IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of Robert C. Chang *et al.*:

U.S. Patent No. 6,831,865

Issued: December 14, 2004

Title: Maintaining Erase Counts in
Non-Volatile Storage Systems

Petition for *Inter Partes* Review

Attorney Docket No.:
337722-000080.865

Customer No.: 26379

Petitioner: Apple Inc.
Real Party-in-Interest: Apple Inc.

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,831,865

Mail Stop Patent Board
Patent Trial and Appeal Board
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Pursuant to the provisions of 35 U.S.C. §§ 311-319, Apple Inc. (hereinafter "Petitioner") hereby petitions the Patent Trial and Appeal Board to institute an *inter partes* review of claims 1-5, 8, 18, and 24-29 of United States Patent No. 6,831,865 (Ex. 1001).

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

Exhibit Number	**Description**
Ex. 1001	U.S. Patent No. 6,831,865 to Chang et al. ("the '865 Patent")
Ex. 1002	Prosecution File History For U.S. Patent No. 6,831,865 to Chang et al.
Ex. 1003	Declaration of Dr. R. Jacob Baker in Support of Petition
Ex. 1004	U.S. Patent No. 5,341,339 to Wells
Ex. 1005	U.S. Patent No. 6,151,246 to So
Ex. 1006	U.S. Patent No. 6,396,744 to Wong
Ex. 1007	U.S. Patent No. 5,838,614 to Estakhri
Ex. 1008	Excerpts from *Designing with FLASH MEMORY: The Definitive Guide to Designing Flash Memory Hardware and Software for Components and PCMCIA Cards* by Brian Dipert & Markus Levy Annabooks (1993, 1994)
Ex. 1009	U.S. Patent No. 5,485,595 to Assar
Ex. 1010	PC Card Standard, Volume 7, Media Storage Formats Specification
Ex. 1011	*A Floating Gate and Its Application to Memory Devices*, D. Kahng and S. M. Sze
Ex. 1012	*New Ultra High Density EPROM and Flash EEPROM with NAND Structure Cell*, Fujio Masuoka *et. al*.

Ex. 1013	Toshiba Web at flash25.toshiba.com
Ex. 1014	U.S. Patent No. 5,418,752 to Harari
Ex. 1015	U.S. Patent No. 6,362,049 to Cagnina
Ex. 1016	Intel FDI User Manual
Ex. 1017	*Cleaning Policies in Mobile Computers Using Flash Memory*, M.-L. Chiang & R.-C. Chang
Ex. 1018	*Managing Flash Memory in Personal Communication Devices*, Mei-Ling Chiang *et. al*.
Ex. 1019	U.S. Patent No. 5,740,395 to Wells
Ex. 1020	U.S. Patent No. 5,940,861 to Brown
Ex. 1021	U.S. Patent No. 7,012,835 to Gonzalez
Ex. 1022	U.S. Publication No. 2003/0046487 to Swaminathan
Ex. 1023	Intel Series 2 Flash Memory Cards Data Sheet
Ex. 1024	*Curriculum Vitae* of Dr. R. Jacob Baker
Ex. 1025	Declaration of Dr. David Hinds in Support of Petition
Ex. 1026	SUPPORTED.CARDS file within published pcmcia-cs-3.1.21.tar
Ex. 1027	CHANGES file within published pcmcia-cs-3.1.21.tar
Ex. 1028	PCMCIA Programmer's Guide within published pcmcia-cs-3.1.21.tar
Ex. 1029	PCMCIA How-To within published pcmcia-cs-3.1.21.tar
Ex. 1030	ftl.h within published pcmcia-cs-3.1.21.tar
Ex. 1031	ftl_cs.c within published pcmcia-cs-3.1.21.tar

Ex. 1032	ftl_check.c within published pcmcia-cs-3.1.21.tar
Ex. 1033	ftl_format.c within published pcmcia-cs-3.1.21.tar
Ex. 1034	iflash.h within published pcmcia-cs-3.1.21.tar
Ex. 1035	iflash2_mtd.c within published pcmcia-cs-3.1.21.tar
Ex. 1036	iflash2+_mtd.c within published pcmcia-cs-3.1.21.tar
Ex. 1037	Excerpts from *Nonvolatile Semiconductor Memory Technology: A Comprehensive Guide to Understanding and Using NVSM Devices*, by William D. Brown and Joe E. Brewer, ed., IEEE Press (1998).
Ex. 1038	U.S. Patent No. 6,381,176 to Kim
Ex. 1039	U.S. Patent No. 6,427,186 to Lin
Ex. 1040	Exhibit A to Joint Claim Construction and Pre-Hearing Statement

I. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), the real party-in-interest is Apple Inc. ("Petitioner").

