

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 IPR2025-01266

U.S. Patent No. 11,544,183

**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 '183 PATENT	3
A.	Priority	3
B.	Prosecution History	3
C.	Alleged Invention	5
V.	FOUNDATIONS	5
VI.	LEVEL OF ORDINARY SKILL	6
VII.	CLAIM CONSTRUCTION	6
VIII.	GROUND I: CLAIMS 1, 2, 8, 10-14, 18, 20 ARE RENDERED OBVIOUS BY ELLIS IN VIEW OF SON	6
A.	Overview of the Prior Art.....	6
1.	ELLIS (EX1004).....	6
2.	SON (EX1005).....	7
B.	Motivation to Combine	9
C.	Claim-by-Claim Analysis.....	12
1.	Claim 1	12
(a)	1[pre]: A memory controller to control flash memory, the flash memory having subdivisions, each of the subdivisions corresponding to a respective group of erase blocks, each of the erase blocks such that it must be erased before storage locations in that erase block can be reprogrammed, the memory controller comprising:	12
(b)	1[a]: a host interface;	13
(c)	1[b]: a memory interface;	13
(d)	1[c]: logic to track information indicating extent of page utilization of each of the subdivisions;	13

(e)	1[d]: logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions, wherein the logic to track is to, in association with execution of the write request, update the information indicating extent of page utilization of the specific one of the subdivisions;	14
(f)	1[e]: logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions;	16
(g)	1[f]: logic to receive an erasure request from the host, via the host interface, the erasure request accompanied by the address corresponding to the specific one of the subdivisions, to responsively control erasure of each unerased erase block in the group respective to the specific one of the subdivisions, and to update the information indicating extent of page utilization of the specific one of the subdivisions; and	18
(h)	1[g]: logic to, in connection with the erasure, detect a defect in one of the erase blocks in the group respective to the specific one of the subdivisions, to responsively substitute one or more different erase blocks of the flash memory for the one of the erase blocks for which the defect was detected.....	20
(i)	1[h]: wherein each said logic comprises at least one of hardware circuitry or instructions stored on non-transitory, machine-readable media that when executed are to control the operation of hardware circuitry.	21
2.	Claim 2.....	22
(a)	2[a]: The memory controller of claim 1, wherein the host interface is also to receive a data copy request from the host, ...	22

(b)	2[b]: wherein the memory controller is to responsively retrieve data from a source location in the flash memory, the source location associated with the specific one of the subdivisions, and write the retrieved data to a destination location in the flash memory, without sending the retrieved data to the host,.....	23
(c)	2[c] and wherein the logic to track is to responsively update the tracked information for one of the subdivisions which corresponds to the destination location, in association with the write of the retrieved data into the destination location.	24
3.	Claim 8: The memory controller of claim 1, wherein the memory controller further comprises logic to receive a query from the host, and wherein the logic to send the information to the host is to do so as a response to the query.....	25
4.	Claim 10.....	26
(a)	10[a] The memory controller of claim 1, wherein the memory controller further comprises logic to store erase count information for each of the subdivisions, ...	26
(b)	10[b] and to send to the host, via the host interface, the erase count information for the specific one of the subdivisions, together with the address corresponding to the specific one of the subdivisions, ...	27
(c)	10[c] and wherein the logic to store the erase count information is to update the erase count information for the specific one of the subdivisions in association with the erasure.....	28
5.	Claim 11: The memory controller of claim 1, wherein the extent of the page utilization of the specific one of the subdivisions represents the amount of valid data written to the specific one of the subdivisions since the specific one of the subdivisions was last erased.	28

6.	Claim 12: The memory controller of claim 1, wherein the extent of the page utilization of the specific one of the subdivisions represents a remaining space of the specific one of the subdivisions that is available for writing without first erasing the specific one of the subdivisions.....	29
7.	Claims 13, 14, 18 and 20	30
IX.	GROUND II: CLAIMS 6, 9, 17, 19 ARE RENDERED OBVIOUS BY ELLIS IN VIEW OF SON AND YAMADA	31
A.	Overview of the Prior Art.....	31
1.	Yamada (EX1006)	31
B.	Motivation to Combine	32
C.	Claim-by-Claim Analysis.....	34
1.	Claim 6.....	34
(a)	6[a] The memory controller of claim 2, the memory controller further comprises logic to track metadata for each of the subdivisions, the metadata indicating a time since data was programmed,.....	34
(b)	6[b] to compare the tracked metadata to a threshold, ...	35
(c)	6[c] and, responsive to a determination that the tracked metadata for the specific one of the subdivisions satisfies the threshold, to transmit to the host the address corresponding to the specific one of the subdivisions; and the data copy request is accompanied by the address corresponding to the specific one of the subdivisions.....	36
2.	Claim 9.....	37
(a)	9[a] The memory controller of claim 1, wherein: the memory controller further comprises logic to assign a write destination in the flash memory, in which data accompanying the write request is to be stored, the write destination corresponding to the specific one of the subdivisions,.....	37

(b)	9[b] and to transmit to the host a read address that will be used to retrieve the data specifically from the write destination;	38
(c)	9[c] and the memory controller further comprises logic to receive, from the host, and via the host interface, a read request for the data, accompanied by the read address, and to identify both of the specific one of the subdivisions and the write destination from the read address.	39
3.	Claims 17 and 19.....	41
X.	GROUND III: CLAIMS 1, 2, 8, 10-14, 18, 20 ARE RENDERED OBVIOUS BY SINCLAIR IN VIEW OF YU AND SON	41
A.	Overview of the Prior Art.....	41
1.	Sinclair (EX1007)	41
2.	Yu (EX1008).....	42
B.	Motivation to Combine	43
C.	Claim-by-Claim Analysis.....	46
1.	Claim 1	46
(a)	1[pre].....	46
(b)	1[a].....	46
(c)	1[b].....	46
(d)	1[c]	47
(e)	1[d].....	47
(f)	1[e]	48
(g)	1[f]	50
(h)	1[g].....	51
(i)	1[h].....	52
2.	Claim 2	52
(a)	2[a]	52
(b)	2[b].....	53
(c)	2[c].....	54

3.	Claim 8.....	55
4.	Claim 10:.....	55
	(a) 10[a].....	55
	(b) 10[b].....	56
	(c) 10[c].....	57
5.	Claim 11	57
6.	Claim 12.....	58
7.	Claims 13, 14, 18 and 20	59
XI.	GROUND IV: CLAIMS 6, 9, 17, 19 ARE RENDERED OBVIOUS BY SINCLAIR IN VIEW OF YU, SON AND JUNG.....	59
A.	Overview of the Prior Art.....	59
1.	Jung (EX1010).....	59
B.	Motivation to Combine	60
C.	Claim-by-Claim Analysis.....	62
1.	Claim 6.....	62
	(a) 6[a].....	62
	(b) 6[b].....	63
	(c) 6[c].....	64
2.	Claim 9.....	66
	(a) 9[a].....	66
	(b) 9[b].....	67
	(c) 9[c].....	68
3.	Claims 17 and 19.....	69
XII.	MANDATORY NOTICES AND FEES	69
A.	Real Party-In-Interest	69
B.	Related Matters.....	69
C.	Counsel and Service Information.....	69
D.	Payment of Fees	70
XIII.	CONCLUSION.....	70

EXHIBIT LIST

Ex.	Description
1001	U.S. Patent No. 11,544,183 (“’183 Pat.)
1002	Prosecution History of U.S. Patent No. 11,544,183
1003	Expert Declaration of Dr. Heiner Litz
1004	U.S. Patent 9,239,781 (“Ellis”)
1005	U.S. Patent Application No. 2012/0096217 (“Son”)
1006	U.S. Pat. Pub. No. 2011/0271032 (“Yamada”)
1007	U.S. Pat. No. 7,739,444 (“Sinclair”)
1008	U.S. Pat. Pub. 2012/0284587 (“Yu”)
1009	U.S. Pat. No. 7,409,489 (“Sinclair II”)
1010	U.S. Pat. Pub. No. 2008/0189485 (“Jung”)
1011	Nick McKeown et al, OpenFlow: Enabling Innovation in Campus Networks

I. INTRODUCTION

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc., (collectively, “Samsung” or “Petitioner”) respectfully requests *inter partes* review of claims 1, 2, 6, 8-14 and 17-20 of U.S. Patent No. 11,544,183 (the “’183 patent”).

The ’183 patent relates to systems and methods for a flash memory controller where memory is organized into units termed subdivisions, metadata including page utilization and erase count is stored for the subdivisions, and metadata is provided to the host for use in memory management.

The ’183 patent purports to describe novel “techniques for cooperative interaction between a memory controller and host.” Shared responsibility for memory management was known in the art prior to the effective filing date of the ’183 patent. Indeed, the prior art in this Petition teaches cooperative interaction between a memory controller and a host, where metadata is shared with the host for purposes of memory management.

During prosecution of the ’183 patent, its purported novelty was the ability to maintain page utilization metadata for subdivisions, which are defined as groups of multiple blocks. This too was exceedingly well-known at the time, as illustrated by the prior art in this Petition.

A reasonable likelihood exists that Petitioner will prevail on at least one challenged claim.

II. STANDING

Petitioner certifies the '183 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 or blocks. EX1001, 1:30-47. Blocks are comprised of pages. *Id.* Pages represent the smallest unit that may be written. *Id.* To write the pages of a block with new data, the block must first be erased. EX1009, 1:65-2:4; EX1003, ¶29. A block may only be erased a certain number of times prior to inoperability. EX1008, [0009].

To increase the efficiency of flash memory, SSDs may employ a larger unit of memory. For example, SSDs may be organized into groupings of blocks operated as a single unit, termed, *e.g.*, “superblocks,” EX1004, 6:43-65, or “metablocks,” EX1007, 3:43-46.

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. EX1005, [0062]-[0068]; EX1003, ¶¶34-35. To identify data for maintenance, metadata is tracked describing the history of the units of memory. EX1005, [0084]; EX1006, [0081].

The memory management functions may be carried out by the controller (internal to the SSD) or via shared responsibility with the host (external to the SSD).

EX1005, [0014]; EX1006, [0012]-[0035]; EX1001, 1:48-52; EX1003, ¶¶38-40. In the art, share responsibility may be referred to as “a cooperative memory management system.” EX1010, [0011]-[0012].

IV. OVERVIEW OF THE '183 PATENT

A. Priority

The '183 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 '183 Patent was filed on August 19, 2020.

The examiner issued a non-final rejection on October 1, 2021. EX1002, 425-458. The examiner rejected all claims (1-20) under 35 U.S.C. 102(b) as anticipated by Selinger. Applicant amended the claims and asserted that Selinger did not teach certain limitations, such as those directed to “erasure” and “remapping” defective blocks. *Id.*, 401-424.

The examiner issued a non-final rejection on June 28, 2022. EX1002, 129-148. The examiner rejected claims (1-4, 11-14, 23-25, and 29-31) under 35 U.S.C. 103(a) as obvious in light of Selinger in view of Gorobets. The examiner rejected

the remaining claims (21-22, and 27-28) as obvious in light of Selinger in view of Gorobets and Allen. *Id.*

In response, Applicant submitted an Amendment, relying on an Examiner Interview. Applicant stated that “the features of independent claim 1 and how those features differ from the disclosures contained in the cited Selinger and Gorobets documents were discussed” in that interview. EX1002, 122. Applicant focused on a single limitation and “asserted that the cited documents fail to disclose that information indicating an extent to which each of the subdivisions is full is tracked.” *Id.* Applicant argued that the art did not track page utilization for each of the blocks in a subdivision: “Selinger fails to disclose ‘logic to track information indicating an extent to which each of the erase blocks is full.’” *Id.*, 106. The examiner “agreed that the cited documents do not appear to disclose this feature,” and “[i]t was proposed that [the claim] language be changed to ‘logic to track information indicating the extent of page utilization of each of the Subdivisions[.]’” *Id.*, 122. The Applicant amended the claim language, *i.e.*, “logic to track information indicating extent of page utilization of an extent to which each of the subdivisions is full.” *E.g.*, *id.*, 112.

