, 6:1-8; EX1003, ¶144. For write commands, the host “provides the data to the SSD controller” to write in flash memory, and for read commands, the controller “convey[s] the data [in flash memory] to the Host,” EX1004, 6:1-8 [*to execute the read commands and the write commands by transferring associated data between the host and the NAND flash memory*]. *Id.*, 8:64-65, 79:20-23, Figs. 1, 4, 5; EX1003, ¶144.

- (b) and each read command directed to the SSD is accompanied by a logical read address, each logical read address has a first address portion which identifies one of the zones according to a value of the first portion, and a second address portion which identifies a physical memory location in the SSD in a manner that depends on the first address portion.**

Olbrich teaches that host initiated read and write commands include a “Command Descriptor Block (‘CDB’)” which “specifies the address range to be read

from or written to, in Logical Block Addresses (‘LBAs’)” [***and each read command directed to the SSD is accompanied by a logical read address***]. EX1004, 6:1-8.

Olbrich teaches that a read command includes “the LBA of the data” [***each logical read address***]. *Id.*, 8:36-7, 46:30-37. Olbrich teaches that the “LBA received from a Host ... is divided by 15, with the whole-number quotient used” to identify the SuperBlock—the quotient represents ***a first address portion which identifies one of the zones according to a value of the first portion***. *Id.*, 46:30-37, 89:30-32; EX1003, ¶145. Olbrich further teaches that “the actual SPage Sector in the SuperPage [the precise physical location] is obtained by taking the remainder from the division” [***a second address portion which identifies a physical memory location in the SSD in a manner that depends on the first address portion***]. EX1004, 46:51-60; EX1003, ¶145.

8. Claim 11: The SSD of claim 2 wherein the flash memory comprises multiple dies of flash memory and the erase units on different planes are also in different dies.,

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶146-148. *See supra* Claim 2.

Olbrich teaches that SuperBlocks comprise pages and blocks which span “***four different Flash Memory Dies*** in the same Flash Bank” [***wherein the flash memory comprises multiple dies of flash memory and the erase units on different planes are also in different dies***]. EX1004, 16:12-67.

9. **Claim 12: The SSD of claim 2 wherein the logic further operates to, upon the performance of the multi-plane write, update the metadata to indicate an amount of unerased host accessible space in the specified zone.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶149-151. *See supra* Claim 2.

As shown in Claim 1[c], Olbrich teaches a controller with logic to *perform a multi-plane write*.

Olbrich further teaches that as the constituent SuperPages are written, the “Metadata Table,” “is written with the appropriate LBA, and the SuperPage Time Stamp entry in Column 3505 is written with the time the SuperPage was written.” EX1004, 48:47-55. The available “room for a write,” *e.g.*, the number of “free SuperPages shown on the SuperPage Pointer list,” is tracked [*logic further operates to ... update the metadata to indicate an amount of unerased host accessible space in the specified zone*]. *Id.*, 79:60-67; EX1003, ¶151.

10. **Claim 13: The SSD of claim 12 wherein the logic further operates to: upon the performance of the multi-plane write, detect that the specified zone contains no erased host accessible space; and, in response to said detection, transition the specified zone to a full state.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶152-155.

As shown in Claim 12, the combined system of Olbrich, Yamada and NVMe1.0b teaches ***the SSD of claim 12, wherein the controller has logic to perform a multi-plane write.***

Olbrich teaches that the controller has logic to classify a SuperBlock as “closed.” EX1004, 77:59-67. After data has been written “and the currently open ***SuperBlock is full***” “that SuperBlock must be [transitioned to] closed” and a “closed flag is set.” [***detect that the specified zone contains no erased host accessible space; and, in response to said detection, transition the specified zone to a full state***]. EX1004, 77:59-67, 79:46-67, 48:44-46; EX1003, ¶155.

11. Claim 15: The SSD of claim 2 wherein the logic further operates to receive a maintenance command directed to the specified zone and, in response to the maintenance command, erase host accessible space only within the specified zone and transition the specified zone to an empty state.

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶156-159.

As described in Claims 1 and 2, the combination of Olbrich and Yamada teaches ***the SSD of claim 2 wherein the logic further operates to receive a maintenance command directed to the specified zone.*** The combined system incorporates Yamada’s teaching that the controller receives a maintenance command—a POSA would understand that executing the maintenance command includes erasing the blocks identified in the maintenance command (consistent with

the teaching in Olbrich) [*in response to the maintenance command, erase host accessible space only within the specified zone*]. EX1004, 91:64-92:22; EX1003, ¶158. Olbrich teaches that after erasure of the SuperBlock pursuant to maintenance, the SuperBlock is transitioned to a “free” state and placed on the “SuperBlock FreeList” [*and transition the specified zone to an empty state*]. *E.g.*, EX1004, Fig. 56, 83:6-9.

12. **Claim 16: The SSD of claim 2 wherein the metadata is dependent on an erase count for the specified zone; and the logic further operates to compare the metadata with at least one threshold for erase count, wherein making the notification available to the host is dependent on the comparison.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶160-164.

As described in Claims 1, 2 and 4, the combination of Olbrich, Yamada and NVMe1.0b teaches *the SSD of Claim 2 with logic to compare the metadata with at least one threshold ... wherein making the notification available to the host is dependent on the comparison.*

Olbrich further teaches that SuperBlock metadata may include “a count of the number of times the SuperBlock has been erased, [which] can be used, for example, to prioritize SuperBlocks with higher erase counts for garbage collection” [*wherein the metadata is dependent on an erase count for the specified zone*]. EX1004, 47:50-54. And, Olbrich teaches erase count thresholds to trigger maintenance

operations [*and the logic further operates to compare the metadata with at least one threshold for erase count*]. EX1004, Claims 1, 24, 47:50-54, 92:18-22; EX1003, ¶163.

A POSA would be motivated to rely on the erase count metadata as a threshold for triggering a notification to the host in the combination system, as Yamada already teaches a controller with logic to “transmit[] a maintenance request” to the host based on comparison against a “criterion value” or “threshold.” EX1005, [0081], [0189]; EX1003, ¶164. Indeed, Yamada teaches that the controller in “the nonvolatile memory system ... dynamically changes the criterion value used to determine whether maintenance needs to be performed,” EX1005, [0183], [0189], thus it would be obvious to rely on Olbrich’s additional forms of SuperBlock metadata such that *making the notification available to the host is dependent on the comparison*. EX1003, ¶164.