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), Petitioner states that Longitude Flash Memory Systems S.A.R.L. ("Patent Owner") is asserting U.S. Patent 6,831,865 (the "'865 patent") against the Real Party-In-Interest in a suit filed September 23, 2014, styled *Longitude Licensing Ltd., and Longitude Flash Memory Systems S.A.R.L. v. Apple Inc.*, Case No. 3:14-cv-4275, pending in the United States District Court for the Northern District of California (the "Related Litigation").

Petitioner has filed, or soon will file, petitions for inter partes review of U.S. Patent Nos. 6,510,488; 6,763,424; 6,968,421; 7,012,835; 7,120,729; 7,224,607; 7,181,611; 7,657,702; 7,818,490; 7,970,987; 8,050,095; and 8,316,177.

As of the filing of this petition, no other judicial or administrative matters are known to Petitioner that would affect, or be affected by, a decision in an *inter partes* review of the '865 patent.

C. Lead and Back-up Counsel

Lead counsel for this matter is Brent Yamashita (USPTO Reg. No. 53,808), and back-up counsel for this matter is Edward Sikorski (USPTO Reg. No. 39478), both at the e-mail address: Apple-Longitude-IPR@dlapiper.com. The postal and hand delivery address for both is DLA Piper LLP (US), 2000 University Avenue, East Palo Alto, California, 94303, and the telephone and fax numbers are (650) 833-2348 (for phone) and (650) 687-1206 (for fax).

D. Service Information

Pursuant to 37 C.F.R. § 42.8(b)(4), papers concerning this matter should be served on the following email address: Apple-Longitude-IPR@dlapiper.com.

II. GROUNDS FOR STANDING

Pursuant to 37 CFR § 42.104(a), Petitioner certifies that the ’865 patent is available for *inter partes* review, and Petitioner is not estopped or barred from requesting *inter partes* review challenging the ’865 patent on the grounds identified in this petition.

III. RELIEF REQUESTED

Petitioner asks that the Board review the accompanying prior art and analysis, institute a trial for *inter partes* review of claims 1-5, 8, 18, and 24-29 of the ’865 patent, and cancel claims 1-5, 8, 18, and 24-29 as invalid for the reasons set forth below.

IV. THE REASONS FOR THE REQUESTED RELIEF

The full statement of the reasons for relief requested is as follows:

A. Summary of Reasons

- **Challenge #1:** Claims 1-3, 5, 18, and 25-27 of the ’865 patent are anticipated by U.S. Patent No. 5,485,595 (“Assar”).
- **Challenge #2**: Claims 8, 24, 29 are obvious over Assar in view of the knowledge of a person of ordinary skill in the art.
- **Challenge #3:** Claims 4, 5, and 28 are rendered obvious by Assar in view of U.S. Patent No. 5,838,614 (“Estakhri”).
- **Challenge #4:** Claim 18 is anticipated by U.S. Patent No. 6,427,186 to Lin (“Lin”).
- **Challenge #5:** Claim 24 is rendered obvious by Lin in view of the knowledge of a person of ordinary skill in the art.
- **Challenge #6:** Claims 1-3, 8, 25-27 and 29 of the ’865 patent are rendered obvious by U.S. Patent No. 6,381,176 to Kim (“Kim”) alone, or in view of pcmcia-cs-3.1.21.tar (“the Linux Publication”) and Volume 7, Media Storage Formats Specification to the PC Card Standard (“PC Card standard”)
- **Challenge #7**: Claims 4, 5, 24 and 28 are rendered obvious by Kim in view of the Linux Publication and the PC Card Standard, and further in view of Lin