Two weeks later, on September 9, 2022, the examiner issued a Notice of Allowance. *Id.*, 26. The ’183 Patent issued January 3, 2023.

C. Alleged Invention

The '183 patent describes a memory system, including a memory controller (in the SSD) and a host, where responsibility for memory management is shared between the controller and host. EX1001, 4:46-51.

The memory is organized by “subdivisions,” where “each subdivision is an EU [erase unit] or page of NAND flash memory.” *Id.*, 7:46-51.

Memory management functions are based on “stored information” which describes “subdivisions of memory.” *Id.*, 4:46-65, 16:54-61. This information includes the “status” of data, including “page utilization.” *Id.* The '183 patent defines page utilization—“‘page utilization’ as used herein generally refers to the extent to which pages of a given EU are valid (in use) or are stale or not in use,” *id.*, 16:54-61, or may represent SSD-wide or “global page utilization.” *Id.*, 23:49-53.

The '183 Patent teaches that this data describing subdivisions may be provided by the controller to the host. *Id.*, 32:5-50. This information is used for maintenance operations such as, e.g., “garbage collection.” *Id.*

V. GROUNDS

Ground	Basis	Reference(s)	Claims
I	§ 103	Ellis, Son	1, 2, 8, 10-14, 18, 20
II	§ 103	Ellis, Son, Yamada	6, 9, 17, 19
III	§ 103	Sinclair, Yu, Son	1, 2, 8, 10-14, 18, 20
IV	§ 103	Sinclair, Yu, Son, Jung	6, 9, 17, 19

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSA”) at the time of the ’183 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, ¶43. 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 ’183 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.

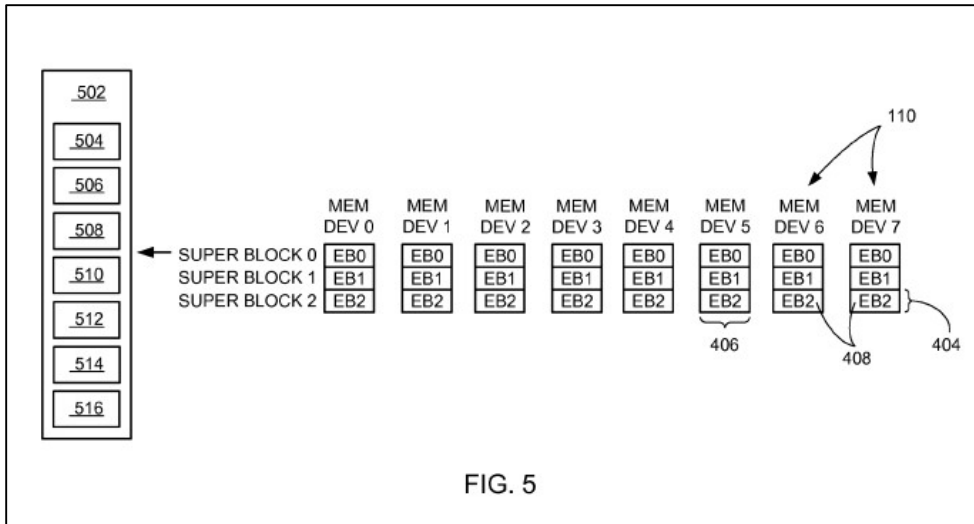
VIII. GROUND I: CLAIMS 1, 2, 8, 10-14, 18, 20 ARE RENDERED OBVIOUS BY ELLIS IN VIEW OF SON

A. Overview of the Prior Art

1. ELLIS (EX1004)

U.S. Patent 9,239,781 (“Ellis”) was filed October 10, 2012 and is prior art under at least pre-AIA 35 U.S.C. § 102(e).

Ellis teaches a NAND storage control system sub-divided into superblocks—each superblock is a group of erase blocks. EX1004, 6:43-65. Figure 5 illustrates superblocks which comprise one erase block from each channel.



Id., Fig. 5.

Ellis teaches storing metadata for each superblock including, *e.g.*, “valid-page count,” defined as the “***number of pages that are valid in each of the super blocks 502.***” *Id.*, 11:58-12:2, 12:37-41, Claims 1, 5.

Ellis teaches the memory management is done at the superblock-level: “host writes, recycling operations, or erasing of blocks can be performed on ... one of the super blocks ***as a single unit.***” *Id.*, 8:2-7. Ellis also teaches that memory management may be shared: either “the memory controller [or] the host system ... can operate any of the modules and functions of the memory controller.” *Id.*, 4:58-65.

2. SON (EX1005)

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 pre-AIA 35 U.S.C. §§ 102(a) and 102(e).

Son teaches an SSD management system which relies on sharing of file system (host) information and controller information to improve performance. EX1005. Son describes the “shortcomings of conventional systems,” and describes the benefits of shared memory management where the host “incorporates both file system information and information related to the underlying physical solid-state storage media into its operations.” *Id.*, [0014]. Shared memory management “may achieve various advantages ... such as enhanced I/O performance, simplified SSD firmware, and extended SSD lifespan.” *Id.*

Son’s system is subdivided into superblocks which are “a grouping of ... blocks.” *Id.*, [0029]. Son teaches the memory management is done at the superblock level, *e.g.*, write and erasure commands using a “superblock address.” *Id.*, [0029], [0038]. Son also teaches storing metadata for the superblocks, in a “Superblock Table.”

Superblock Table ³⁵⁰										
	335a Super Block ID	335b Wear Count	335c # Valid Pages	335d # Invalid Pages	335e State	335f Next Superpage	335g Next Page	335h Page Detail	335j Last Mod. Time	335k Data Type
340a	0	115	0	0	CLEAN	0	0	Pagelist ₁ []	5/5/11 09:11:01	Metadata
340b	1	462	856	741	IN_USE	68	5	Pagelist ₂ []	1/1/09 22:08:11	User Data
	...	...	...	...	...	...	...	...	...	...

FIG. 3C

Id., Fig. 3C, [0080], [0029], [0038]. The metadata includes, *e.g.*, “a wear count [], *a valid page count* [], an invalid page count [], ... a next page indicator [], ... [and] a last modification time[.]” *Id.*, [0039].

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Ellis and Son. 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. *See, e.g.*, EX1003, ¶¶63-75.

First, Ellis *explicitly cites to and references* Son. EX1004, Pg. 3.

Second, Son and Ellis both concern the optimization of NAND Flash SSD memory management. EX1003, ¶¶64-66; EX1004, 7:32-45; EX1005 [0002], [0014]. Ellis and Son are in the same field of endeavor.

Third, both Ellis and Son address the same problem—simplifying systems, improving efficiency and performance, and prolonging life-span—through the use of larger units of memory, superblocks. EX1005, [0014]; EX1004, 21:21-35, 7:32-45, 10:66-11:7; EX1003, ¶¶64, 67, 70. And both systems use similar techniques to solve the same problem, *e.g.*, rely on superblock level metadata to optimize memory management, such as health and longevity. EX1004, 11:58-12:23; EX1005, [0078]-[0080]; EX1003, ¶¶67-69.

A POSA would be motivated to incorporate the additional forms of superblock metadata described in Son into the system described in Ellis. EX1003, ¶¶73-74. Ellis teaches that “metadata can include ... [any] NAND statistics that [are] collected as a consequence of normal operatives,” and “any additional information ... that can aid in retrieval, organization, health of, and current condition or state of the memory devices,” EX1004, 10:21-32, 12:26-29. Son teaches the benefits of such metadata including for, *e.g.*, write operations and maintenance. *E.g.*, EX1005, [0078]-[0080]. A POSA would have had a reasonable expectation of success in combining Son and Ellis as it merely represents adding the additional variants of superblock metadata describing in Son, in *e.g.*, the Superblock table relating to page utilization, to the set of superblock metadata already described in Ellis. EX1003, ¶74.

Fourth, in light of variable host and controller responsibility described in Ellis and Son, a POSA would be motivated to combine Son’s host functionality with the system described in Ellis. EX1003, ¶¶70-72, 75. Ellis teaches the memory management responsibility may be shared between the controller and the host. EX1004, 4:58-65, 20:40-44. A POSA would therefore consider Son’s teaching regarding the shortcomings of SSDs which do not share responsibility between the host and controller, and teaches the benefits of “incorporating both file system [(host)] and physical storage information [(controller)]” and “moving solid-state

management functions above the firmware level.” EX1005, [0012]-[0014]; EX1003, ¶76.

In light of the commonality in system design, flash memory organized into superblocks where superblock metadata is used for memory management, and the identified benefits of shared memory management, a POSA would be motivated to combine Ellis and Son. A POSA would have had a reasonable expectation in moving (above the firmware) certain responsibility from SSD controller to host-side software. This exercise, moving functionality in device firmware to the host-side software is common, straightforward and routine. EX1003, ¶80. Indeed, such translocation for cooperative memory management had been well-described in the art. For example, Samsung’s U.S. Pat. Pub. 2008/0189485A1 describes “method[s] of *cooperative memory management* where “at least one of several memory management functions [is] performed by the host.” EX1010, Abstract.

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

C. Claim-by-Claim Analysis

1. Claim 1

- (a) 1[pre]: A memory controller to control flash memory, the flash memory having subdivisions, each of the subdivisions corresponding to a respective group of erase blocks, each of the erase blocks such that it must be erased before storage locations in that erase block can be reprogrammed, the memory controller comprising:

To the extent the preamble is limiting, it is disclosed by Ellis. Ellis teaches that: “[t]he *memory controller* 104 provides data control and management[,] interfaces with the host system 108 and *controls the memory array*.” EX1004, 3:17-21. Ellis teaches the “memory array” is controlled by the controller is “*flash memory*.” *Id.*, 3:22-24.

Ellis teaches that flash memory is divided into “superblocks” [*subdivisions*], that are “defined as logical groupings of the erase blocks,” which means that *each subdivision corresponds to a respective group of erase blocks*. *Id.*, 7:32-39, 6:43-65, 3:10-45.

Ellis teaches that each block in the superblock is an “erase block ... of a NAND flash memory” and “represent[s] the smallest unit of memory that can be erased.” *Id.*, 6:50-67. A NAND erase block must be erased prior to reprogramming of the individual pages within the block. EX1003, ¶76; EX1001, 30:58-60. Thus

Ellis's NAND erase blocks “*must be erased before storage locations in that erase block can be reprogrammed.*”

Ellis renders the preamble obvious. EX1003, ¶¶76-77.

(b) 1[a]: a host interface;

Ellis teaches that the “memory controller 104 can include a control unit 202, a storage unit 204, a memory interface unit 206, and *a host interface unit* 208. EX1004, 3:38-45; EX1003, ¶¶78-79.

(c) 1[b]: a memory interface;

Ellis teaches that the “memory controller 104 can include a control unit 202, a storage unit 204, *a memory interface unit* 206, and a host interface unit 208. EX1004, 3:38-45; EX1003, ¶¶80-81.