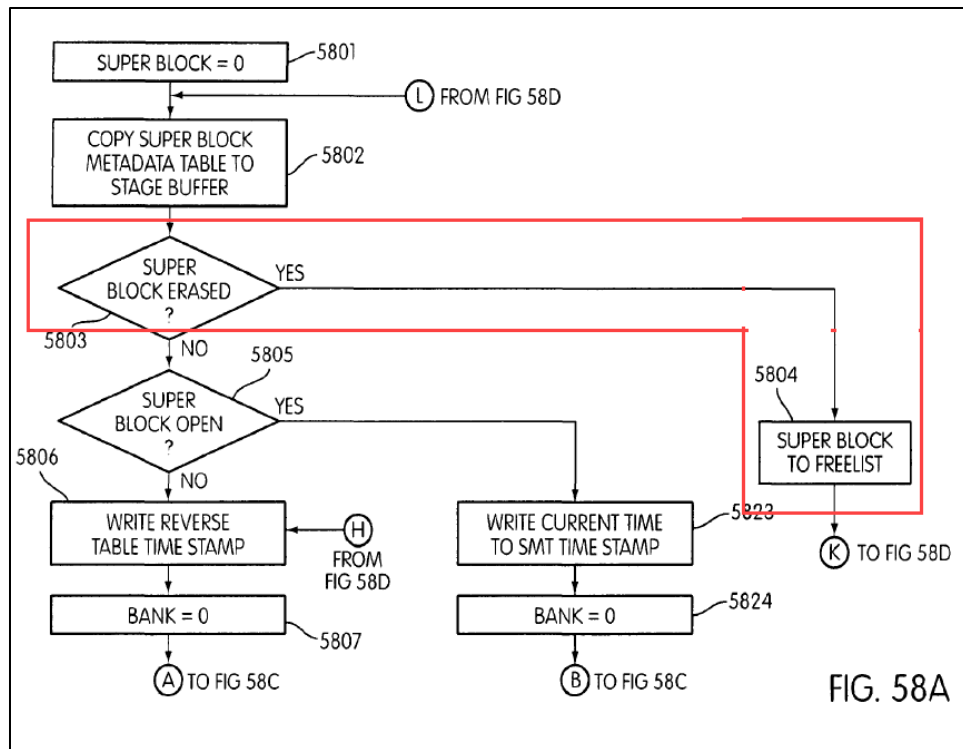
13. **Claim 17: The SSD of claim 2 wherein the logic further operates to: in response to a maintenance command from the host directed to the zone start logical block address of the specified zone, perform an erase of erase units in the specified zone, and transition a state of the specified zone to an empty state whereby all host accessible pages of the specified zone are available for writing by the host.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶165-169.

As described in Claim 15, the combined system of Olbrich, Yamada and NVMe1.0b, teaches ***the SSD of claim 2 wherein the logic further operates to: in response to a maintenance command from the host ... perform an erase of the erase units in the specified zone, and transition a state of the specified zone to an empty state.***

As described in Claim 1[b]-[c], Olbrich teaches that a command to a SuperBlock may be directed to a “***starting address***” [a zone start logical block address], representing the first SuperPage in the SuperBlock. A POSA would understand that in the combined system the maintenance command would include the ZSLBA [***maintenance command from the host directed to the zone start logical block address of the specified zone***]. EX1004, 72:33-38, Fig. 16; EX1003, ¶168.

Olbrich teaches that, after erasure in maintenance, the SuperBlock is transitioned to a state of “free,” meaning the SuperBlock is empty and is “therefore available to be written” by the host [***and transition a state of the specified zone to an empty state whereby all host accessible pages of the specified zone are available for writing by the host***]. EX1004, 41:3-7; 79:46-59. This process is shown in, e.g., Figure 58A (annotated red).



Id., Fig. 58A.

- 14. Claim 18: The SSD of claim 17 wherein the maintenance command from the host is in response to the notification that recommends performing maintenance on the specified zone.**

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶170-172.

As described in Claim 1[e], the combined system of Olbrich, Yamada and NVMe1.0b teaches a maintenance request from the controller to the host for performing maintenance on SuperBlock, *to the notification that recommends performing maintenance on the specified zone*. A POSA would understand that the maintenance request in the combined system would lead to *a maintenance*

command from the host is in response to the notification because Yamada teaches that upon “receiving a maintenance request from the nonvolatile memory module,” the host “generat[es] a maintenance command.” EX1005, [0100], Claim 1; EX1003, ¶172.

15. Claim 19: The SSD of claim 17 wherein the metadata is dependent on erase counts for erase units in the specified zone; and the logic further operates to update the metadata responsive to performance of the erase.

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶173-176. *See supra* Claim 17.

As described for Claim 16, the combined system of Olbrich, Yamada and NVMe1.0b teaches a maintenance operation *wherein the metadata is dependent on erase counts for erase units in the specified zone*.

Olbrich teaches that the controller has logic which “increments Erase Count Field 3507 in the SuperBlock Metadata Table, reflecting the fact that the SuperBlock has now been erased” [*logic further operates to update the metadata responsive to performance of the erase*]. EX1004, 82:57-60.

16. Claim 20: The SSD of claim 17 wherein performing the erase of erase units in the specified zone comprises performing the erase on erase units of different planes in the specified zone

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶177-179.

As described for Claim 17, Olbrich teaches the SSD of claim 17 with logic for ***performing the erase of erase units in the specified zone.***

As described for Claim 1[a], Olbrich teaches an erasure of a SuperBlock comprising erase units on different planes, ***performing the erase on erase units of different planes in the specified zone.***

17. Claim 22: The SSD of claim 2

The combined system of Olbrich, Yamada and NVMe1.0b renders this claim obvious. EX1003, ¶¶180-186. *See supra* Claim 2.

- (a) wherein the logic further operates to, in response to a write command from the host to the zone start logical block address of the specified zone:**

As described for Claim 1[b]-[c], Olbrich teaches a controller “may receive a ***write command***” ***from the host***, EX1004, Fig. 5, which includes a starting address for a SuperBlock, *id.*, 72:33-38, 15:27-49 [***zone start logical block address of the specified zone***]. EX1003, ¶182. *See also* EX1004, 6:1-8.

- (b) assign a logical block address within the range of logical block addresses of the specified zone, transfer data from the host to the specified zone at the assigned logical block address,**

Olbrich teaches a controller which may assign LBAs within the range of LBAs comprising a SuperBlock. Upon receiving a write command, identifying a SuperBlock, the controller starts a “Input-Output Process” (“IOP”) which is comprised of “Transfer Requests” to transfer the data from the host to the memory.

Id., 6:1-24. Olbrich teaches, as part of the IOP process, the controller may “automatically” assign LBAs in the write destination within the “LBA range” as part of the IOP process [***assign a logical block address within the range of logical block addresses of the specified zone***]. *Id.*, 6:1-24, 86:48-55; EX1003, ¶183.

Based on the “Transfer Request[s],” the controller “write[s] a segment of Flash Memory Module 108 corresponding to a portion of the LBA range specified by the Transfer Request” [***transfer data from the host to the specified zone at the assigned logical block address***]. EX1004, 6:9-24, 52:63-53:4; EX1003, ¶184.

As an additional disclosure rendering this limitation obvious, Olbrich teaches that as data is written into a range of LBAs in a SuperBlock, the controller determines and stores the LBAs in which data is written. EX1004, 48:47-54.

(c) and return the assigned logical block address to the host.

Yamada teaches that, upon completion of a write operation (*e.g.*, as part of maintenance operations), the controller transmits to the host a “maintenance completion notification [which] contains a logical address of a block ... for which maintenance has been completed” [***return the assigned logical block address to the host***]. EX1005, [0108]-[0111]. The host is provided this information because it maintains mapping information for data in the memory, *e.g.*, *id.* [0101]. In the combined system of Olbrich and Yamada, a POSA would be motivated to incorporate this teaching in order to provide the host the logical addressing

information necessary for host-initiated maintenance commands already incorporated. EX1003, ¶186; *supra* Claim 1[e].

18. Claim 25: The SSD of claim 2

The combined system of Olbrich, Yamada and NVMe1.0b renders this limitation obvious. EX1003, ¶¶187-191. *See supra* Claim 2.

(a) wherein the metadata is dependent on a time period since data was written to the specified zone;

Olbrich teaches metadata which “indicat[es] the time when each SuperBlock was originally closed for writing” and when each constituent SuperPage was written [*the metadata is dependent on a time period since data was written to the specified zone*]. EX1004, 47:45-48:38, 18:8-9; EX1003, ¶189.

To the extent Olbrich does not teach this limitation, a POSA would be motivated to incorporate the broader metadata variant taught by Yamada—“*access history of the nonvolatile memory 120 (information about the time at which data has been read from or written to the memory)*”—into the combined system’s controller functionality [*wherein the metadata is dependent on a time period since data was written to the specified zone*]. EX1005, [0201]; EX1003, ¶190.