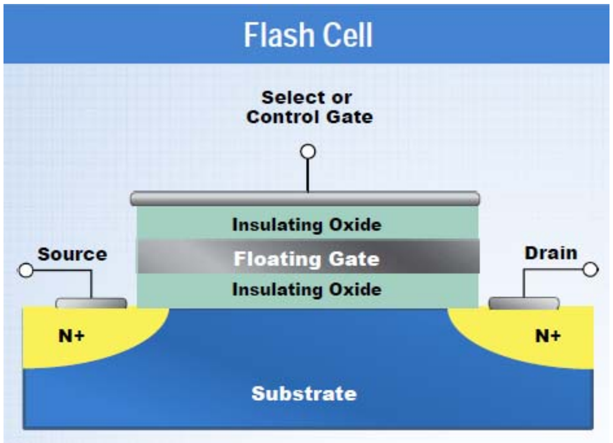
B. Relevant Background Technology

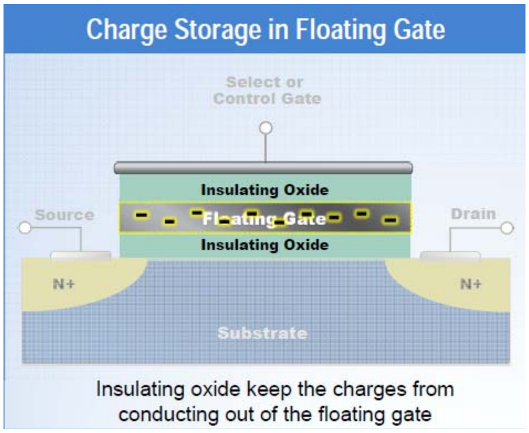
1. Overview of Flash Memory

Flash memory is a type of solid state semiconductor non-volatile memory. These devices are now ubiquitous in consumer electronic devices as data storage devices, including as a replacement for magnetic disk drives even in desktop computers. Ex. 1003, Declaration of Dr. Jacob Baker ("Baker Decl.") at ¶15.

Flash memory typically comprises an array of flash memory cells organized in rows and columns, as in conventional memory systems (such as DRAM or SRAM). Ex. 1003 at ¶¶ 24, 32, 33. Each flash memory cell utilizes a floating gate within a field effect transistor ("FET") to store electrical charge. Ex. 1003 at ¶ 19.

Shown below is an illustration of a typical flash memory cell with a floating gate added to a standard FET structure. Ex. 1003 at ¶¶ 20, 21.





Insulating oxide keep the charges from conducting out of the floating gate

The amount of electrical charge stored in the floating gate can be used to represent data bits ("1" or "0"). Ex. 1003 at ¶¶ 21,22. Since the "floating gate" is electrically insulated from the terminals of the FET, charge cannot readily conduct into or out of the floating gate, which allows long-term storage of the charge even when power is removed from the device. Ex. 1003 at ¶ 19. However, by applying a carefully controlled and sufficiently high voltage across the appropriate terminals, charge can be added to ("programmed") or removed from ("erased") the floating gate. *See e.g.* Ex. 1008 at 27, 28, 33, 34, 36.

Two common types of flash memory that were already commercially available by the early 1990s are NOR flash and NAND flash. Ex. 1003 at ¶ 31. They are so named because of the way that individual cells are electrically connected to form a memory cell array. In general, NAND flash memory has higher storage density compared to NOR flash, and are more advantageous in some storage applications. *Id.* at ¶¶ 34-35. However, the two types of flash both use the floating gate FET structure as the individual memory cell for charge storage. *Id.* at ¶ 31. Accordingly, problems that generally affect floating gate cells (to be discussed below) are known to affect both NOR- and NAND-type flash devices.