(d) 1[c]: logic to track information indicating extent of page utilization of each of the subdivisions;

Ellis teaches that the controller may contain “*logic*” and “software” in order to perform its functions. EX1004, 3:38-50. The controller has logic for tracking superblock metadata including the “*valid-page count*,” representing the “number of pages that are valid in each of the super blocks 502,” [*information indicating extent of page utilization of each of the subdivisions*]. *Id.*, 11:58-12:2, 18:54-56, 19:8-12. This metadata is “tracked,” because the controller “updat[es]” the metadata values, including “valid-page count” after “performing an operation on the superblock.” *Id.*, Claims 1, 5, 20:11-20.

To the extent Ellis does not expressly teach this limitation, a POSA would be motivated to combine its teaching with Son which includes additional useful types of metadata in “a superblock table [that] *associates each superblock [] with free page count, valid page count*, invalid page count” and “a next page indicator.” EX1005, [0065], [0039]. A POSA could add these additional types of metadata in a typical and straightforward manner, EX1003, ¶¶82-85.

The combined system of Ellis and Son thus renders this limitation obvious.

Id.

- (e) **1[d]: logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions, wherein the logic to track is to, in association with execution of the write request, update the information indicating extent of page utilization of the specific one of the subdivisions;**

Ellis teaches that the controller contains logic “for a write operation ... for host write data,” EX1004, 14:25-34, where the “host interface” is the “communication mechanism” for the “host system,” *id.*, 4:46-50. Ellis thus teaches a controller with “*logic to receive a write request from a host via the host interface.*”

Ellis teaches that write request may designate “a set of the erase blocks [] as a single unit.” EX1004, 7:64-8:7. Ellis teaches the use of “super block identification (ID) values.” *Id.*, 10:27-32. In order to perform a “host write” directed to a superblock, the write address necessarily must

identify or reference a superblock ID value, [*“accompanied by an address that designates a specific one of the subdivisions”*]. EX1003, ¶¶87-88.

After writing, Ellis teaches the controller has logic to *“updat[e] metadata”* such as *“valid-page count in ... the super blocks.”* *Id.*, Claims 1, 12, 20:11-20.

To the extent Ellis does not teach this limitation, a POSA would be motivated to combine its teaching with Son. Son teaches that write requests include an assigned file system logical block address (“FLBA”) which is associated with a “superblock *address.*” EX1005, [0023], [0029], [0038]. Son thus teaches a *“write request”* accompanied by an FLBA *designating* a superblock address. After writing, the superblock *“valid page count, invalid page count, [and] next page,” are “update[d].”* *Id.*, [0050]-[0052]. Note, in Ellis, the superblock metadata is updated by the controller, EX1004, Claims 1, 12, therefore, “[i]n the combined system the additional superblock metadata described in Son would also be updated by the controller.” EX1003, ¶91.

The combined system of Ellis and Son thus renders this limitation obvious. EX1003, ¶¶87-92.

- (f) **1[e]: logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions;**

Ellis teaches that the controller has *logic to send “control and status information generated by the memory controller 104 to the host system 108.”*

EX1004, 4:51-53. Ellis therefore teaches “logic to send the host [] information.”

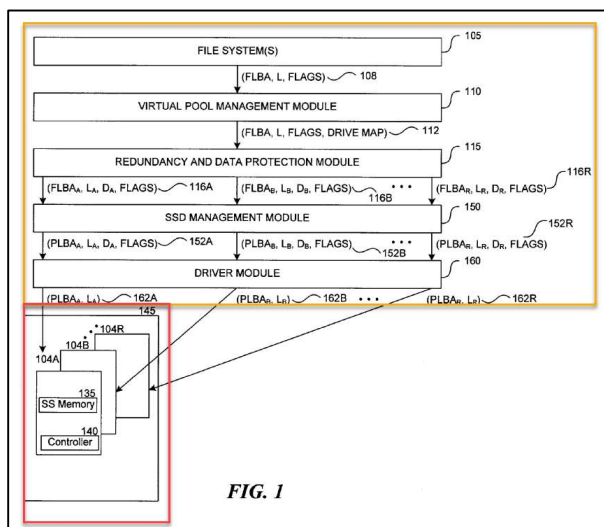
Ellis further teaches that the control and status information is stored as metadata, *id.*, 12:34-41 (“metadata ... used for storing the health and use status of the super blocks 502”), and includes “NAND statistics [] collected as a consequence of normal operations,” *id.*, 10:27-32, 12:26-29. The “NAND statistics” include “*valid page count*” [“page utilization”]. *Id.*, 11:58-62. Ellis thus teaches sending “*valid-page count*,”—“the information indicating extent of page utilization of the specific one of the subdivisions”—as “control and status *information ... to the host system.*” EX1003, ¶93 (“A POSA would understand that ‘control and status information,’ *i.e.*, the state of the units of memory in the SSD, would include, the ‘valid-page count’ for metablocks.”). Indeed, Ellis describes instances where the controller sends the host “metadata” showing the “valid state of data stored within the one of the superblocks.” EX1004, 12:34-37.

Ellis also teaches that the stored “control and status” metadata sent to the host includes “logical block address (LBA) information, [and] super block identification (ID) values,” which are *addresses corresponding to the specific one of the*

subdivisions as discussed above, and “logical to physical mapping of data.” *id.*, 10:27-32, 12:26-41. For example, Ellis describes instances where the controller sends “metadata pages” to the host including address “mapping data.” *Id.*

To the extent Ellis does not teach this limitation, in light of Ellis’s disclosure that memory management may be shared between the controller and host, EX1004, 4:58-65, a POSA would incorporate the teaching of Son which describes the benefit of sharing information with the host in order to perform memory management functions, EX1005, [0013]-[0014].

Son teaches the SSD is managed by a host which is comprised of “an SSD management module (‘the SSD module’),” *id.*, [0016] as shown in annotated Figure 1 (yellow box is host, red box is SSD).



EX1003, ¶49. Son teaches that the host may acquire superblock metadata stored in flash memory “*by querying ... the controller.*” EX1005, [0047]. *See id.*, [0034],

Figs. 3A-D. Thus, Son teaches that the controller, upon receipt of the query, has “logic to send the host [] information.” The host may query page utilization (“*the number of free, valid and/or invalid pages in each candidate superblock*”) for each superblock, and the “*superblock address*.” EX1005, [0065], [0039], [0047]. The host uses this information in order to perform operations such as “garbage collection or wear leveling.” *Id.*, [0017]. Son therefore teaches this limitation.

The combined system of Ellis and Son thus renders this limitation obvious. EX1003, ¶¶93-97.

- (g) **1[f]: logic to receive an erasure request from the host, via the host interface, the erasure request accompanied by the address corresponding to the specific one of the subdivisions, to responsively control erasure of each unerased erase block in the group respective to the specific one of the subdivisions, and to update the information indicating extent of page utilization of the specific one of the subdivisions; and**

Ellis teaches that the host or controller may request “*erasing* of blocks ... within one of the *super blocks 502 as a single unit*,” EX1004, 8:2-7, 4:58-65. Ellis teaches host erase requests may be sent, via the “*host interface*,” to the “*memory controller*,” which has “*logic*” to process those commands. *Id.*, 4:46-50, 3:38-45, 20:40-44.

Ellis teaches that “groups of the erase blocks 408 can be operated as one virtual erase block,” where erasure is completed on a “super block[] 502 as a single unit.” EX1004, 7:64-8:7. A POSA would understand that, for erasures to be carried

out on a superblock as a single unit, the erasure request must be *accompanied by* an address corresponding to “a superblock ID value” which, as discussed above, represents “*the address corresponding to the specific one of the subdivisions.*” EX1003, ¶32.

Ellis teaches that in response to the erase request, “*all the erase blocks 408 in each of the super blocks 502 can be erased* at the same time.” *Id.*, 17:31-36. A POSA would understand that this disclosure teaches erasing “each unerased erase block in the group respective to the specific one of the subdivisions,” specified by the superblock ID value.

Following erasure, Ellis describes that “*the valid-page count ... [is] updated.*” *Id.*, 20:11-20, Claims 1, 5. Thus Ellis teaches “and to update the information indicating extent of page utilization of the specific one of the subdivisions.”

To the extent Ellis does not teach this limitation, a POSA would look to Son’s disclosure which provides additional implementation details for host initiated erase commands, where “*the SSD module [(host)]* initiates the deletion operation” by issuing “an *explicit command ... in association* with a particular FLBA,” to the controller. EX1005, [0038], [0059], Fig. 5. The delete command with FLBA sent to the controller designates a “*superblock address*” and a “PLBA.” *Id.* Son thus teaches a deletion command [“an erasure request from the host, via the host interface”], where that command includes an address designating a superblock address [“the

erasure request accompanied by the address corresponding to the specific one of the subdivisions”].

Like Ellis, Son teaches that “erasures” are carried out “at the superblock-level.” EX1005, [0029], [0058]-[0061]. Son therefore discloses “to responsively control erasure of each unerased erase block in the group respective to the specific one of the subdivisions.” Also, Son teaches that in connection with the erasure “the SSD module *updates the page status ... [and] valid page count*, ... associated with the superblock corresponding to the PLBA.” *Id.*, [0060]. Additionally, Son teaches updating superblock page utilization following erasures in “garbage collection” which provide separate disclosures of this limitation. *Id.*, [0061], [0074-0075], [0064].

The combined system of Ellis and Son thus renders this limitation obvious. EX1003, ¶¶98-102.

- (h) **1[g]: logic to, in connection with the erasure, detect a defect in one of the erase blocks in the group respective to the specific one of the subdivisions, to responsively substitute one or more different erase blocks of the flash memory for the one of the erase blocks for which the defect was detected**

Ellis teaches a controller with logic to detect a defect in a block in a superblock and remap a substitute block.

Ellis teaches “*erase blocks [may be] determined to be bad during normal operation of the storage device*” by controller. EX1004, 11:41-45. Thus, “[t]he

super blocks 502 can include remapped blocks for the unusable blocks 514.” *Id.*, 13:32-34. *See also, id.*, Claim 5. Determining which blocks are bad occurs during normal “*operation[s] including* reading, writing, or *erasing.*” *Id.*, 3:34-36, 19:66-20:4.

Ellis teaches that after detecting the bad block, *the controller contains logic to “remap” the bad block, which “specifically refers to the substitution of one physical erase block for another.*” EX1004, 11:41-45, 11:24-26, 20:59-67.

The combined system of Ellis and Son thus renders this limitation obvious. EX1003, ¶¶103-104.

- (i) **1[h]: wherein each said logic comprises at least one of hardware circuitry or instructions stored on non-transitory, machine-readable media that when executed are to control the operation of hardware circuitry.**

Ellis teaches that the control unit in the controller “may be a processor, an embedded processor, a microprocessor, *a hardware control logic...* or a combination thereof.” EX1004, 3:43-50. *See id.*, 3:1-9 (“The term “*module*” referred to herein can include software, hardware, or a combination thereof *the software can be machine code, firmware hardware can be circuitry*”), 2:24-25; EX1003, ¶¶105-106.

2. Claim 2

- (a) 2[a]: The memory controller of claim 1, wherein the host interface is also to receive a data copy request from the host,

The combination system of Ellis and Son teaches the controller of Claim 1, *see supra*.