(b) and the logic further operates to compare the metadata with at least one threshold, wherein making the notification available to the host is dependent on the comparison

As described for Claim 1[e], Yamada teaches the benefits of comparing metadata against a threshold, a “criterion value,” to determine the need for a

maintenance request. Yamada further teaches that the metadata threshold comparison may be based on “access history.” EX1005, [0032]-[0033], [0187], [0201], [0204], [0211], Fig. 10. A POSA would understand that, to incorporate this teaching into the combined system, this functionality must be added to the controller where the metadata and related analysis occur, and that the resulting maintenance request would be directed to a SuperBlock. EX1003, ¶191. The combined system thus teaches a controller with *logic that further operates to compare the metadata with at least one threshold, wherein making the notification available to the host is dependent on the comparison. Id.*

19. Claim 26: The SSD of claim 25 wherein comparing the metadata with at least one threshold and making the notification available to the host are unsolicited by the host

The combined system of Olbrich, Yamada, and NVMe1.0b renders this claim obvious. EX1003, ¶¶192-194.

As described for Claims 5 and 25, the combined system of Olbrich, Yamada and NVMe1.0b teaches *the SSD of claim 25 wherein comparing the metadata with at least one threshold and making the notification available to the host are unsolicited by the host.*

20. Claim 27: The SSD of claim 2 wherein logic further operates to, in response to a command from the host,

transmit to the host information dependent on a time period since data was written to the specified zone.

The combined system of Olbrich, Yamada, and NVMe1.0b renders this claim obvious. EX1003, ¶¶195-197. *See supra* Claim 2.

Yamada teaches that the host may request and “obtain[] the write history”—“information about the time at which data has been previously written.” EX1005, [0163]-[0164], *see supra* Claims 1[e], 25. A POSA would understand that to incorporate this teaching in the combined system, where metadata is tracked by the controller, in order to provide this metadata to the host (for *e.g.*, use in maintenance operations) the controller may transmit it to the host in response to a command or query [*and in response to a command from the host, transmit to the host information dependent on a time period since data was written to the specified zone*]. EX1003, ¶197.

21. Claim 28: The SSD of claim 2 wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and transmit to the host the zone start logical block address and a state of the specified zone.

The combined system of Olbrich, Yamada, and NVMe1.0b renders this claim obvious. EX1003, ¶¶198-204. *See supra* Claim 2.

Olbrich teaches that a controller may detect a “physical defect [which] is permanent” in a SuperBlock and set a “Defect Flag”—*wherein the logic further operates to detect a failure condition of one of the erase units of the specified*

zone—and teaches the controller’s “issuance of an appropriate ... code to the Host, indicating a [] failure”—***transmit to the host the zone start logical block address and a state of the specified zone***. EX1004, 84:22-49, 48:13-20, 44:64-45:7, 16:64-17:27; EX1003, ¶200-202.

To the extent Olbrich does not teach this claim, Yamada teaches that the controller contains logic to detect when the number of errors in a block amounts to a failure condition [***logic further operates to detect a failure condition of one of the erase units of the specified zone***], triggering a maintenance request with the “logical address of the start of the maintenance-needed block.” EX1005, [0032], [0006], [0101]. A POSA would understand that to apply Yamada’s teaching to the combination system, the starting logical block address would correspond to a SuperBlock [***transmit to the host the zone start logical block address***]. EX1003, ¶203.

To the extent the combination of Yamada and Olbrich does not teach transmitting to the host the “state of the specified zone,” a POSA would be motivated to incorporate the teaching of the NVMe1.0b in which the controller may report information to the host including “SMART / Health functionality,” where the underlying health [***state***] information is sent per namespace. EX1006, Fig. 59, § 5.10.1.2; EX1003, ¶204.

A POSA would understand that to apply these teachings to the combination system in Olbrich, the failure state would correspond to a SuperBlock [*a state of the specified zone*]. EX1003, ¶204.

22. Claim 29: The SSD of claim 2 wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and, in response to a write command from the host within the range of logical block addresses of the specified zone, transmit to the host a notification of a write failure.

The combined system of Olbrich, Yamada, and NVMe1.0b renders this claim obvious. EX1003, ¶¶205-210.

As shown for Claim 28, the combined system of Olbrich, Yamada, and NVMe1.0b teaches the SSD of claim 2, wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and ... transmit to the host the state of the zone [*a failure condition*].

Olbrich teaches that a defect flag for a SuperBlock may be identified “as being defective during a flash read or a write” [*wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and, in response to a write command from the host within the range of logical block addresses of the specified zone*]. EX1004, 48:13-20.

To the extent Olbrich does not teach this claim, a POSA would be motivated to incorporate the teaching of Yamada regarding the maintenance of metadata including the errors in “data that is written to nonvolatile memory,” where a

maintenance request may be sent to the host based on that metadata. EX1005, [0081]. To incorporate this teaching into the combination system, a POSA would understand the write failure would correspond to a SuperBlock [*wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone and, in response to a write command from the host within the range of logical block addresses of the specified zone*]. EX1003, ¶¶208, 210.

To the extent the combination of Yamada and Olbrich does not teach this limitation, A POSA would be motivated to incorporate NVMe1.0b's teaching of an explicit *notification of a write failure in response to a write command from the host*. EX1006, § 6.9.1 Command Completion; EX1003, ¶209.

23. Claim 30: The SSD of claim 2 wherein the metadata is dependent on read errors detected for the specified zone.

The combined system of Olbrich, Yamada, and NVMe1.0b renders this claim obvious. EX1003, ¶¶211-214. *See supra* Claim 2.

Yamada further teaches that the metadata triggering a maintenance request may be “the number of errors included in a data read” [*wherein the metadata is dependent on read errors detected for the specified zone*]. EX1005, [0032], [0081]-[0082]; EX1003, ¶¶213-214.

IX. GROUND II: CLAIMS 1-6 ARE RENDERED OBVIOUS BY GONZALEZ IN LIGHT OF GOSS AND NVME1.0B

A. Overview of the Prior Art

1. Gonzalez (EX1007)

U.S. Pub. Patent 2010/0049908 (“Gonzalez”) was filed on October 30, 2009, and was published on February 25, 2010. It is prior art under AIA 35 U.S.C. § 102.

Gonzalez describes a memory system organized into units of metablocks, where each metablock includes blocks from different planes to increase parallelism and performance. EX1007, [0013]-[0018]; EX1003, ¶61.

2. Goss (EX1008)

U.S. Pat. Pub. No. 2011/0302477 (“Goss”) was filed on June 3, 2010, and was published December 8, 2011. It is prior art under AIA 35 U.S.C. § 102.

Goss teaches a memory system where the controller maintains metadata regarding the status of data, including, *e.g.*, wear information, where this information is provided to the host for maintenance. EX1008, [0021], [0024], [0048], [0085]; EX1003, ¶¶63-64.

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Goss and NVMe1.0b with Gonzalez. Specifically, a POSA would have been motivated to (i) incorporate Goss’s teaching regarding notification to the host regarding the status of

data and the need for maintenance, and (ii) implement various aspects of the NVMe1.0b standard. EX1003, ¶¶215-219.

First, Gonzalez and Goss involve similar techniques to solve the same problem, *i.e.*, collecting, storing and using metadata for maintenance activities to improve the efficiency and performance of NAND flash memory systems. *E.g.*, EX1007, [0003]-[0008], [0066], [0075], [0088]; EX1008, [0005], [0021], [0024], [0048], [0085]; EX1003, ¶216.

Second, a POSA would have been motivated to incorporate Goss’s teaching, maintenance commands by the host based on metadata and notifications provided by the controller, because Gonzalez already teaches that memory management responsibilities may be distributed between the host and controller, *i.e.*, including “direct control by the host system.” EX1007, [0037]. A POSA would have had a reasonable expectation of success in incorporating this shared host functionality—adding controller notification to the host and translocating the maintenance command responsibility to host-side software—and it would have been straightforward and routine. EX1003, ¶217.