2. Failure Mechanisms in Flash Memory

When floating gate cells are used in real-world devices, electrons are moved back and forth across the floating gate's oxide region as the memory undergo numerous program and erase cycles ("P/E cycles"). These operations create stress on the oxide layer, and eventually cause the oxide to break down. When the oxide breaks down, the cell short circuits and becomes unusable. Ex. 1008 at 40. This failure mechanism is known as "oxide breakdown." Ex. 1003 at ¶¶ 36-37. In addition, electrons migrating through can also become trapped in the oxide layer due to "electron trapup," rendering the cell unusable for any practical purposes. Ex. 1003 at ¶ 38. By 2002, both oxide breakdown and electron trapup were well known to affect all floating gate type memory cells, and were thus extensively studied. Ex. 1037 at 69. *See also id.* at 130-144. It was well known that both NAND and NOR flash were susceptible to these failure modes. Ex. 1003 at ¶ 36.

3. Managing Flash Memory Failures

In real-world devices, given the inevitable failure of these cells, well-designed flash memory systems must manage such failures accordingly. Indeed, it was known that some flash memory devices were actually shipped with pre-existing defects in some of the cells. This is not surprising because real-world manufacturing processes generally do not yield completely defect-free devices. Ex. 1003 at ¶ 41. In practice, quality control testing may be performed by the manufacturer at the factory to identify the locations of these defective blocks. Manufacturers would thereafter program a special data bit pattern into the flash memory itself in order to identify these defects so that the users of the product could avoid trying to program into these defective blocks. Ex. 1003 at ¶ 43.

In addition to managing these manufacturing defects, techniques were also developed to manage defects that develop during normal usage (*e.g.* due to oxide breakdown or electron trapup). Known commercial implementations of flash memory devices at the time generally verify the nominal program/erase operations to ensure the program or erase operation succeeded. If the verification fails after exceeding a set time limit, the cells are recognized by the system as "bad." Ex. 1008 at 41, 140-143, 154. By tagging the cells as "bad," the memory system can avoid reusing these defective cells. Ex. 1003 at ¶ 39.

To mitigate these failures that result from normal use, it is important to equalize as much as possible the programming and erasing activity across all memory cells in the flash array so that certain memory cells do not fail prematurely from excessive concentrated use. This process is known as "wear leveling," and was already well known and commonplace by 2002. Ex. 1008 at 263-265. Ex. 1003 at ¶¶ 44-46. This often involved copying data from one block (the "source" block) to another block (the "destination" block) to balance the usage among the blocks. Ex. 1003 at ¶¶ 45-46. Balancing usage generally involved some comparison of the erase counts between the erase blocks in the device. Ex. 1003 at ¶¶ 44-46. Therefore, it was widely known to preserve and keep track of erase count information for memory blocks in erasable flash memory. Ex. 1003 at ¶ 46.

4. Organization of Data and Metadata Within Flash Memory

Digital data comprises a string of 1's and 0's, and therefore, data stored in such storage media must be done in an organized manner (*e.g.* a "storage format") so that they can be retrieved and parsed in a meaningful way. This is not unique to flash memory media, and for example, in magnetic disk drives, it was well known to structure the stored data in an appropriate format (*e.g.* by using the file allocation table "FAT" for DOS operating systems). The significance of data structures in general can be readily appreciated by recalling that in the early 1990s a DOS (or Windows) formatted disk was incompatible with Macintosh computers. The entire disk is essentially inaccessible and useless under the wrong format.

By extension, it was also well known to maintain data structures about the flash memory array itself, like the locations of free blocks and any defective blocks. ("Whatever the type of flash file system, there will be varying amounts of data structures stored on the flash memory in addition to the user's data."). *See* Ex. 1008 at 269-70. Other important flash media parameters such as memory capacity, block and sector sizes and configurations, as well as other operational parameters and media management metadata discussed above such as erase counts and defect mappings, etc., likewise were organized in data structures within the flash memory itself so they can be retrieved. Ex. 1003 at ¶ 65.

In order to facilitate use of flash memory in computer systems as a modular replacement of traditional hard disk drives, methods were developed to provide an additional layer of software between the physical flash memory and the computer systems that would control the memory. *See* Ex. 1010 at 1. This layer could essentially allow the host system to interact with both traditional hard disk drives and flash memory using the same set of commands without having to accommodate for the unique requirements of flash at the system level. *See* Ex. 1010 at 1. As even a layperson can appreciate from the Macintosh vs. DOS disk format example above, a standardized data structure format would ensure the greatest compatibility across devices and host systems. *See* Ex. 1008 at 276-277. The result of one such standardization effort occurred in the early 1990s within an industry group known as the Personal Computer Memory Card International Association (PCMCIA).