Ellis teaches the controller is able to execute “*host [] recycle operations*” which are data copy operations where data from a source superblock is copied to a target superblock. EX1004, 8:67-9:3; 19:4-15 (“[d]uring the recycle operation, valid pages can be read from a donor super block and written to a new open super block”). Ellis teaches the host or controller may initiate copy commands. *Id.*, 20:40-44. Upon receiving a host-copy command through the “host interface,” the controller executes the copy request via the controller’s “memory access module.” *Id.*, 4:33-5:3.

To the extent Ellis does not disclose this claim, Son teaches that the host, “the *SSD module[,] instructs the SSD controller to copy data* ... to the selected target superblock,” EX1005, [0065]-[0068], [0017].

Thus the combination of Ellis and Son renders this limitation obvious. EX1003, ¶¶107-109.

- (b) 2[b]: wherein the memory controller is to responsively retrieve data from a source location in the flash memory, the source location associated with the specific one of the subdivisions, and write the retrieved data to a destination location in the flash memory, without sending the retrieved data to the host,

Ellis teaches that in response to the copy command as part of “the recycle operation, valid pages can be read [via the logic the controller],” EX1004, 19:8-15, 11:51-53, [*“respectively retrieve data”*]. The data is retrieved “from a donor super block,” *id.*, [*“a source location... associated with the specific one of the subdivisions”*]. Ellis teaches that the controller then takes that data and “writ[es] [the retrieved data] to a new open super block,” [*“a destination location in the flash memory”*]. *Id.*

Ellis teaches that this process may be carried out entirely by the controller’s “memory access module.” *Id.*, 18:15-34. *See also id.*, 4:66-5:3, 11:51-53, 3:39-42. Thus, Ellis teaches executing a data copy request “*without sending the retrieved data to the host.*” EX1003, ¶111.

To the extent Ellis does not disclose this claim, Son teaches a host copy command, where the host “identifies a source superblock and target superblocks” and “*simply instructs the controller to copy the contents of all pages in the target superblock over to the fresh superblock,*” EX1005, [0065], [0085]. This process, where the controller receives a copy request, retrieves the data from a “target” (or source) superblock and then writes the data “the fresh superblock,” is done “*using*

firmware calls [in the controller] to move the data within the drive” without sending the data to the host. *Id.*, [0068], EX1003, ¶112 (“Son’s disclosure that the copying is done by the firmware, which a POSA would understand to be done by the controller’s firmware, is a clear teaching that the copy operation is completed entirely by the controller without sending data to the host.”).

Thus the combination of Ellis and Son render this limitation obvious. EX1003, ¶¶98-102.

- (c) **2[c] and wherein the logic to track is to responsively update the tracked information for one of the subdivisions which corresponds to the destination location, in association with the write of the retrieved data into the destination location.**

Ellis teaches “*updating a valid page count for the [destination] super block*” after writing data for a “recycle” copy operation. EX1004, 20:11-20, Claim 5. Thus, Ellis teaches a controller with logic to update valid page count [“the logic to track is to responsively update the tracked information for one of the subdivisions which corresponds to the destination location”], following writing data in a copy operation [“*in association with the write of the retrieved data into the destination location*”].

To the extent Ellis does not disclose this claim, Son teaches to “*update the information* in the superblock table [(e.g., *page utilization*)] *to indicate the target location was written with valid data.*” EX1005, [0069]. Thus Son teaches this limitation.

The combined system of Ellis and Son renders this limitation obvious.
EX1003, ¶¶114-116.

3. Claim 8: The memory controller of claim 1, wherein the memory controller further comprises logic to receive a query from the host, and wherein the logic to send the information to the host is to do so as a response to the query.

The combination system of Ellis and Son teaches the controller of Claim 1, *see supra*.

Ellis teaches a system where the controller comprises logic to send “*status information* generated by the memory system *to the host system*.” EX1004, 4:51-57. A POSA would understand that the controller may send status information to the host in response to a host command or query for said information. EX1003, ¶57 (describing standard specification including a Host Command for “Get Log Page” including “SMART / Health information and Error information).

Ellis also teaches the “*host system 108 can request the memory controller 104 for reading*” “*metadata specific to the one of the super blocks*.” EX1004, 12:14-50, 3:22-28. Ellis thus teaches the host may issue a query or a read request directed to superblock metadata [“*memory controller further comprises logic to receive a query from the host*”], and that the controller provides that data in response to the query or read command, [“*and wherein the logic to send the information to the host is to do so as a response to the query*”].

To the extent Ellis does not teach this limitation, a POSA would be motivated to implement Son's teaching that the host may retrieve information by "*querying the SSD controller,*" EX1005, [0047], [0092], where that information is stored in flash memory, *id.*, [0034]. EX1003, ¶120.

The combined system of Ellis and Son renders this claim obvious. EX1003, ¶¶117-121.

4. Claim 10

- (a) **10[a] The memory controller of claim 1, wherein the memory controller further comprises logic to store erase count information for each of the subdivisions,**

The combination system of Ellis and Son teaches the controller of Claim 1, *see supra*.

Ellis teaches that the controller has logic configured to store "*program-erase count*" for each superblock, which is "defined as a numerical value *indicating a number of times ... the super blocks 502 are* programmed or *erased.*" EX1004, 11:58-67, 13:14-24, Claim 1.

To the extent Ellis does not teach this limitation, Son teaches the maintenance of "*wear count,*" "*indicat[ing] how many times a particular superblock has been erased,*" EX1005, [0080], [0039], ["erase count" for each "subdivision"].

The combined system of Ellis and Son thus renders this limitation obvious. EX1003, ¶¶122-125.

- (b) **10[b] and to send to the host, via the host interface, the erase count information for the specific one of the subdivisions, together with the address corresponding to the specific one of the subdivisions,**

For the reasons set out in Claim 1[d]-[f], the combined system of Ellis and Son teaches logic “to send to the host, via the host interface, ... information for the specific one of the subdivisions, together with the address corresponding to the specific one of the subdivisions” where that information includes page utilization and address information for a superblock. In addition, Ellis teaches that the controller may send the host “program-erase count.” Ellis discloses that the controller has logic to send “*control and status information generated by the memory controller 104 to the host system 108,*” EX1004, 4:51-53, including “NAND statistics” such as “*program-erase count*” for each superblock. *Id.* 10:27-32, 11:58-67. This erase count information may be sent by the controller with a “*superblock ID value,*” EX1005, [0065], [0039], [“the address corresponding to the specific one of the subdivisions”]. EX1003, ¶128.

Son also teaches that the host may retrieve information by “*querying the SSD controller,*” such as “*wear count*” and “*superblock address.*” EX1005, [0047], [0092].

Thus, the combination system of Ellis and Son thus render this limitation obvious. EX1003, ¶¶126-130.

- (c) **10[c] and wherein the logic to store the erase count information is to update the erase count information for the specific one of the subdivisions in association with the erasure.**

Ellis teaches that the controller has logic to *update “the program-erase count” “each time [the superblocks] are accessed for a program operation or an erase operation* by the memory access module 308.” EX1004, 13:14-25.

Similarly, Son describes the need to update “wear count” after an erasure—to “reflect [an] erasure of the [a] superblock, the SSD module modifies the entry in the superblock table ... *incrementing the wear count to reflect the erasure.*” EX1005, [0083].

Thus, the combined system of Ellis and Son thus render this limitation obvious. EX1003, ¶¶131-133.

- 5. Claim 11: The memory controller of claim 1, wherein the extent of the page utilization of the specific one of the subdivisions represents the amount of valid data written to the specific one of the subdivisions since the specific one of the subdivisions was last erased.**

The combination system of Ellis and Son teaches the controller of Claim 1, *see supra*. As described in Claim 1[c], Ellis and Son teach a memory controller with logic to track “extent of the page utilization of the specific one of the subdivisions.”

As described in Claim 10, Ellis and Son teach metadata tracking the erase count for each superblock.

A POSA would understand that where the erase count is greater than zero (the

subdivision has been erased at least once), the updated valid page count represents the amount of valid data written to a superblock since its last erasure. EX1003, ¶¶134-137. Thus Ellis and Son teach the tracking of metadata (erase count and valid pages written) which collectively “*represents the amount of valid data written ... since the specific one of the subdivisions was last erased.*” See also EX1005, [0040] (describing metadata describing Superblock state, e.g., “‘CLEAN,’ meaning the superblock has been erased and is available to take new write”).

6. Claim 12: The memory controller of claim 1, wherein the extent of the page utilization of the specific one of the subdivisions represents a remaining space of the specific one of the subdivisions that is available for writing without first erasing the specific one of the subdivisions.

The combination system of Ellis and Son teaches the controller of Claim 1, *see supra*. As described in Claim 1[c], Ellis and Son teach a memory controller with logic to track “extent of the page utilization of the specific one of the subdivisions.”

Ellis teaches that the superblocks have a defined size (total number of pages and blocks). EX1004, 7:7-31. Ellis also teaches tracking the number of valid pages in each superblock. *See supra*. A POSA would understand that “remaining space” is equal to the total number of pages minus the number of valid pages. EX1003, ¶¶138-140. Thus, Ellis teaches metadata including the number of pages written and the total number of pages in a superblock, where the difference in those two figures “*represents a remaining space ... that is available for writing without first erasing*

the specific one of the subdivisions.”

To the extent Ellis does not teach this limitation, a POSA would be motivated to incorporate the additional types of metadata described in Son, including the “*number of remaining free pages*,” available for each superblock without erasure.

The combined system of Ellis and Son thus discloses this limitation.

7. Claims 13, 14, 18 and 20

Petitioner relies on the same disclosures and reasoning discussed with respect to the Corresponding Claim Limitations, above, to show that claims 13, 14, 18, and 20 are rendered obvious by the prior art. Specifically, Petitioner contends the same disclosure that renders obvious a Corresponding Claim Limitation (set out below) also renders obvious the limitations of claims 13, 14, 18, and 20.

Claim Limitation	Corresponding Claim Limitation(s)
13[pre]	1[pre]-1[b]
13[a]	1[c]
13[b]	1[d]
13[c]	1[e]
13[d]	1[f]
13[e]	1[g]
14	2
18	8
20	10

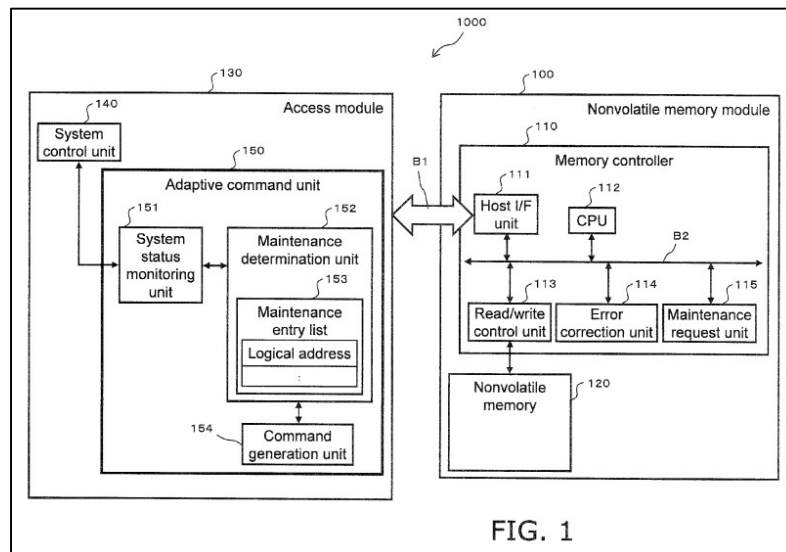
IX. GROUND II: CLAIMS 6, 9, 17, 19 ARE RENDERED OBVIOUS BY ELLIS IN VIEW OF SON AND YAMADA

A. Overview of the Prior Art

1. Yamada (EX1006)

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 at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Yamada teaches an SSD system where memory management is shared and requests for copying data originate from the controller and are commanded by the host. Yamada teaches that the SSD includes a controller with a “maintenance request unit,” and the access device (host) includes a “maintenance determination unit.” EX1006, [0012], [0017], [0031].