Third, a POSA would be motivated to ensure the combination system complies with the NVMe1.0b, as it represents a then-contemporaneous widely adopted industry standard for NAND flash memory devices. EX1003, ¶218.

C. Claim-by-Claim Analysis

1. Claim 1[pre]

Gonzalez teaches the preamble. EX1003, ¶¶220-222.

To the extent the preamble is limiting, it is disclosed by Gonzalez. Gonzalez discloses “non-volatile memory products” with “*NAND*” *flash memory comprising “block[s] [a]s the erase unit,” where “[e]ach block typically stores one or more pages of data.”* EX1007, [0003]-[0008]. Gonzalez discloses methods *for providing a host with access to an SSD that contains ... logic comprising circuitry and/or a processor executing instructions.* EX1007, [0011], Figs. 1, 2.

(a) Claim 1[a]

Gonzalez teaches or renders obvious this limitation. EX1003, ¶223.

Gonzalez teaches that to “*blocks may be linked together as metablocks*” including “*one physical block from each plane*” [dividing the flash memory into zones; wherein the zones are non-overlapping subdivisions of the flash memory, each zone comprises two or more erase units, the erase units in each zone are on two or more different planes]. EX1007, [0010], [0018].

(b) Claim 1[b]

Gonzalez teaches or renders obvious this limitation. EX1003, ¶¶224-226.

Gonzalez teaches methods of mapping metablocks including those using “starting LBAs” [*and each zone is associated with a zone start logical block address*]. EX1007, [0076]; EX1003, ¶225. Gonzalez further teaches that each

“metablock is a logical grouping of blocks ... in some range of consecutive host addresses” [*a range of logical block addresses*]. EX1007, [0044], [0018]; EX1003, ¶226.

(c) Claim 1[c]

Gonzalez teaches or renders obvious this limitation. EX1003, ¶¶227-230.

Gonzalez teaches that write commands may be directed to metablocks where “[d]ata are simultaneously programmed to one block of each plane making up a metablock” spanning “a number of consecutively addressed data sectors” [*in response to a write command from the host to write data to the range of logical block addresses of a specified zone, performing a multi-plane write of said data to erase units on different planes in the specified zone*]. EX1007, [0057], [0018]; EX1003, ¶229.

Gonzalez further teaches that “[t]he metablock is identified by *a common logical address as a destination for programming* and reading *data*,” [using a common page address for the respective erase units on the different planes]. EX1007, [0010].

(d) Claim 1[d]

Gonzalez and/or the combined system of Gonzalez and Goss thus render this limitation obvious. EX1003, ¶¶231-234.

Gonzalez teaches that “programming characteristic[s]” are “monitored and stored by the controller” “on a metablock basis,” including “host accesses” [*generating metadata based on accessing ... of erase units in the specified zone*]. EX1007, [0066], [0075], [0088]. Gonzalez also teaches tracking the “number of times one or more pages or blocks have been written due to host commanded writes versus those due to data consolidation (garbage collection)” [*generating metadata based on ... maintenance of erase units in the specified zone*]. *Id.*

To the extent Gonzalez does not teach this limitation, a POSA would be motivated to combine Goss’s teaching that the controller may monitor the “total number of access operations” and the “number of erase cycles for the erasure block” [*generating metadata based on accessing and maintenance*]. EX1008, [0021]; EX1003, ¶234.

(e) Claim 1[e]

The combined system of Gonzalez and Goss thus renders this limitation obvious. EX1003, ¶¶235-239.

Gonzalez teaches the need to identify circumstances under which maintenance is performed. EX1007, [0052]. A POSA would be motivated to incorporate the teaching of Goss, which describes “methods for enhancing the reliability and integrity of data stored in a non-volatile memory” in SSDs. EX1008, [0018]-[0024]; EX1003, ¶237. Goss teaches that the controller, in the context of wear-leveling, may

include “[s]elf-monitoring routines [to] provide ***real-time notifications to a host*** with regard to the status of the array and/or the data, allowing the user to take affirmative steps to alter the operation of the device to extend its useful life.” EX1008, [0024], [0085].

A POSA would understand that the “status” information provided to the host may be based on erase count or “wear history of each block” and corresponds to a set of cells and includes “overhead control information such ***as the logical addresses of the data***” [***based on the metadata, making a notification available to the host***]. EX1008, [0035], [0048], [0050]; EX1003, ¶¶237-238. A POSA would further understand that the “affirmative steps” in response may include host-initiated maintenance commands [***that recommends performing maintenance***]. *Id.*

In incorporating this teaching, a POSA would understand that the status information corresponding to a set of cells necessarily corresponds to the metablocks in Gonzalez—this implementation is specifically contemplated by Goss, where metadata is not limited to a set of cells but may correspond to a larger memory unit. EX1008, [0076], [0035]. Accordingly, in the combination system, the notification sent to the host would be premised on metadata describing Gonzalez’s metablocks and the notification would identify the metablock in need of maintenance through, *e.g.*, “starting LBA.” [***wherein the recommendation is associated with the zone start logical block address of the specified zone***]. EX1003, ¶239.

(f) Claim 1[f]

As described in Claim 1[pre], the combined system of Gonzalez and Goss teaches an SSD.

Goss teaches that the *host communicates with the SSD controller over a host interface*. EX1008, [0026].

A POSA would be motivated to incorporate the teaching of the NVMe1.0b standard into the combined system of Gonzalez and Goss [*wherein the SSD and the host communicate over a host interface, using commands from a Non-Volatile Memory Express (NVMe) standard*], as it represents a then-contemporaneous widely adopted applicable standard. EX1003, ¶¶240-242.

2. Claim 2

As described in Claim 1[pre], the combined system of Gonzalez teaches an SSD. EX1003, ¶243.

(a) 2[a]

As described in Claim 1[f], Gonzalez, Goss and NVMe1.0b renders this limitation obvious. EX1003, ¶244.

(b) 2[b]

As described in Claim 1[pre], Gonzalez teaches this limitation. EX1003, ¶245.

(c) 2[c]

As described in Claim 1[f], the combined system of Gonzalez, Goss and NVMe1.0b renders this limitation obvious. EX1003, ¶¶246-247. *See also* EX1008, 20:63-21:4.

(d) 2[d]

As described in Claim 1[a], Gonzalez teaches this limitation. EX1003, ¶248.

(e) 2[e]

As described in Claim 1[b], Gonzalez teaches this limitation. EX1003, ¶249.

(f) 2[f]

As described in Claim 1[c], Gonzalez teaches this limitation. EX1003, ¶250.

(g) 2[g]

As described in Claim 1[d], the combined system of Gonzalez and Goss renders this limitation obvious. EX1003, ¶251.

(h) 2[h]

As described in Claim 1[e], the combined system of Gonzalez and Goss renders this limitation obvious. EX1003, ¶252.

3. Claim 3

The combined system of Gonzalez, Goss and NVMe1.0b renders this claim obvious. EX1003, ¶¶253-255. *See supra* Claim 2.

Gonzalez teaches that the metadata may be “be stored as overhead data within the data sectors stored in those blocks, or in separate physical blocks” [*wherein the*

logic further operates to store the generated metadata in the additional non-volatile memory]. EX1007, [0065].

4. Claim 4

The combined system of Gonzalez, Goss and NVMe1.0b renders this claim obvious. EX1003, ¶¶256-262. *See supra* Claim 2.