5. Overview of the PCMCIA Standard : FTL Specification

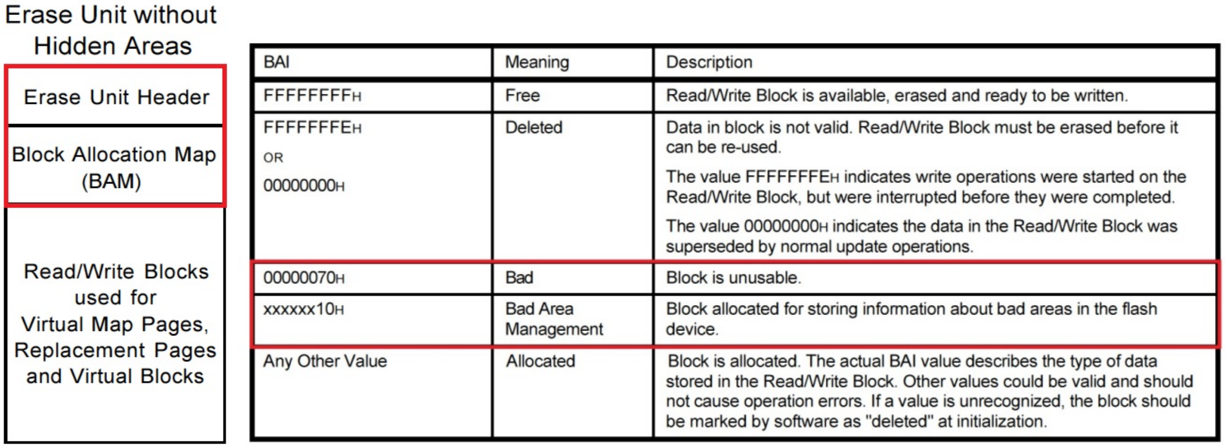
The PCMCIA formed in early 1990 to standardize specifications for peripheral cards (which became known as PCMCIA Cards, also commonly referred to as PC Cards). The PCMCIA consisted of several hundred companies that developed and maintained standards for these devices, and the first version (PCMCIA 1.0) was published in 1990. The standard is comprehensive, and specifies many aspects of these peripheral cards such as physical dimensions, electrical connections and other such characteristics to ensure compatibility. This effort allowed users to connect a wide variety of peripheral cards such as network cards, modems, and flash memory cards, to their personal computers. *See, e.g.,* Ex. 1008 at 187-189, 276-277.

Of specific relevance to flash memory, the PCMCIA also approved the Flash Translation Layer (FTL) specification around 1994, which became a heavily-used interface between the file system and the storage media. Persons of ordinary skill in the art would have been generally familiar with the concepts and terminologies used in this standard. Ex. 1003 at ¶73. Exhibit 1010 is the Media Storage Formats Specification (Volume 7) from the PC Card Standard Release 7.0 (1999), referred to herein as the "PC Card Standard."

The PC Card Standard establishes a data format for PC Cards that allows those cards to be used on different host systems, emulating a traditional block device. Ex. 1010 at 1, 24. In order to manage data blocks, FTL organizes the memory into "partitions" (a concept that is similar to traditional hard disks) comprising multiple "erase units," *i.e.* the smallest unit of flash memory that can be individually erased (referred to as "blocks" in the '865 patent). Ex. 1003 at ¶70. FTL keeps track of certain data about a given partition by including an Erase Unit Header (EUH) and a Block Allocation Map (BAM) in each erase unit. See, e.g., Ex. 1010 at 27. The formatted partition "uses a well-defined header to allow directory functions and file access to be performed across a wide variety of operating environments and host platforms." *Id.* at 19.

The header is important as it maintains global information about the partition in which the erase unit belongs. "The Erase Unit Header contains information specific to the Erase Unit and global information about the entire FTL partition." Ex. 1010 at 34. It includes information such as the size of the erase unit, the number of erase units in the partition, and the formatted size of the partition. *Id.* at 34-37. Skilled artisans recognized the importance of such global information within this header, as it is redundantly stored in every erase unit within the partition, allowing the system to readily locate any EUH, and thus retrieve the "self-management" information about the flash memory partition. *Id.* at 34, 38.