The controller identifies an LBA(s) based on comparison of metadata against thresholds. *Id.*, [0013], [0032], [0080]-[0082]. The controller then transmits the

address to the host via a maintenance request; the host may then issue a maintenance command, *e.g.*, where copying of data from a source location to a destination location “is to be performed.” *Id.*, [0014], [0109]-[0116].

B. Motivation to Combine

A POSA would have been motivated to combine the teachings of Yamada with the combination of Ellis and Son. Specifically, a POSA would have been motivated to incorporate Yamada teaching regarding host-initiated maintenance/copy commands. EX1003, ¶¶145-149.

First, Ellis, Son, and Yamada involve similar techniques to solve the same problem, *e.g.*, collecting, storing and using metadata for maintenance activities to improve the efficiency, performance and “integrity of data” in NAND flash memory systems. *E.g.*, EX1006, [0005]-[0011], [0207]-[0211], [0029].

Second, Ellis and Son both teach various forms of shared host and controller responsibility. Yamada describes the benefits of a particular form of shared host and controller responsibility.

Yamada teaches the benefits of a system with host-initiated maintenance/copy commands based on metadata received from the controller. First, incorporating the teaching of Yamada “enables data maintenance to be performed in accordance with the status of the nonvolatile memory system, without requiring the data maintenance to be performed together with normal processing of the system.” EX1006, [0036].

Second, it “enables the maintenance to be performed with a sufficiently long processing time allocated to the maintenance, and improves the data retention properties.” *Id.* Third, it “enables the system management status, such as the battery level, to be controlled in a manner that the maintenance will be completed in a reliable manner.” *Id.*

In light of the teaching in Ellis that memory management responsibility may be shared between the controller and the host, EX1004, 4:58-65, 20:40-44, Son’s teaching of the benefits of sharing responsibility between the host and controller, *see, e.g.*, EX1005, [0013]-[0014], and the benefits described in Yamada, a POSA would be motivated to incorporate Yamada’s teaching regarding host-initiated copy/maintenance commands which are carried out by the controller and are based on metadata collected by the controller, EX1003, ¶148.

Incorporating Yamada’s host-initiated copy commands based on controller request with the combination system of Ellis and Son would have been readily accomplished. EX1003, ¶¶147-149. Yamada provides a discrete embodiment of the shared controller and host responsibility described by Son and Ellis, where the responsibility for issuing a copy/maintenance command originates from the host as opposed to the controller. *Id.* A POSA would have had a reasonable expectation of success in incorporating this host command functionality—moving the copy

command responsibility from SSD controller firmware to host-side software—and it would have been straightforward and routine. *Id.*

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

C. Claim-by-Claim Analysis

1. Claim 6

- (a) **6[a] The memory controller of claim 2, the memory controller further comprises logic to track metadata for each of the subdivisions, the metadata indicating a time since data was programmed,**

The combination system of Ellis and Son teaches the controller of Claim 2, *see supra*.

As described in Ground I, Claim 2[a]-[c], Ellis teaches a “recycle operation” which includes a copy request for a superblock. To identify superblock to copy, Ellis teaches that the controller has logic to track time since data was programmed and to identify “super blocks 502 that are *older or written a longer time in the past*,” EX1004, 19:4-15, which is “metadata indicating a time since data was programmed.”

To the extent Ellis 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*,” EX1005, [0042], [0080].

The combination system of Ellis and Son thus renders this limitation obvious.
EX1003, ¶¶150-152.

(b) 6[b] to compare the tracked metadata to a threshold,

Ellis’s disclosure that data may be copied based on “super blocks that are older or written a longer time in the past,” EX1004, 19:4-15, is a clear teaching to a POSA that a comparison must be conducted—the terms “older” and “longer” inherently require a comparison—thus teaching this limitation. EX1003, ¶153.

In light of Ellis’s teaching that time since last programmed is metadata that may be used in copy operation, a POSA would be motivated to incorporate the teaching of Son which provides additional details on use of time since data was programmed metadata. Son teaches *comparing the “last modification time”* [“the tracked metadata”] *against “a predetermined time period”* [“to a threshold”]. EX1005, [0042], [0081], Claim 36 (“selecting a source superblock ... [based on] last modification times”); EX1003, ¶154. Son teaches that this comparison is useful for selecting superblocks in a copy operation. *Id.*

The combination system of Ellis and Son thus renders this limitation obvious.
EX1003, ¶¶153-155.

- (c) 6[c] and, responsive to a determination that the tracked metadata for the specific one of the subdivisions satisfies the threshold, to transmit to the host the address corresponding to the specific one of the subdivisions; and the data copy request is accompanied by the address corresponding to the specific one of the subdivisions.

As set out in Ground I, Claim 2, and Ground II, Claim 6[b], Ellis and Son teach selecting a superblock for a copy operation.

In the combined system of Ellis and Son, a POSA would be motivated to incorporate Yamada's teachings regarding a controller-initiated copy request as it describes additional utility for the claimed metadata in a system with shared host and controller functionality.

Yamada teaches that the controller, the maintenance request unit, "*determines whether an area (in units of blocks) in the nonvolatile memory 120 from which data has been read needs maintenance,*" *e.g.*, to be copied. EX1006, [0081]. Yamada teaches that the controller identifies blocks to be copied based on comparison to a threshold, a "*criterion value.*" *Id.*, [0032]-[0033]. Yamada teaches that where the blocks *satisfy the threshold* or criteria, the controller sends a "maintenance request" to the host, *e.g.*, the claimed "*data copy request,*" *id.*, [0032], [0033], [0099]. The data copy request sent by the controller to the host may be *accompanied by* a "*plurality of logical addresses of a plurality of blocks,*" *id.*, [0116], [0082], [0095], [0213].

Yamada further teaches that data in need of maintenance (copy request) may be identified because it “*has not been accessed for a long period of time*” or based on the “*elapsed time when ... written previously,*” and *comparing* this metadata against a *threshold* or criterion value. EX1006, [0211], [0164], [0204].

A POSA would apply Yamada’s teaching to the superblocks in Ellis, such that the maintenance request notification to the host requesting copying would include a superblock ID value or address [“*the address corresponding to the specific one of the subdivisions*”]. EX1003, ¶158.

The combined system of Ellis, Son and Yamada thus renders this limitation obvious. EX1003, ¶¶150-152.

2. Claim 9

- (a) **9[a] The memory controller of claim 1, wherein: the memory controller further comprises logic to assign a write destination in the flash memory, in which data accompanying the write request is to be stored, the write destination corresponding to the specific one of the subdivisions,**

The combination system of Ellis and Son teaches the controller of Claim 1, *see supra*.

As set out in Ground I, Claim 1[d], Ellis teaches that the host issues requests to a superblock as a single unit.

Ellis teaches after the controller receives a logical address from the host corresponding to a superblock, the controller includes logic to “*assign a write*

destination,” by performing “a mapping operation,” where the controller assigns or “map[s]” the data accompanying the write request “to physical pages” “to be stored.” EX1004, 15:45-53, Claim 15. Ellis teaches that the write destination corresponds to a superblock location where the data is stored [*“the write destination corresponding to the specific one of the subdivisions”*]. EX1003, ¶¶161-163.

Thus, the combination system of Ellis and Son renders this limitation obvious.

(b) 9[b] and to transmit to the host a read address that will be used to retrieve the data specifically from the write destination;

A POSA would be motivated to incorporate the teaching of Yamada in the combination system of Ellis and Son. Yamada provides additional implementation details for shared host and controller memory management responsibility where the controller provides address information to the host upon completion of operations. EX1003, ¶164.

Yamada teaches that a write operation is carried out as part of a maintenance request (from the controller) and maintenance command (from the host). *E.g.*, EX1006, [0109]. After the data is written, *the controller transmits a notification to the host which includes the “address[(es)] of [] block[s] included in the nonvolatile memory for which the [writing] has been completed,” id.*, [0111], and updates “write history,” *id.*, [0166]. Yamada teaches that the host may then use that address “for reading data from ... a plurality of blocks.” *Id.*, [0211]-[0213].

The combined system of Ellis, Son and Yamada renders this limitation obvious. EX1003, ¶¶164-167.

- (c) **9[c] and the memory controller further comprises logic to receive, from the host, and via the host interface, a read request for the data, accompanied by the read address, and to identify both of the specific one of the subdivisions and the write destination from the read address.**

Ellis teaches that the controller is configured to receive “*host read operations*” and to identify the “*data requested for a host read operation.*” EX1004, 9:4-10. As described for Ground I, Claim 1, the read request is mediated through the “host interface.”

Ellis also teaches that “[t]he host system 108 can request the memory controller 104 for reading.... data from ... a logical address space of a storage” [“*a read request for the data, accompanied by the read address*”]. *Id.*, 3:25-28. The controller uses this read request and the associated logical address (read address) to perform “*a mapping operation*” to identify the superblock where the data is stored and the write address. *Id.*, 15:45-53. Ellis thus teaches this limitation.

To the extent Ellis does not disclose this limitation, a POSA would be motivated to combine it with Yamada which provides implementation details for shared host and controller memory management responsibility where the controller provides address information to the host upon completion of operations. EX1003, ¶170. Yamada teaches that, upon writing data to a location, *e.g.*, in conjunction with

a maintenance command (erase and write), the controller sends the host the destination address. Subsequently, when the host initiates a read command for that data, it transmits ***“a command for reading valid data including an address (logical address) of data to be read, and then the read/write control unit [(in the controller)] reads data from a designated address in the nonvolatile memory 120 (a physical address corresponding to the logical address designated by the access module 130).”*** *Id.*, [0095]. A POSA would apply Yamada’s teaching to the subdivisions (superblocks) in Ellis.

Alternatively, a POSA would be motivated to incorporate Son’s teaching that the host may issue a ***“read request”*** to the controller, and that the read request includes the “PLBA” or the physical destination location of the data previously written: ***“The SSD module may then send a read request, including the identified PLBA, to the SSD.*** In response, the SSD module may receive a copy of the data associated with that PLBA that it can then provide to the file system or otherwise utilize.” EX1005, [0056]. As discussed in Ground I, Claim 1[d], the PLBA corresponds to a particular superblock. Son thus teaches that the controller has logic to receive “a read request,” which includes a PLBA, which is a read address that the controller uses to ***“identify both of the specific one of the subdivisions and the write destination from the read address.”***

The combined system of Ellis, Son and Yamada thus renders this limitation obvious. EX1003, ¶¶168-172.

3. Claims 17 and 19

Petitioner relies on the same disclosures and reasoning discussed with respect to the Corresponding Claim Limitations, above, to show that claims 17 and 19 are rendered obvious by the prior art. Specifically, Petitioner contends the same disclosure that renders obvious a Corresponding Claim Limitation (set out below) also renders obvious the limitations of claims 17 and 19.