Gonzalez teaches that the controller functionality includes “**comparing the difference of the frequency of two or more access usage patterns with a threshold number**” [wherein the logic further operates to compare the metadata with at least one threshold]. EX1006, [0065]. Additionally, Goss teaches that the controller “compar[es] the WI value to a predetermined threshold T,” where WI is “wear indication” metadata. EX 1008, [0074]-[0075]. As shown in Claim 1[e], Goss teaches that during controller initiated “monitoring,” the controller may “provid[e] a notification to the host.” *Supra*, 1[e], EX 1008, [0085]. That notification may be triggered based on “data [] be[ing] below retention specifications.” EX 1008, [0085]. A POSA would understand that where the WI metadata satisfies a threshold, a notification is sent to the host identifying the need for maintenance as part of the controller’s self-monitoring [**wherein making the notification available to the host is dependent on the comparison**]. *Id.*; EX1003, ¶¶260-261.

As further evidence the combined system teaches this claim, the NVMe1.0b teaches unsolicited host notification based on a threshold comparison. EX1006, § 5.2, Figs. 31, 59.

5. Claim 5

The combined system of Gonzalez, Goss and NVMe1.0b renders this claim obvious. EX1003, ¶¶263-269. *See supra* Claim 2.

(a) 5[a]

As described for Claim 4, the combined system of Gonzalez, Goss and NVMe1.0b, discloses an SSD with logic ***to determine, based on the metadata, that maintenance is recommended for the specified zone.***

As described for Claim 1[e], the combined system of Gonzalez, Goss and NVMe1.0b, discloses an SSD with logic ***to identify the zone start logical block address associated with the specified zone for which maintenance is recommended.***

(b) 5[b]

As described for Claims 1[e] and 5[a], the combined system of Gonzalez, Goss and NVMe1.0b, discloses an SSD with logic ***wherein determining that maintenance is recommended for the specified zone, identifying the zone start logical block address associated with the specified zone.***

Goss teaches that the notification of the need for maintenance may be made available to the host pursuant to “real-time notifications to a host” during the

controller’s “[s]elf-monitoring routines.” EX1008, [0024], [0085]. A POSA would understand ***making the notification*** available to the host and the underlying metadata comparison, as part of the controller’s self-monitoring routines, ***is unsolicited by the host***, EX1008, [0024]; EX1003, ¶268. *See also* Claim 4.

6. Claim 6

The combined system of Gonzalez, Goss, and NVMe1.0b renders this claim obvious. EX1003, ¶¶270-272. *See supra* Claim 2.

A POSA would be motivated to incorporate the teaching of NVMe1.0b—as the combined system already complies with that standard as described in Claim 1[f]—disclosing this claim, *see supra* Ground I, Claim 6. EX1003, ¶272.

X. GROUND III – CLAIM 8 IS RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, NVME1.0B, AND BAN

A. References

1. Ban (EX1010)

U.S. Pat. No. 5,404,485 (“Ban”) was filed on March 8, 1993, and issued on April 4, 1995. It is prior art under AIA 35 U.S.C. § 102.

Ban teaches a method “to control and manage access to a flash memory so that the flash memory” using various techniques of address construction. EX1010, 1:51-2:55; EX1003, ¶70.

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Ban with the combined system of Gonzalez, Goss and NVMe1.0b, namely, Ban's teaching regarding deriving data location from address information. EX1003, ¶¶273-276.

First, Ban, like Gonzalez and NVMe1.0b, teaches the use of groupings of blocks as the unit of a flash memory, "zones." EX1010, 2:11-19. Ban, like Gonzalez, manages memory, including mapping and maintaining status information, based on those units. EX1010, Claim 1; EX1003, ¶274.

Second, a POSA would have been motivated to combine the teaching of Ban, representing a discrete teaching regarding a foundational means to make addressing more efficient and effective by enabling the host to access and direct data in a targeted manner. A POSA would be able to implement this teaching with a reasonable expectation of success, using well-known, straightforward and routine techniques. EX1003, ¶275.

C. Claim-by-claim Analysis

1. Claim 8

The combined system of Gonzalez, Goss, NVMe1.0b, and Ban renders this claim obvious. EX1003, ¶¶277-282. *See supra* Claim 2.

(a) 8[a]

Gonzalez teaches that the *host provides “logical block addresses (LBAs)” in read and write commands to the controller*, where, in write commands, data is *transferred from the host to the NAND memory and in read commands data is transferred from the NAND memory to the host*. EX1007, [0010], [0037].

(b) 8[b]

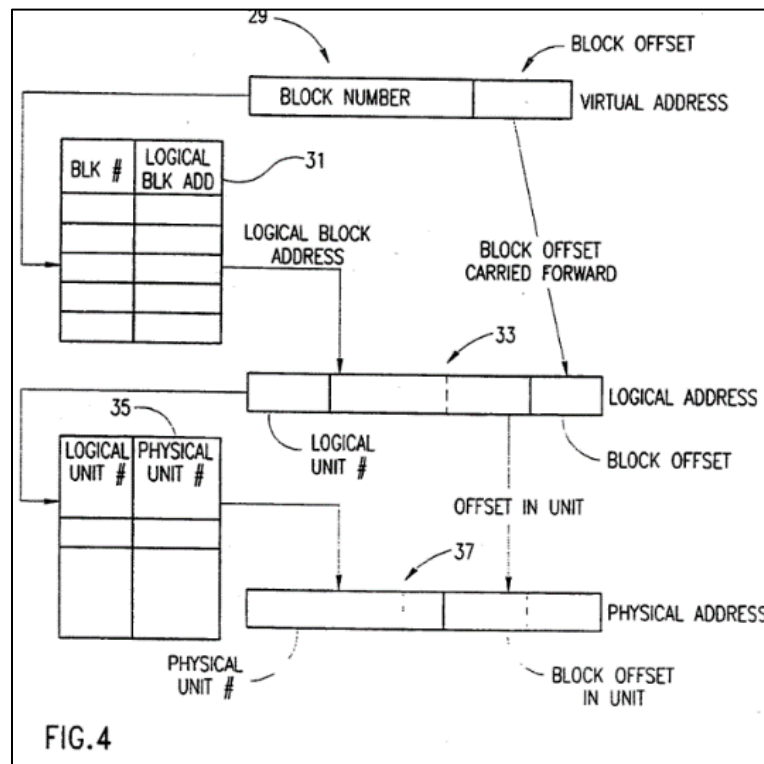
Gonzalez teaches that the host provides “logical block addresses (LBAs)” in read and write commands to the controller, where in response the controller provides a “translation between logical block addresses (LBAs) received from a host, and physical block numbers (PBNs) and page addresses within the memory cell array.” EX1007, [0010]. Because LBA translation needs to be unique, an LBA must map on one PBN and one page address, thus the LBA (comprised of a set of address bits) must be partitioned into two disjoint subsets (portions) of address bits.

A POSA would understand that these subsets, used for deriving the PBN and page addresses from the LBA *may rely on a first portion and a second portion of an LBA depending on the first portion*. *Id.*; EX1003, ¶280. The derived PBN, *the first portion, identifies a metablock (zone)* based on the LBA—“Data with a given logical block address (LBA) is mapped by the controller into a specific physical memory block within that plane that is identified by a physical block number

(PBN),” and the derived Page Addresses, *the second address portion, identifies a physical memory location in the SSD.* EX1007, [0047]; EX1003, ¶280.

Alternatively, a POSA would be motivated to incorporate the teaching of Ban which describes methods of optimizing mapping in flash memory. EX1003, ¶281; EX1010, 2:15-19. Ban represents known methods for addressing in memory systems. Ban teaches that a host supplies an address which is composed of two derivable portions directed to “block number and a block offset,” where the controller relies on those portions to identify the physical block for the data and the location of the data within the block respectively, EX1010, 4:32-44, as illustrated in

Figure 4:



Id., Fig. 4. A POSA would understand that to implement this addressing in the system in Gonzalez, the first portion of the address would identify the metablock, and the second portion, based on the first, would identify the physical location within the metablock. EX1003, ¶282.