The standard also specifies a Block Allocation Map (BAM), adjacent to the EUH header. The BAM is a sequence of 4-byte values (called Block Allocation Information, or BAI) each of which tracks a corresponding block in the data erase unit using a virtual address. Ex. 1010 at 28-30. The BAM may hold information about "bad" blocks on the memory, as shown in the figures reproduced from Ex. 1010, with annotations added below:


Erase Unit without Hidden Areas
Erase Unit Header
Block Allocation Map (BAM)
Read/Write Blocks used for Virtual Map Pages, Replacement Pages and Virtual Blocks

BAI	Meaning	Description
FFFFFFFF$_H$	Free	Read/Write Block is available, erased and ready to be written.
FFFFFFFE$_H$ OR 00000000$_H$	Deleted	Data in block is not valid. Read/Write Block must be erased before it can be re-used. The value FFFFFFFE$_H$ indicates write operations were started on the Read/Write Block, but were interrupted before they were completed. The value 00000000$_H$ indicates the data in the Read/Write Block was superseded by normal update operations.
00000070$_H$	Bad	Block is unusable.
xxxxxx10$_H$	Bad Area Management	Block allocated for storing information about bad areas in the flash device.
Any Other Value	Allocated	Block is allocated. The actual BAI value describes the type of data stored in the Read/Write Block. Other values could be valid and should not cause operation errors. If a value is unrecognized, the block should be marked by software as "deleted" at initialization.

Ex. 1010 at Figure 5-3 (excerpt, red lines added), 29.

It is unsurprising that the PC Card standard expressly contemplates including means for managing bad blocks or areas within the flash memory, given the prevalence of defects as explained above.

6. Overview of the ’865 Patent

The ’865 patent, titled “Maintaining Erase Counts in Non-Volatile Storage Systems,” was filed October 28, 2002 and issued December 14, 2004 to inventors Robert C. Chang, Bahman Qwami, and Farshid Sabet-Sharghi. Ex. 1001.

The ’865 patent generally discloses a system that facilitates performing of wear leveling operations in a flash memory storage system. *See id.* at 3:19-26. The ’865 patent describes a wear leveling operation based on “a counter which keeps track of how many times a block has been erased may be maintained an incremented each time the block is erased,” *i.e.* erase counts. *Id.* at 9:39-42. Of relevance to the challenged claims, the ’865 patent discloses to store erase counts and other information about the blocks in the non-volatile memory, such as indications that certain blocks are unusable due to factory defects or growing defects. *Id.* at 3:57-60. Information may be stored in tables or arrays, or in an exemplary embodiment, an “erase count

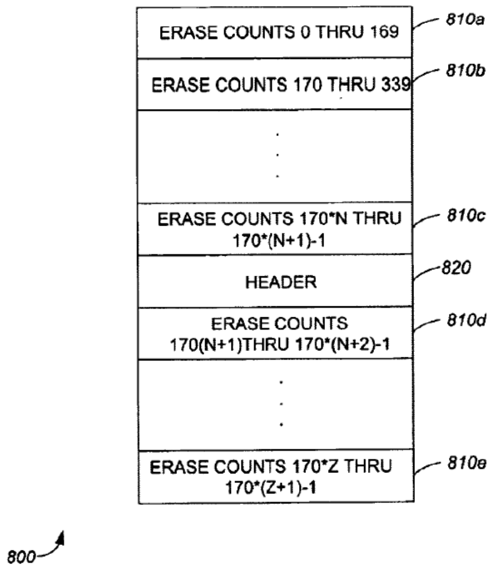

ERASE COUNTS 0 THRU 169
810a
ERASE COUNTS 170 THRU 339
810b
ERASE COUNTS 170*N THRU 170*(N+1)-1
810c
HEADER
820
ERASE COUNTS 170(N+1)THRU 170*(N+2)-1
810d
ERASE COUNTS 170*Z THRU 170*(Z+1)-1
810e
800

Fig. 8a

block" ("ECB"). *Id.* at 3:29-31, 5:52-6:9, Figs. 8a-d. The ECB is depicted in Fig. 8a, reproduced on the right:

The '865 patent generally discloses wear leveling operations that utilize the stored erase count information. For example, a data block may be identified and exchanged (or "swapped") with one of the least frequently erased blocks based on information stored in the table. *Id.* at 15:55-16:11 and at Fig. 6. However, none of the claims are specifically directed to the actual wear leveling operation or algorithm, and the claims simply require these various indicators be stored in memory.