Claim Limitation	Corresponding Claim Limitation(s)
17	6
19	9

X. GROUND III: CLAIMS 1, 2, 8, 10-14, 18, 20 ARE RENDERED OBVIOUS BY SINCLAIR IN VIEW OF YU AND SON

A. Overview of the Prior Art

1. Sinclair (EX1007)

U.S. Pat. No. 7,739,444 (Sinclair) was filed December 26, 2006, published June 26, 2008, and granted June 15, 2010 and is prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Sinclair teaches an SSD organized into “metablocks,” where “four blocks” “from each plane [are] logically linked together to form a metablock.” EX1007, 9:52-57, 7:2-4. Sinclair teaches that metablock metadata is tracked and dynamically updated, *id.*, 31:65-32:4, including “[t]he number of pages” and “the number of

pages of unprogrammed capacity” for each metablock, *id.*, 26:5-17.

In order to manage the metablocks, Sinclair teaches shared host and controller responsibility. For example, Sinclair describes the benefits of the controller transmitting information to the host, including, *e.g.*, physical characteristics of the metablocks such as “[t]he lowest logical address that is mapped to a first page of a physical metablock.” *Id.*, 18:16-28.

2. Yu (EX1008)

U.S. Pat. Pub. 2012/0284587 (Yu) was filed July 2, 2012 and published November 8, 2012, and is prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and 102(e).

Yu describes an SSD system comprising a host and a controller, where the host and controller work cooperatively to optimize memory storage and to prolong longevity. The system “uses a barrage of advanced management techniques that together reduce the number of writes to flash, hence reducing program-erase cycles on the underlying flash memory,” EX1008, [0012], “to solve the endurance issues of NAND flash memory,” *id.*, [0291]. Yu describes the use of an “Endurance Translation Layer” to “manage the interaction of DRAM and Flash Memory.” *Id.*, [0070].

Yu teaches the maintenance of metadata, including page utilization through maintenance of a “page status table,” and other information describing block-level usage, *e.g.*, (“erase counters and block status”). *Id.*, [0083].

Yu describes functionality where the host is able to query the controller for metadata through “Self-Monitoring, Analysis and Reporting Technology” (“S.M.A.R.T”) commands. *Id.*, [0064], [0085], [0225]-[0228]. The S.M.A.R.T. functionality is a discrete example of shared responsible between the host and controller to “monitor and control error correction, wear, bad block, and other flash memory management.” *Id.*

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

B. Motivation to Combine

POSA would have been motivated to combine the teachings of Sinclair, Yu and Son.

First, the systems in Sinclair, Son and Yu each teach “advanced management techniques” to optimize storage and retention of memory data in NAND flash storage devices. EX1008, [0012], [0291], EX1005, [0002], [0014], EX1007, 8:46-53, 3:52-5:13. Each reference discloses techniques of shared host and controller memory management that rely on collected metadata such as page utilization.

Second, in light of the substantial similarities between the systems of Sinclair and Son, a POSA would be motivated to incorporate Son’s teaching into the system

of Sinclair. Sinclair provides three key teachings to a POSA regarding means to optimize memory management.

Sinclair teaches the use of metablocks, comprising “four or more blocks” “in order to increase the degree of parallelism in data programming and reading.” EX1007, 3:39-48, 9:52-55. Like Sinclair, Son, as described *supra* with respect to Ground I, teaches the use of superblocks, which are analogous to metablocks, as a means to increase SSD performance.

Sinclair also describes the utility of storing metadata at the metablock level, where the metadata includes the number of pages of valid data in each metablock, where that metadata used for, *e.g.*, maintenance operations. *Id.*, 31:40-32:10, 26:5-17. Like Sinclair, as described *supra* with respect to Ground I, Son also teaches superblock-level metadata such as page count for use in maintenance operations.

Sinclair teaches that “improved performance and endurance may also be realized” through shared memory management responsibility between the host and controller, *e.g.*, *id.*, 4:23-47, and discloses systems with variable host control, as shown in, *e.g.*, Fig 8A, 8B. Like Sinclair, as described *supra* with respect to Ground I, Son teaches the benefits of sharing memory management responsibility between the host and controller.

In light of the identity in core features between Sinclair and Son, a POSA would be motivated to incorporate the teachings of Son into the system in Sinclair.

EX1003, ¶¶176-189. Specifically, a POSA would incorporate Son's teaching regarding implementation details for superblock metadata and commands, to the metablock metadata and commands in Sinclair. The incorporation of additional types of metadata and logic for commands to the system in Sinclair would be a straightforward and routine modification given the overlap in the systems' design. EX1003, ¶187.

Third, a POSA would be motivated to incorporate Yu's host and controller S.M.A.R.T. functionality, which is a narrow, straightforward means to increase SSD endurance and longevity into the combination system of Sinclair and Son.

Like Sinclair and Son, Yu teaches the maintenance of metadata including page utilization and that memory management is shared between the host and controller. EX1008, [0055]. Yu teaches that the host may issue commands to, and requests for information from, the controller through "S.M.A.R.T." functionality. The host initiated "S.M.A.R.T." commands enable the host to "monitor some important data from the SSD device" including "wear," "wear-leveling count" "and other flash memory management." *Id.*, [0064]. The incorporation of S.M.A.R.T. functionality, a discrete implementation of the shared controller and host responsibility, would also be a straightforward and routine modification in light of the variable degrees of shared host and memory management disclosed in Sinclair (and Son). EX1003, ¶¶183-186.

A POSA would be motivated to combine Sinclair, Son and Yu.

C. Claim-by-Claim Analysis

1. Claim 1

(a) 1[pre]

To the extent the preamble is limiting, it is disclosed by Sinclair. Sinclair teaches use of “*controllers that interface with the host and control operation of the flash memory,*” EX1007, 7:2-4, [“a memory controller to control flash memory”].

Sinclair teaches that the “*flash memory*” is comprised of “*metablock units,*” [“subdivisions”] where “*four blocks*” “from each plane [are] logically linked together to *form a metablock.*” EX1007, 9:52-57, 7:2-4.

Sinclair also teaches that each block in a metablock “*must be erased before any data stored in the block can be overwritten.*” *Id.*, 3:15-18.

Therefore, Sinclair renders the preamble obvious. EX1003, ¶¶190-191.

(b) 1[a]

Sinclair teaches a controller with a “*host interface*” in Figure 1. EX1007, Fig. 1. “The memory system 2 of FIG. 1 includes flash memory 7, and circuits 8 that ... *interface with the host.*” *Id.*, 7:2-31; EX1003, ¶¶192-193.

(c) 1[b]

Sinclair teaches a “*memory interface*” as shown in Figure 2, where the controllers “control operation of the flash memory.” EX1007, 7:2-31, Fig. 2; EX1003, ¶¶194-195.

(d) 1[c]

Sinclair teaches a controller with logic to track information about metablocks in lists, including the “*amount of valid data*” and the “*number of pages of unprogrammed capacity*” for each metablock [“page utilization of each of the subdivisions”]. EX1007, 31:46-49, 26:5-17—*Note*, block and metablock are used interchangeably in Sinclair unless specified otherwise, *see id.*, 16:22-26.

To the extent Sinclair does not teach this limitation, a POSA would be motivated to incorporate the additional variants of superblock metadata disclosed in Son. As discussed for Ground I, Claim 1[c], Son teaches the claimed page utilization metadata.

The combination system of Sinclair and Son thus render this limitation obvious. EX1003, ¶¶196-197.

(e) 1[d]

Sinclair teaches that “a host ... needs to know the physical characteristics of the memory with which it is operating,” which are acquired from the controller, including [“[t]he lowest logical address that is mapped to a first page of a physical metablock”]. EX1007, 18:16-43. That means that the logical address designates a specific metablock or subdivision.

Sinclair teaches that the *controller has logic to receive host write commands accompanied by “logical addresses assigned by the host,”* EX1007, 13:9-12, 12:20-

30. The logical addresses accompanying the write request are converted by the controller to “*physical addresses within a metablock*,” e.g., a metablock address designating a specific one of the subdivisions. EX1007, 13:4-12, 12:20-30. Thus, Sinclair teaches that the controller has “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions.”

Additionally, Sinclair teaches that page utilization is updated or “*changes constantly as data are written*.” *Id.*, 31:65-32:4. Thus Sinclair teaches this limitation.

To the extent Sinclair does not teach this limitation, a POSA would be motivated to incorporate the teaching from Son regarding write requests associated with a particular superblock, as discussed *supra* for Ground I, Claim 1[c].

Sinclair and the combination system of Sinclair and Son thus render this limitation obvious. EX1003, ¶¶198-202.

(f) 1[e]

Sinclair teaches the benefits of “possible divisions of [] functions as between the host and the memory system.” EX1007, 16:26-67, Figs. 8A-8C. In light of this teaching and that of Son, discussed *supra*, a POSA would be motivated to incorporate the teaching of Yu.

Yu teaches a particular form of optimizing a flash file system through host monitoring and requesting data for memory management. EX1008, [0228], [0064]. Yu describes a “S.M.A.R.T.” function (“stands for Self-Monitoring, Analysis and Reporting Technology”), where the host may “monitor some important data from the [controller]” including “health information,” “wear,” “wear-leveling count,” “bad block, and other flash memory management” responsibilities. *Id.*, [0064], Claim 6.

Yu teaches the data “*can be requested by the host* through S.M.A.R.T. commands or vendor commands” *and sent by the controller to the host*—the controller has “logic to send the host [] information” about memory storage. *Id.*, [0085], [0232]. S.M.A.R.T. aggregates this data to send to the host, where that the aggregated data is stored in various “tables.” *Id.*, [0085]. S.M.A.R.T. functionality is flexible and may “be expanded [to] collect[] other important data from the SSD” which the host may use “to analyze the data for debugging, statistics, etc.” *Id.*, [0225].

Yu also teaches the use of a “[p]age status table,” as well as other metrics representing page utilization, *id.*, [0083], that “show[] the status of each individual page of flash,” *e.g.*, “an empty page” or “a used page.” *Id.*, [0083], [0092], [0226].

A POSA would understand, in order to implement Yu’s S.M.A.R.T. functionality in the combined system of Sinclair and Son, that the S.M.A.R.T.

information sent to the host would include the combined system's metablock-level metadata such as page utilization. EX1003, ¶207. Also, to apply the teaching of Yu to the combined system, a POSA would understand that the information sent by the controller to the host in connection with the S.M.A.R.T. command must include a superblock or metablock address, as described for Ground I, Claim 1[d]-[f]. *Id.* Thus, the combined system teaches ***“logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.”***

The combination of Sinclair, Yu and Son renders this limitation obvious. EX1003, ¶¶203-208.

(g) 1[f]

Sinclair teaches that the host issues, and the controller has logic to receive, “delete” or erasure commands. EX1007, 21:50-57, 27:61-28:12. Sinclair teaches that an erasure request may be directed to a metablock as a single unit. *Id.*, 27:61-28:12, 9:39-67. A POSA would understand that an erasure request from the host directed to a metablock as a single unit must include an address designating a single metablock [***“logic to receive an erasure request from the host, via the host interface, the erasure request accompanied by the address corresponding to the specific one of the subdivisions”***]. EX1003, ¶209.

Sinclair teaches that upon receipt of the erase request identifying a metablock, “[a]ll of the *cells within a metablock are typically erased together*” [“to responsively control erasure of each unerased erase”]. EX1007, 9:39-67.

Sinclair further teaches that after erasure, the metadata representing page utilization is “*updated constantly*” including when “data [is] ... *erased*.” EX1007, 31:65-32:4, 16:22-26.