XI. GROUND IV - CLAIM 11 IS RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, NVME1.0B, AND SPROUSE

A. References

1. Sprouse (EX1011)

U.S. Pat. Pub. No. 2011/0153911 (“Sprouse”) was filed on December 18, 2009, published June 23, 2011, and issued on April 4, 1995. It is prior art under AIA 35 U.S.C. § 102.

Sprouse describes a memory system comprised of multiple dies a metablock may be comprised of blocks spanning multiple dies, “achieving memory die parallelism at block level.” EX1011, [0004]-[0007]; EX1003, ¶72.

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Sprouse with the combined system of Gonzalez, Goss and NVMe1.0b, namely, Sprouse’s teaching regarding metablocks composed of blocks from different dies. EX1003, ¶¶283-288.

First, Sprouse is a contemporaneous patent to Gonzalez that is also assigned to SanDisk Technologies LLC. EX1003, ¶285.

Second, Sprouse, like Gonzalez, teaches the use of groupings of blocks as the unit of NAND flash memory, “metablocks,” to achieve higher degrees of parallelism. EX1011, [0027], [0032]. EX1003, ¶286.

Third, a POSA would have been motivated to combine the teaching of Sprouse given that a central aim of Gonzalez is to increase parallelism and Sprouse provides an additional means to bolster parallelism. EX1011, [0004]-[0007], EX1003, ¶287.

Considering the commonality in organization, shared goal of increasing parallelism, and shared assignee, a POSA would have been motivated to incorporate the teaching of Sprouse, representing complementary techniques in the same field of endeavor and could have done so with a reasonable expectation of success using routine techniques. EX1003, ¶287.

C. Claim-by-claim Analysis

1. Claim 11

The combined system of Gonzalez, Goss, Sprouse and NVMe1.0b renders this limitation obvious. EX1003, ¶¶289-291. *See supra* Claim 2.

A POSA would be motivated to incorporate the teaching of Sprouse, including the improved parallelism achieved by “metablock level interleaving” across dies, [*wherein the flash memory comprises multiple dies of flash memory and the erase units on different planes are also in different dies*]. EX1011, [0047].

XII. GROUND V – CLAIMS 12, 13, 15-20, 22, 25-27 ARE RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, NVME1.0B, AND SON

A. References

1. SON (EX1009)

U.S. Patent Application No. 2012/0096217 (“Son”) was filed on Oct. 14, 2011, and was published on April 19, 2012, and is prior art under at least AIA 35 U.S.C. § 102.

Son teaches an SSD management system which relies on sharing of file system (host) information and controller information to improve performance in shared host memory management. EX1003, ¶¶66-67. Son’s system is subdivided into SuperBlocks which are “a grouping of ... blocks.” EX1005, [0029]; EX1003, ¶68.

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Son into the combined system. Specifically, a POSA would have been motivated to incorporate Son’s teaching regarding (i) additional forms of SuperBlock metadata and (ii) the sharing of SuperBlock metadata with a host for purposes of memory management. EX1003, ¶¶292-297.

First, Son, like Gonzalez and Goss, involves similar techniques to solve the same problem, *i.e.*, collecting, storing and using metadata for maintenance activities

to improve the efficiency and performance of NAND flash memory systems. *E.g.*, EX1009, [0002], [0014], [0029], [0038], [0078]-[0080]. Son, like Gonzalez, also relies on SuperBlock-level metadata to optimize memory management, such as health and longevity. *Id.*; EX1003, ¶293.

Second, A POSA would be motivated to incorporate the additional forms of SuperBlock metadata described in Son into the combined system. EX1003, ¶294. Son teaches the benefits of such metadata including for, *i.e.*, write operations and maintenance. *E.g.*, EX1009, [0078]-[0080]. A POSA would have had a reasonable expectation of success in incorporating the additional variants of SuperBlock metadata described in Son, *e.g.*, in the SuperBlock table relating to page utilization, to the set of metablock metadata already described in Gonzalez. EX1003, ¶294.

Third, a POSA would be motivated to combine Son's host functionality with the combined system. EX1003, ¶295. As described *supra*, Gonzalez teaches memory management responsibility may be shared between the controller and the host. A POSA would therefore consider Son's teaching regarding the shortcomings of SSDs which do not share responsibility between the host and controller and its teaching of the benefits of "incorporating both file system [(host)] and physical storage information [(controller)]" and "moving solid-state management functions above the firmware level." EX1009, [0012]-[0014]; EX1003, ¶295.

In light of the commonality in system design, flash memory organized into SuperBlocks or metablocks where metadata is used for memory management, and the identified benefits of shared memory management, a POSA would be motivated to incorporate the teaching of Son. EX1003, ¶296. A POSA would have had a reasonable expectation in moving (above the firmware) certain responsibility from SSD controller to host-side software, a common, straightforward and routine technique. *Id.*

C. Claim-by-claim Analysis

1. Claim 12

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶298-304.

As shown in Claims 1 and 2, the combination of Gonzalez, Goss and NVMe1.0b teaches *the SSD of Claim 2* and *logic for performance of the multi-plane write*.

A POSA would be motivated to implement, to Gonzalez's metablocks, the additional forms of metadata and categorization shown in Son, due to the utility described therein. EX1003, ¶¶301-304.

Specifically, a POSA would be motivated to combine Son's teaching of additional useful types of metadata that "associates each SuperBlock [] with free page count, valid page count, invalid page count," the "number of remaining free

pages” and “the current state of each SuperBlock” *metadata to indicate an amount of unerased host accessible space in the specified zone*. EX1009, [0065], [0039], [0067]. Son teaches that, after writing, the SuperBlock “state, valid page count, [and] invalid page count” are “update[d]” [*logic further operates to, upon the performance of the multi-plane write, update the metadata*]. *Id.*, [0050]-[0052].

2. Claim 13

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶305-307.

As shown in Claim 12, the combined system of Gonzalez, Goss, Son and NVMe1.0b teaches the SSD of claim 12, wherein the controller has *logic for performance of the multi-plane write, and to detect that the specified zone contains no erased host accessible space, e.g., the “number of remaining free pages.”*

A POSA would be further motivated to combine Son’s teaching of the classification of SuperBlocks, where the “possible *SuperBlock states include ‘SEALED,’ indicating that there are no free pages within the SuperBlock because all pages have been written*”—a POSA would understand that, following erasure, the SuperBlock is transitioned to a SEALED state (a FULL state). EX1009, [0040]; EX1003, ¶307.

3. Claim 15

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this

claim obvious. EX1003, ¶¶308-311.

As described in Claims 1 and 2, the combination of Gonzalez, Goss, and NVMe1.0b teaches *the SSD of claim 2 wherein the logic further operates to receive a maintenance command directed to the specified zone*. In the combined system, the maintenance operation, *e.g.*, garbage collection (or consolidation), includes erasing data from a source metablock and copying it to a target metablock [*in response to the maintenance command, erase host accessible space only within the specified zone*]. EX1007, [0051]; EX1008, [0044]; EX1009, [0070]; EX1003, ¶310.

A POSA would be further motivated to combine Son’s teaching of the classification of SuperBlocks, including possible SuperBlock states including “*CLEAN, meaning the SuperBlock has been erased and is available to take new writes*”—a POSA would understand that, following erasure, the SuperBlock is transitioned to a CLEAN state (an EMPTY state). EX1009, [0040]; EX1003, ¶311.

4. Claim 16

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶312-315.

As described in Claims 2 and 4, the combination of Gonzalez, Goss, and NVMe1.0b teaches the SSD of Claim 2 *with logic to compare the metadata with at least one threshold ... wherein making the notification available to the host is*

dependent on the comparison.