C. Level of Ordinary Skill in the Art

A person of ordinary skill in the art ("POSITA") is a hypothetical person who is presumed to have known the relevant art at the time of the alleged invention. *Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed. Cir. 1986). Petitioner submits that a person of ordinary skill in the art ("POSITA") at the time of the '865 patent would have a Bachelor of Science degree in electrical engineering, computer science, computer engineering, or related field, and at least two years of experience working in the field of semiconductor memory design, or equivalent. Ex. 1003 at ¶ 77.

D. Claim Construction under 37 C.F.R. § 42.104(b)(3)

Pursuant to 37 C.F.R. §§ 42.100(b) and 42.204(b)(2), this petition presents claim analysis construing claim language such that it is "given its broadest reasonable construction in light of the specification of the patent in which it appears." Proposed claim constructions contained below are presented using the broadest reasonable interpretation standard, which is applied solely for the purposes of *inter partes* review. Because the standard for claim construction at the PTO is different than that used in litigation, *see In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364, 1369 (Fed. Cir. 2004); MPEP § 2111, Petitioner expressly reserves the right to argue in litigation constructions for any term in the '865 patent, as appropriate to that proceeding. Petitioner further notes that in the Related Litigation, Patent Owner takes the position that with the exception of a single means plus function limitation, none of the claim terms in the challenged claims of the '865 patent require construction and each should be given its plain and ordinary meaning. Moreover, Patent Owner expressly disputes that the preambles of the challenged independent claims 1, 18 and 25 are limiting. Petitioner concedes that Patent Owner's positions on claim construction are broader than Petitioner's construction. Therefore, to the extent the Board deems any of Patent Owner's positions to be reasonable, then Patent Owner's views should be adopted in this proceeding under the broadest reasonable interpretation ("BRI") standard.

1. “data structure” (claims 1-5, 8 and 25-29)

In the Related Litigation, Patent Owner contends that the preambles of the challenged claims are not limiting. To the extent the Board disagrees with Patent Owner, then “data structure” is a limitation, and each of the limitations recited in a given claim needs to be found in the same “data structure” to fall within the scope of the claim. Data structure is a term of art that is known and understood, and examples of data structures include records, arrays, tables, linked lists, and the “struct” structure as used in *e.g.* C programming language. Ex. 1003 at ¶83. In the Related Litigation, Patent Owner has not offered a construction for this term. Ex. 1040 at 41.

2. “means for indicating in the system memory a number of times each usable block included in the plurality of blocks has been erased” (claim 18)

Petitioner and Patent Owner agree in the Related Litigation that this term is governed by Pre-AIA 35 U.S.C. § 112 ¶ 6, but disagree on the construction of the recited function and the corresponding structure. Patent Owner takes the position that the recited function is “indicating in the system memory a number of times each usable block included in the plurality of blocks has been erased,” while Petitioner argues the recited function should be construed to mean “storing a separate erase count in the system memory for each usable block in the plurality of blocks.” Ex. 1040 at 45. The recited “each usable block in the plurality of blocks” references by antecedent basis an earlier open-ended limitation of "the non-volatile memory including a plurality of blocks." Under the BRI standard, since any plurality of blocks within a non-volatile memory system can be identified in analyzing the scope of the claim, a reasonable interpretation of the modifier "each" in "each usable block" should be that a separate erase count is stored for each usable block identified in the plurality of blocks. Otherwise, the modifier "each" can be rendered superfluous by excluding all the usable blocks that do not have a corresponding erase count in system memory. However, if the Board finds the Patent Owner's position to be reasonable under BRI, then its construction should be adopted.