To the extent Sinclair does not teach this limitation, as discussed *supra* for Ground I, Claim 1[f], Son discloses that “erasures” are carried out “at the superblock-level,” EX1005, [0029], which include a host erasure command with a FLBA corresponding to a “superblock address,” *id.*, [0038], where page utilization is updated, *id.*, [0060]-[0061].

The combination of Sinclair and Son renders this limitation obvious. EX1003, ¶¶209-211.

(h) 1[g]

Sinclair teaches that if a defect is detected in a block within the metablock a different block is responsively substituted:

One or more of these extra blocks may be provided as redundant blocks for *substitution for other blocks that may become defective during the lifetime of the memory*. The logical grouping of blocks contained within individual metablocks may usually be changed for various reasons, including the *substitution of a redundant block for a defective block originally assigned to the metablock*.

EX1007, 13:1-9. Thus, Sinclair teaches a controller with logic to “detect a defect” in a metablock and “responsively substitute [a] different erase block” into that metablock. A POSA would understand that detection of a defective block “during the lifetime of the memory” may be “*in connection with the erasure.*” EX1003, ¶212.

To the extent that Sinclair does not teach this limitation, a POSA would be motivated to incorporate Yu’s teaching of “a process for detection and handling of bad pages and bad blocks” where “*[a]fter the block is erased, the status of each page in that block is checked to determine if it is bad.*” EX1008, [0241]-[0243].

The combination of Sinclair and Yu teach this limitation. EX1003, ¶¶212-213.

(i) 1[h]

Sinclair teaches a controller where the logic is stored on “*circuitry* of a [] flash memory system,” including on “*a single integrated chip*” with “*dedicated circuits* 33 and 34, *when utilized, execute specific algorithms that could otherwise be executed by the processor 27 under firmware control.*” See, e.g., EX1007, 7:3-8:23, Fig. 1. Thus, Sinclair renders this limitation obvious. EX1003, ¶¶214-215.

2. Claim 2

The combination system of Sinclair, Yu, and Son renders Claim 1 obvious.

(a) 2[a]

Sinclair teaches reclaim commands which are copy operations termed reclaim operations, where the controller has logic for “*[c]opying of the valid data of one file*

from the source reclaim [metablock] to the destination [metablock],” “[i]f the host has issued an idle command” [“wherein the host interface is also to receive a data copy request from the host”]. EX1007, 32:11-52.

A POSA would understand that the idle command is equivalent to “reclaim_on” command or an affirmative copy command. Sinclair includes a “cross-reference” to “Ser. No. 11/259,423, entitled ‘Scheduling of Reclaim Operations in Non-Volatile Memory’” now U.S. Pat. No. 7,409,489 (EX1009). Sinclair specifically references this patent in connection with the reclaim operation process described. EX1007, 32:11-26. This patent provides a definition for Sinclair’s use of a host “idle command” as equivalent to a host reclaim or copy command. EX1009, 23:36-46 (“The ‘*Reclaim_on*’ is equivalent to an ‘idle’ command”); *id.*, 18:63-65.

Thus the combined system of Sinclair, Yu and Son renders this limitation obvious. EX1003, ¶¶216-219.

(b) 2[b]

Sinclair teaches a reclaim command that triggers “[c]opying of the valid data of one file from the source reclaim [metablock],” [*a source location in the flash memory*], and writing it “to the destination [metablock]” [*and write the retrieved data to a destination location in the flash memory*]. EX1007, 32:11-52, 30:25-29, 29:17-27, 8:32-45. The reclaim process is done entirely by the controller, following

a host command [*“without sending the retrieved data to the host”*]. *Id.*; EX1003, ¶220.

To the extent Sinclair does not disclose this limitation, as discussed *supra* for Ground I, Claim 2, Son teaches this limitation.

Thus the combined system of Sinclair, Yu and Son renders this limitation obvious. EX1003, ¶¶220-222.

(c) 2[c]

As described in Ground III, Claim 1[d]-[f], Sinclair teaches that the controller has logic to update metablock page utilization after write operations. As described in Claim 2[a]-[b], the reclaim operation includes a write operation. Thus Sinclair teaches a controller with logic to update page utilization after writing to a destination metablock pursuant to a reclaim operation [*“the logic to track is to responsively update the tracked information for one of the subdivisions which corresponds to the destination location, in association with the write of the retrieved data into the destination location”*].

To the extent Sinclair does not disclose this limitation, as discussed *supra* for Ground I, Claim 2[c], Son teaches this limitation.

Thus the combined system of Sinclair, Yu and Son renders this limitation obvious. EX1003, ¶¶223-225.

3. Claim 8

The combination system of Sinclair, Yu and Son teaches the controller of Claim 1, *see supra*.

Sinclair teaches that information may be “supplied to the host” by the controller, including information regarding “the characteristics of the physical memory blocks.” EX1007, 4:54-60. Sinclair further teaches, *e.g.*, that there can be a “vendor specific ***command that is issued by the host***,” which acts as a query, where the ***controller “then operates to return the stored parameter values to the host.”*** *Id.*, 18:48-60.

As discussed with respect to 1[e], Yu teaches a controller which includes logic for S.M.A.R.T. queries from the host [***“logic to receive a query from the host”***], and sends to the host the information requested in response to a host query, [***“and wherein the logic to send the information to the host is to do so as a response to the query”***].

As described for Ground I, Claim 8, Son teaches this limitation.

The combination system of Sinclair, Son and Yu renders this claim obvious. EX1003, ¶¶226-230.

4. Claim 10:

(a) 10[a]

The combination system of Sinclair, Yu and Son teaches the controller of Claim 1, *see supra*.

Sinclair teaches that the controller has logic to store the “experience *count of the number of times the [metablock] has been erased* and re-programmed” [“logic to store erase count information”]. EX1007, 10:65-11:5.

As described for Ground I, Claim 10[a], Son teaches this limitation.

The combination system of Sinclair, Son and Yu renders this limitation obvious. EX1003, ¶¶231-234.

(b) 10[b]

The combination system of Son, Sinclair and Yu has Yu’s “S.M.A.R.T. functionality.”

As discussed for Claim 1[e], in response to a host S.M.A.R.T. command, the controller includes logic to send to the host metadata including, “*wear-leveling count*,” e.g., erase information. EX1008, [0064], [0236]; EX1003, ¶237. Also, Yu teaches that the controller stores a “[b]lock erase count table [that] keeps track of erase counters.” *Id.*, [0083].

A POSA would understand, in order to implement Yu’s S.M.A.R.T. functionality in the combined system of Sinclair and Son using metablocks, that (i) the S.M.A.R.T. erase count information sent to the host would correspond to individual metablocks [“*information for the specific one of the subdivisions*”], and (ii) the information would be sent along with a superblock or metablock address

[“*with the address corresponding to the specific one of the subdivisions*”], as described for Ground I, Claim 1, and Ground III, Claim 1. EX1003, ¶238.

Thus the combined system of Sinclair, Son, and Yu renders this limitation obvious. EX1003, ¶¶235-239.

(c) 10[c]

As stated for Claim 1, the combination system of Sinclair, Yu and Son teaches the controller of Claim 1.

As described for Claim 1[c], Sinclair teaches tracking erase count for metablocks. A POSA would understand that in order to accurately track the number of times a block or subdivision has been erased, the erase count must be updated to account for subsequent erasure. EX1003, ¶240.

A POSA may also incorporate the explicit teachings from Son and Yu. As described for Ground I, Claim 10, Son teaches this limitation. Yu also teaches that erase count is updated after erasure. *See, e.g.*, EX1008, Claim 17 (“when an erased block in a flash memory has been erased, increasing an erase count”).

Therefore, the combination system of Sinclair, Son and Yu renders this limitation obvious. EX1003, ¶¶240-243.

5. Claim 11

The combination system of Sinclair, Yu and Son teaches the controller of Claim 1, *see supra*.

As described in Ground III, Claim 1[c] and Claim 10[a], Sinclair teaches the controller has logic to track data for each metablock (subdivision) including the “*number of pages*” written in a metablock and *erase count*. EX1007, 26:5-17, 10:65-11:5. A POSA would understand that if the erase count is greater than zero (has been erased at least once), the updated valid page count “*represents the amount of valid data written to the specific one of the subdivisions since the specific one of the subdivisions was last erased.*” EX1003, ¶¶244-247

The combined system of Sinclair, Son and Yu renders this claim obvious.

6. Claim 12

The combination system of Sinclair, Yu and Son teaches the controller of Claim 1, *see supra*.

Sinclair teaches the controller has logic to track “*pages of unprogrammed capacity*” for each metablock. EX1007, 26:5-17; *see also* Ground III, Claim 1[c]. A POSA would understand that “pages of unprogrammed capacity” in a metablock represents “*represents a remaining space of the specific one of the subdivisions that is available for writing without first erasing the specific one of the subdivisions.*”

The combined system of Sinclair, Son and Yu thus renders this claim obvious. EX1003, ¶¶248-249.

7. Claims 13, 14, 18 and 20

Petitioner relies on the same disclosures and reasoning discussed with respect to the Corresponding Claim Limitations, above, to show that Claims 13, 14, 18, and 20 are rendered obvious by the prior art. Specifically, Petitioner contends the same disclosure that renders obvious a Corresponding Claim Limitation (set out below) also renders obvious the limitations of Claims 13, 14, 18, and 120.

Claim Limitation	Corresponding Claim Limitation(s)
13[pre]	1[pre]-1[b]
13[a]	1[c]
13[b]	1[d]
13[c]	1[e]
13[d]	1[f]
13[e]	1[g]
14	2
18	8
20	10

XI. GROUND IV: CLAIMS 6, 9, 17, 19 ARE RENDERED OBVIOUS BY SINCLAIR IN VIEW OF YU, SON AND JUNG

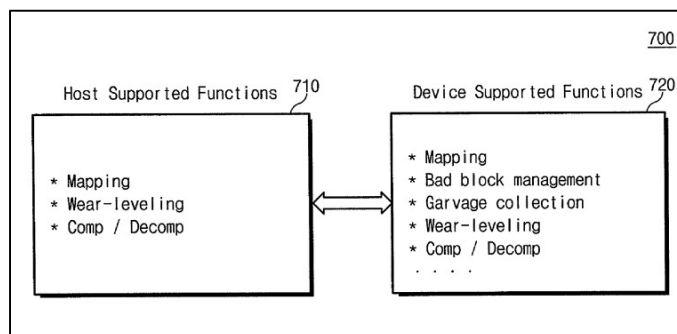
A. Overview of the Prior Art

1. Jung (EX1010)

U.S. Pat. Pub. No. 2008/0189485 (Jung) was filed August 3, 2007 and published August 7, 2008 and is prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Jung teaches a system of “Cooperative memory management,” where responsibilities for memory management are split or shared between the host and

controller. *E.g.*, EX1010, [0044]. Jung describes various examples regarding how certain functions may be shared between the host and controller, *e.g.*, Figure 7:



Id., Fig. 7. To facilitate the host’s memory management responsibilities, Jung teaches that the controller “loads meta information, such as an erase count, garbage collection data and a mapping table” to the host. *Id.*, [0038].

B. Motivation to Combine

Jung is in the same field as endeavor as Sinclair, Yu, and Son. Jung also teaches means to optimize memory systems such as “NAND flash” “SSD[s].” *Id.*, [0003]-[0004]. *See* EX1003, ¶¶257-265.