As described for Claims 1[e] and 4, Goss teaches that the controller “compar[es] the WI value to a predetermined threshold T,” where WI is “wear indication” and may represent the “total number of access operations (e.g., erasures).” EX 1008, [0074]-[0075].

In light of the existing functionality in the combined system to compare wear information to a threshold to trigger the host notification, it would be obvious to a POSA to incorporate Son’s teaching of “wear count” metadata—a fundamental proxy of wear “indicat[ing] how many times a particular SuperBlock has been erased”—and the utility of comparing it to a threshold. EX1009, [0039], [0079]-[0080]; EX1003, ¶315. A POSA would incorporate this metadata into the existing controller functionality where it would be compared to threshold, and, upon satisfaction, trigger a host notification. *Id.*

5. Claim 17

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶316-320.

As described in Claim 15, the combined system of Gonzalez, Goss, Son and NVMe1.0b, teaches ***the SSD of claim 2 wherein the logic further operates to: in response to a maintenance command from the host ... perform an erase of the erase units in the specified zone, and transition a state of the specified zone to an***

empty state whereby all host accessible pages of the specified zone are available for writing by the host. See, e.g., EX1009, [0040].

As described in Claim 1[b]-[c], [e], the combined system teaches that a command to a metablock may be directed to a “**starting address**” [a zone start logical block address]. When incorporating the teaching of Goss into the system of Gonzalez, a POSA would understand that the maintenance notification as applied to a metablock would include a start logical block address corresponding to the metablock [*in response to a maintenance command from the host directed to the zone start logical block address of the specified zone*]. EX1003, ¶320.

6. Claim 18

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶321-325. *See supra* Claim 17.

As described in Claim 1[e], the combined system of Gonzalez, Goss, and NVMe1.0b teaches a maintenance request from the controller to the host for performing maintenance on metablock [*the notification that recommends performing maintenance on the specified zone*]. A POSA would understand that the purpose of “real-time notifications to a host,” is to provide relevant metadata to the host such that the host may command maintenance [*wherein the maintenance command from the host is in response to the notification*]. EX1008, [0024], [0085]; EX1003, ¶323.

To the extent of the combined system of Gonzalez, Goss and NVMe1.0b, does not teach this limitation, a POSA would be motivated to incorporate the teaching of Son which describes the benefits of host-initiated maintenance commands. Son teaches that the host issues maintenance commands based “upon the occurrence of a[] condition.” *Id.*, [0063]; EX1003, ¶324. A POSA would understand that such a condition would be the “real-time notification[]” sent to the host in the combined system [*wherein the maintenance command from the host is in response to the notification*]. EX1003, ¶325.

7. Claim 19

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶326-328. *See supra* Claim 17.

As described for Claims 15 and 17, the combined system of Gonzalez, Goss, Son and NVMe1.0b teaches a maintenance operation *wherein the metadata is dependent on erase counts for erase units in the specified zone* where the *metadata is updated responsive to the performance of the erase*—transitioned to a state of empty.

Additionally, Son teaches that following host-initiated garbage collection, “update[] management data structures [(e.g., erase count metadata)] to reflect the erasure of the source SuperBlock.” EX1009, [0074], [0062]-[0082].

8. Claim 20

The combined system of Gonzalez, Goss, Son and NVMe1.0b renders this claim obvious. EX1003, ¶¶329-330.

As described for Claims 1 and 17, the combined system of Gonzalez, Goss, Son and NVMe1.0b teaches the SSD of claim 17 with logic for ***performing the erase of erase units in the specified zone***, where the zone is a metablock comprising blocks from multiple planes, which may be erased as a single unit as a result of the controller ***performing the erase on erase units of different planes in the specified zone***.

9. Claim 22

The combined system of Gonzalez, Goss, Son, and NVMe1.0b renders this claim obvious. EX1003, ¶¶331-336.

(a) 22[a]

As described for Claim 1[b]-[c], Gonzalez, Goss, and NVMe1.0b teaches a controller may receive a ***write command from the host*** where the SuperBlock includes a ***start logical block address within the range of logical block addresses of the specified SuperBlock***.

(b) 22[b]

As described for Claim 1[b]-[c], Gonzalez, Goss, and NVMe1.0b teaches a controller may receive a ***write command from the host*** where the controller

transfers data from the host to the specified zone at the assigned logical block address.

Gonzalez teaches that the controller may receive a host write command with “[d]ata with a given logical block address (LBA).” EX1007, [0047]. Upon receipt, that data “is mapped by the controller into a specific physical memory block within that plane that is identified by a physical block number (PBN),” through the determination of “a logical block number (LBN) and logical page,” ***assign a logical block address within the range of logical block addresses of the specified zone, transfer data from the host to the specified zone at the assigned logical block address.*** *Id.*, [0047], [0010]; EX1003, ¶¶334-335.

(c) 22[c]

A POSA would be motivated to incorporate the teaching from Son into the combination system regarding the benefits of sharing “***underlying physical***” mapping information with the host. EX1009, [0014]. A POSA would therefore understand that upon completion of the write command, in the combined system, the controller would transmit the underlying mapping information to the host including “a logical block number (LBN) and logical page” [***and return the assigned logical block address to the host***]. *Id.*, [0010], [0014]; EX1003, ¶326.

10. Claim 25

The combined system of Gonzalez, Goss, Son, and NVMe1.0b renders this claim obvious. EX1003, ¶¶337-341. *See supra* Claim 2.

(a) 25[a]

To the extent the combined system does not teach this limitation, a POSA would be motivated to incorporate the additional types of metadata described in Son, including “last modification time,” which is defined as “the most recent time that any page within the SuperBlock was ... programmed.” EX1009, [0042], [0080]; EX1003, ¶339.

(b) 25[b]

To the extent the combined system does not teach this limitation, a POSA would be motivated to incorporate Son’s teaching to compare the “last modification time” [***the tracked metadata***] against “a predetermined time period” [***to a threshold***]. EX1009, [0042], Claim 36. Son teaches that this comparison is useful for selecting SuperBlocks in a maintenance operation. *Id.* A POSA would incorporate this teaching into the combined system’s controller. EX1003, ¶¶340-341.

A POSA would also understand that this metadata comparison, similar to the threshold comparison taught in Goss, EX1008, [0075], and the erase count threshold comparison taught in Son, *e.g.*, Claim 16, in the combined system would result in ***making the notification available to the host dependent on the comparison.*** EX1003, ¶341.

11. Claim 26

The combined system of Gonzalez, Goss, Son, and NVMe1.0b renders this claim obvious. EX1003, ¶¶342-344.

As described for Claims 5 and 25, the combined system of Gonzalez, Goss, Son, and NVMe1.0b teaches ***the SSD of claim 25 wherein comparing the metadata with at least one threshold and making the notification available to the host is unsolicited by the host.***

12. Claim 27

The combined system of Gonzalez, Goss, Son, and NVMe1.0b renders this claim obvious. EX1003, ¶¶345-347. *See supra* Claim 2.

Son teaches that the host may “query” SuperBlock metadata, such as ***time period since data was written to the specified zone***, where the controller ***transmits to the host the requested information in response to the query command from the host.*** EX1009, [0042], [0047].

Goss teaches that the host may issue “a read request” for metadata which is “returned to the host responsive to” that command [***in response to a command from the host***]. EX1008, [0043]. A POSA would understand that in the combined system this read request (which ***transmits to the host information***) would include metadata ***dependent on a time period since data was written to the specified zone.*** EX1003, ¶347. *See supra* Claims 1, 25.