Jung represents the same techniques as Sinclair, Yu and Son to address the challenges of optimizing memory management. EX1003, ¶258. Jung is an additional disclosure that shared memory management responsibility between the controller and the host is a means to “efficiently distribut[e] the resources of such systems” and to take advantage of the greater resources available to a host (as compared to a controller) to manage memory effectively. EX1010, [0002]-[0009].

Jung teaches the shortcomings of memory management carried out exclusively by the controller: “Unfortunately, the resources of flash controllers are generally smaller than those of hosts. Thus, a resulting execution speed of such FTL functions performed in the controller is typically slow. In some cases, the FTL functions cannot be performed at all, such as where the controller becomes overloaded.” EX1010, [0005]. Jung, like Sinclair, Yu and Son, describes a solution to these shortcomings based on shared host and controller responsibility. EX1003, ¶¶259-260.

Specifically, Jung teaches “Cooperative memory management,” where certain “memory management functions [are] performed by the host for the device.” EX1010, [0011]. Jung teaches that those certain “functions, such as an address mapping function, a wear-leveling function and a garbage collection function,” may be performed by the host, *id.*, [0032], and are based on “meta information, such as an erase count, garbage collection data and a mapping table” shared by the controller “to the host,” *id.*, [0038], [0048].

Jung comprises a similar implementation of the shared memory management functionality present in Sinclair, Son and Yu, where the controller sends certain types of metadata to the host to support its memory management. A POSA would be able to incorporate Jung’s teaching, the controller sharing additional types of

metadata with the host, through known techniques with a reasonable likelihood of success. EX1003, ¶¶264-265.

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

C. Claim-by-Claim Analysis

1. Claim 6

(a) 6[a]

As set out in Ground III, the combination system of Sinclair, Yu and Son teaches the controller of Claim 2 with “logic to track metadata for each of the subdivisions,” *e.g.*, tracking the page count for each metablock or subdivision.

A POSA would be motivated to incorporate additional variants of metadata identified in Son for use in Sinclair. For example, as described for Ground II, Claim 6[b], Son teaches that “to select the source and target superblocks,” it is useful to consider “*last modification time*” *for each superblock* (and “valid page count”) to select a metablock to be copied and the use of thresholds for those forms of metadata. A POSA would incorporate that teaching into the system in Sinclair.

This teaching is consistent with Sinclair, EX1007, 31:46-49, and Yu, which teaches storing “*modify date/time*” metadata which is accessed by the host prior to issuing a “read or write” command. EX1008, [0048], [0088], [0128]-[0129].

Thus the combined system of Sinclair, Son and Yu teaches “memory controller further comprises logic to track metadata for each of the subdivisions, the metadata indicating a time since data was programmed.” EX1003, ¶¶266-268.

(b) 6[b]

As described for Ground II, Claim 6[b], Son teaches *to compare “last modification time” against a threshold*. EX1005, [0042], [0065], [0081], Claim 36.

A POSA would be motivated to include the threshold disclosed in Son due to its utility for maintenance and copy operations, as described for Ground II, Claim 6[b]. A POSA would be further motivated to incorporate this threshold because Sinclair already teaches the use of thresholds for reclaim functions. EX1007, 31:56-61 (discussing a “*set threshold amount*” to select ... a chosen reclaim block”).

The incorporation of time-since-last-accessed metadata and related thresholding into the combination system would be straightforward and easily accomplishable. EX1003, ¶269. Indeed, a POSA may also rely on Yu’s teaching of metadata thresholds, e.g., an “*access counter ... threshold*,” EX1008, [0060], [0190].

Thus the combination of Sinclair, Yu and Son teaches “to compare the tracked metadata to a threshold.” EX1003, ¶¶269-272.

(c) 6[c]

As set out in Ground IV, Claim 6[a]-[b], the combined system of Sinclair, Yu and Son, teaches a copy operation where the metablock to be copied is identified based on satisfying a threshold for last modification time metadata. As set out in Ground III, Claim 1[f]-[g] and Claim 2, the combined system issues copy commands at the metablock level, designated by, *e.g.*, a metablock address. Thus, the combined system teaches that “*responsive to a determination that the tracked metadata for the specific one of the subdivisions satisfies the threshold,*” that data is copied using a metablock address in the command operation, [“*address corresponding to the specific one of the subdivisions*”].

In order to implement the copy operation of the combined system, a host would rely on Yu’s S.M.A.R.T. functionality which enables host “S.M.A.R.T.” commands to “monitor some important data from the SSD device” for “flash memory management.” EX1007, [0064], [0085], [0232]. A POSA would understand that instances where the metadata for a *metablock surpasses a threshold is “important data” that the S.M.A.R.T. functionality “monitor[s]” and would report “to the host.” Id. See also* EX1003, ¶275. Thus, in the combined system of Sinclair, Son and Yu, the controller would “*transmit to the host*” information identifying the metablock which exceeds the time since last accessed threshold using an addresses “*corresponding to the subdivision.*”

To the extent the combination of Sinclair, Yu and Son does not teach this limitation, a POSA would be motivated to include the teaching from Jung regarding the scope of information sent to the host.

Sinclair teaches in “a memory system using metablocks,” a host must know, “[t]he lowest logical address that is mapped to a first page of a physical metablock.” EX1007, 18:16-43. Jung’s teaching that complete mapping data should be shared with the host represents a modest extension on top of Sinclair’s existing teaching that the host needs information about the physical memory storage. EX1003, ¶278.

Jung teaches that the *controller sends “mapping” metadata* and other information, *e.g.*, “an erase count [and] garbage collection data,” *“to the host”* so the host can carry out memory management functions. EX1010, [0038], [0010], Claim 59. Jung teaches that such mapping/address information *is sent to the host so that it may issue “a flash copy command”* to the controller [“the data copy request is accompanied by the address corresponding to the specific one of the subdivisions”] for host memory management. *Id.* [0010], [0048] (“*Invocation of the mapping function 910 causes the host to issue one of a plurality of mapping commands 912, such as a flash read command, a flash write command, a flash copy command, or a flash erase command.*”), [0015]. It would be obvious to a POSA to incorporate Jung’s teaching that, when transmitting to the host that a metablock exceeds a threshold, the controller may also transmit to the host address mapping

information to identify or locate the data exceeding the threshold to be copied. EX1003, ¶¶278-282.

Thus the combination of Sinclair, Son, Yu and Jung renders this limitation obvious. EX1003, ¶¶273-283.

2. Claim 9

(a) 9[a]

As set out in Ground III, the combination system of Sinclair, Yu and Son teaches the controller of Claim 1.

Sinclair teaches that “*the host writes data* to the memory system” using logical addresses, and the “*controller converts the logical addresses* assigned by the host to *physical addresses within a metablock*,” “*where the data are actually stored*”—e.g., controller has “logic to assign a write destination in the flash memory, in which data accompanying the write request is to be stored, the write destination corresponding to the specific one of the subdivisions.” EX1007, 13:9-12, 12:20-45, 2:66-3:6.

Yu also teaches that the controller engages in block-level assignment (“block mapping”) where “an entire logical block from the host is mapped to an entire physical block in flash memory.” EX1008, [0096]. Applying this teaching in the combined a system, a POSA would understand the need to implement superblock or metablock-level mapping [“*assign a write destination in the flash memory ... in*

which data accompanying the write request is to be stored, the write destination corresponding to the specific one of the subdivisions”].

Thus the combination of Sinclair, Son, Yu and Jung renders this limitation obvious. EX1003, ¶¶283-286.

(b) 9[b]

A POSA would be motivated to incorporate the teaching of Jung into the combined system of Sinclair, Yu and Son. Jung teaches means of accomplishing shared host and controller memory management responsibilities of the combined system.

Jung teaches that the controller may share mapping and address information with the host to facilitate host management. Specifically, Jung teaches that the controller *“load[s]” or transmits “mapping table data” to the host* [“to transmit to the host a read address”]. *See, e.g.*, EX1010, Claim 59. The “mapping table data” includes the read address for data that has been written. *Id.*, [0015]-[0016], [0048]. The *host then uses this mapping information to issue “a flash read command”* to the written data [“will be used to retrieve the data specifically from the write destination”]. EX1010, [0048]; EX1003, ¶¶287-288.

Thus the combination of Sinclair, Son, Yu and Jung renders this limitation obvious.

(c) 9[c]

As described for Ground III, Claim 2[b], the combined system of Sinclair, Yu and Son includes a controller with logic to “responsively retrieve data from a source location in the flash memory,” based on an address provided by the host [***“to identify both of the specific one of the subdivisions and the write destination from the read address”***].

Sinclair describes “read operations” where the controller is configured to identify and “read[] data from the memory system,” based on a host request [***“and the memory controller further comprises logic to receive, from the host, and via the host interface”***]. EX1007, 32:11-26.

Sinclair teaches that, when a host issues a read command including logical addresses, the *controller converts “those logical addresses” included in the read command* using “the logical to-physical address table ... *to identify the [] physical block to which the data at those logical addresses are stored*” [“the write destination from the read address”]. EX1007, 13:16-33.

Additionally, as described in Ground II, Claim 9[c], Son teaches this limitation.

The combined system of Sinclair, Yu, Son and Jung renders this limitation obvious. EX1003, ¶¶289-293.

3. Claims 17 and 19

Petitioner relies on the same disclosures and reasoning discussed with respect to the Corresponding Claim Limitations, above, to show that claims 17 and 19 are rendered obvious by the prior art. Specifically, Petitioner contends the same disclosure that renders obvious a Corresponding Claim Limitation (set out below) also renders obvious the limitations of claims 17 and 19.

Claim Limitation	Corresponding Claim Limitation(s)
17	6
19	9

XII. 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	John T. Mckee (Reg. No. 65,926) johnmckee@quinnemanuel.com

& SULLIVAN, LLP 295 Fifth Avenue, 9th Floor New York, NY 10016 Tel: (212) 849-7000 Fax: (212) 849-7100	QUINN EMANUEL URQUHART & SULLIVAN, LLP 51 Madison Avenue, 22nd Floor New York, NY 10010 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.

XIII. CONCLUSION

For these reasons, Petitioner requests *inter partes* review of the '183 patent.

Date: July 9, 2025

By: /s/ James Glass

James M. Glass (Reg. No. 46,729)
 QUINN EMANUEL URQUHART &
 SULLIVAN LLP
 295 5th Ave.
 New York, New York 10016
 Email: jimglass@quinnemanuel.com
 Phone: 212-849-7142

Fax: 212-849-7100

Counsel for Petitioners

CERTIFICATE OF LENGTH (37 C.F.R. §§ 42.24(D))

The undersigned hereby certifies that, according to the word-processing system used to prepare the foregoing document, this document has 13,790 words and thus complies with the applicable word limit.

Date: July 9, 2025

By: /s/ James Glass

James M. Glass (Reg. No. 46,729)
QUINN EMANUEL URQUHART &
SULLIVAN LLP
295 5th Ave.
New York, New York 10016
Email: jimglass@quinnemanuel.com
Phone: 212-849-7142
Fax: 212-849-7100

Counsel for Petitioners

CERTIFICATE OF SERVICE (37 C.F.R. §§ 42.6(E), 42.105(A))

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 '183 patent as shown in USPTO PAIR via FedEx:

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

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

Date: July 9, 2025

By: /s/ James Glass
James M. Glass (Reg. No. 46,729)
QUINN EMANUEL URQUHART &
SULLIVAN LLP
295 5th Ave.
New York, New York 10016
Email: jimglass@quinnemanuel.com
Phone: 212-849-7142
Fax: 212-849-7100

Counsel for Petitioners