XIII. GROUND VI - CLAIMS 28-30 ARE RENDERED OBVIOUS BY GONZALEZ IN VIEW OF GOSS, SON, NVME1.0B, AND CORNWELL

A. References

1. Cornwell (EX1012)

U.S. Pat. No. 2012/0072807 (“Cornwell”) was filed on November 28, 2011, and was published March 22, 2012. It is prior art under AIA 35 U.S.C. § 102.

Cornwell describes a memory system comprising a “nonvolatile memory” “a memory 111, a memory controller,” and an external host, “such as a computer.” EX1012, [0004], [0020]; EX1003, ¶74.

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Cornwell with the combined system of Gonzalez, Goss, Son and NVMe1.0b, specifically, to incorporate Cornwell’s teaching regarding sharing certain metadata with the host to inform host commands and operations. EX1003, ¶¶348-351.

First, Cornwell, like Gonzalez, Goss and Son, involves similar techniques to solve the same problem, *i.e.*, collecting, storing and using metadata for maintenance activities to improve the efficiency and performance of NAND flash memory systems. *E.g.*, EX1012, [0004]-[0014], [0021]. Further, like Gonzalez and Son, Cornwell teaches the organization of metadata in groupings of physical blocks, termed “virtual blocks.” *Id.*, 4:35-39, 9:17-40; EX1003, ¶349.

Second, Goss, Son, NVMe1.0b, and the combined system already include functionality enabling the controller to share certain information with the host, including metadata and related notifications. A POSA would have a reasonable likelihood of success in incorporating Cornwell's disclosure of sharing additional types of metadata with the host. EX1003, ¶350.

Such a modification would have been straightforward and routine. *Id.* A POSA would be motivated to do so due given the utility of sharing such additional metadata with the host as described in Cornwell. EX1012, [0006]-[0008], [0043].

C. Claim-by-claim Analysis

1. Claim 28

The combined system of Gonzalez, Goss, Son, Cornwell, and NVMe1.0b renders this claim obvious. EX1003, ¶¶352-357. *See supra* Claim 2.

As described for, *e.g.*, Claims 13 and 15, the combined system of Gonzalez, Goss, Son and NVMe1.0b teaches metadata describing ***a state of the specified zone***. A POSA would understand that the notification to the host in the combined system would include information about the state of the zone subject to the maintenance request. EX1003, ¶353.

To the extent the combined system of Gonzalez, Goss, and NVMe1.0b does not teach this limitation, a POSA would be motivated to incorporate the teaching of Cornwell into the combined system. EX1003, ¶¶354-355. Cornwell describes

functionality where “[m]etadata may be transmitted from a memory of a host [(a controller)] to an external host [(a host)],” “includ[ing] bad block data, where bad block data includes a listing of blocks in the memory that do not satisfy performance criteria.” EX1012, [0004], [0006]. A POSA would understand that to generate bad block metadata the controller must contain ***logic that operates to detect a failure condition of one of the erase units***, and that, when incorporated into the combined system, the failure would correspond to a metablock [***detect a failure condition of one of the erase units of the specified zone***].

Additionally, Cornwell describes transmitting information to the host including “mapping metadata” corresponding to “virtual blocks,” groupings of blocks. EX1012, [0045]. Based on this teaching and the existing teaching of the combined system, *supra* Claim 20[c], the metadata ***transmitted to the host*** would include the ***zone start logical block address***. EX1003, ¶356.

To the extent the combination of Gonzalez, Goss, Cornwell, and NVMe1.0b does not teach this claim, a POSA would be motivated to incorporate the teaching of the NVMe1.0b, *see supra* Claims 1, 27. EX1003, ¶357.

2. Claim 29

The combined system of Gonzalez, Goss, Son, Cornwell, and NVMe1.0b renders this claim obvious. EX1003, ¶¶358-359. *See supra* Claim 2.

As described in Claim 28, the combined system of Gonzalez, Goss, Son and NVMe1.0b teaches ***wherein the logic further operates to detect a failure condition of one of the erase units of the specified zone.***

Cornwell teaches that a failure condition may be identified in connection with a write command. EX1012, [0048]. A POSA would be motivated to incorporate this teaching to make full use of the functionality described in Cornwell and would understand that, to do so in the combined system, the failure information would correspond to the range of LBAs in the metablock subject to the write request [***in response to a write command from the host within the range of logical block addresses of the specified zone, transmit to the host a notification of a write failure***]. EX1003, ¶359.

3. Claim 30

The combined system of Gonzalez, Goss, Son, Cornwell, and NVMe1.0b renders this claim obvious. EX1003, ¶¶360-361. *See supra* Claim 2.

As described for Claims 28 and 29, the combined system of Gonzalez, Goss, Son, Cornwell, and NVMe1.0b teaches sending metadata showing metablock failure conditions to the host. Cornwell teaches that the failure condition may be based on “blocks in the memory that do not satisfy performance criteria.” EX1012, [0006]. A POSA would understand that such a performance criterion would include read

errors—*wherein the metadata is dependent on read errors detected for the specified zone.* *Id.*, [0049]. EX1003, ¶361.

XIV. MANDATORY NOTICES AND FEES

A. Real Party-In-Interest

The real party-in-interest for Petitioner is Samsung.

B. Related Matters

Radian Memory Systems LLC, v. Samsung Electronics Co., Ltd., AND Samsung Electronics America, Inc., Case No. 2:24-cv-1073 (E.D. Tex.).

C. Counsel and Service Information

Petitioner provides the following counsel and service information. Petitioner consents to electronic service the email addresses listed in the table below. Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this Petition.

LEAD COUNSEL	BACK-UP COUNSEL
James M. Glass (Reg. No. 46,729) jimglass@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 295 Fifth Avenue, 9th Floor New York, NY 10016 Tel: (212) 849-7000 Fax: (212) 849-7100	John T. Mckee (Reg. No. 65,926) johnmckee@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 295 5th Avenue, 9th Floor New York, New York 10016 Tel: (212) 849-7000 Fax: (212) 849-7100 Greg Miraglia (Reg. No. 74,641) gregmiraglia@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP

	300 West 6th St, Suite 2010 Austin, TX 78701 Tel: (737) 667-6100 Fax: (737) 667-6110
--	---

D. Payment of Fees

The undersigned authorizes the Office to charge the fee required for this Petition for *Inter Partes* Review to Deposit Account No. 50-5708. Any additional fees that might be due are also authorized.

XV. CONCLUSION

For these reasons, Petitioner requests *inter partes* review of the '772 Patent.

Date: July 17, 2025

By: / *John McKee*

John McKee (Reg. No. 65,926)
johnmckee@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN,
LLP
295 Fifth Ave., 9th Floor
New York, NY 10016
Tel: (212) 849-7000
Fax: (212) 849-7100

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,878. Pursuant to 37 C.F.R. §42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Date: July 17, 2025

Respectfully submitted,

By: /s/ John McKee

Attorney for Petitioners

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e), 42.105(a), the undersigned hereby certifies service on the Patent Owner of a copy of this Petition and its respective exhibits at the official correspondence address for the attorneys of record for the '772 patent as shown in USPTO P-TACTS via FedEx:

Radian Memory Systems, LLC
6275 West Plano Parkway
Suite 500
Plano, TX 75093

Courtesy copies were also sent via electronic mail to Patent Owner's counsel of record in the related district court proceeding:

Andrea Fair

andrea@millerfairhenry.com
radian@caldwellcc.com

DATED: July 17, 2025

Respectfully submitted,

By /s/ John McKee

John McKee (Reg. No. 65,926)
johnmckee@quinnemanuel.com

QUINN EMANUEL URQUHART &
SULLIVAN, LLP
295 Fifth Ave., 9th Floor
New York, NY 10016
Tel: (212) 849-7000
Fax: (212) 849-7100

Attorney for Petitioners