A number of embodiments are identified by both parties in the Related Litigation as corresponding structure. These include the ECB 800 in Fig. 8a, and a page within an ECB (element 810a in Figs. 8b, 8c or 8d). The parties at least agree that any of these could be corresponding structure. Petitioner submits that the '865 patent also discloses that "[s]ystem memory 454 . . . holds a least frequently erased block table 466 and a most frequently erased block table 470." Ex. 1001 at 14:37-38. Either one of these tables also qualify as corresponding structure, because when considering under the BRI the "plurality of blocks" as only the blocks that have been erased least (or most) frequently, then the least (most) frequently erased table indeed stores a separate erase count in the system memory for each usable block in the identified plurality of blocks. *See* Fig. 5a, annotated below where "EC" is the corresponding erase count.

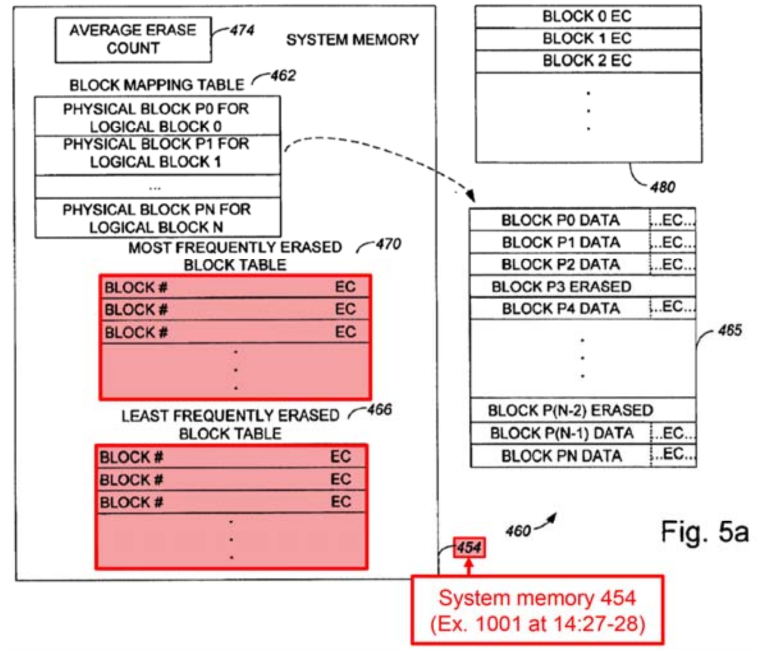


Fig. 5a

Patent Owner does not identify these tables to be corresponding structure, though it is noted that Patent Owner also identifies "Erase Counts" as a possible "corresponding structure," and these tables do include "erase counts" that "indicate in the system memory a number of times each usable block has been erased" under the BRI standard.

E. Challenge #1: U.S. Patent No. 5,485,595 ("Assar") Anticipates Claims 1-3, 5, 18, and 25- 27.

1. Overview of Assar

U.S. Patent No. 5,485,595 (Ex. 1009), titled "Flash Memory Mass Storage Architecture Incorporating Wear Leveling Technique Without Using Cam Cells," was filed on October 4, 1993 and issued to Assar et al. on January 16, 1996. Assar was filed as a continuation-in-part to an application that matured into U.S. Patent

No. 5,388,083 ("Assar '083"). Assar qualifies as prior art to the '865 patent under at least pre-AIA 35 U.S.C. § 102(a), (b), and (e).

Assar, which issued over six years before the '865 patent was filed, describes the use of erase counts and flags to indicate defective blocks:

> [A] circuit and method are provided for evenly using all blocks in [a semiconductor mass storage device]. . . In particular, flags are provided for defective blocks, used blocks, old versions of a block, a count to determine the number of times a block has been erased and written[,] and an erase inhibit flag.

Id. at Abstract (emphasis added).

Specifically, Assar describes a data structure in the form of a "table" of "information blocks," wherein each information block corresponds to a data block of the flash memory. Ex. 1009 at 10:55-66. These are illustrated in Figures 9 (the information table 610, highlighted in blue):

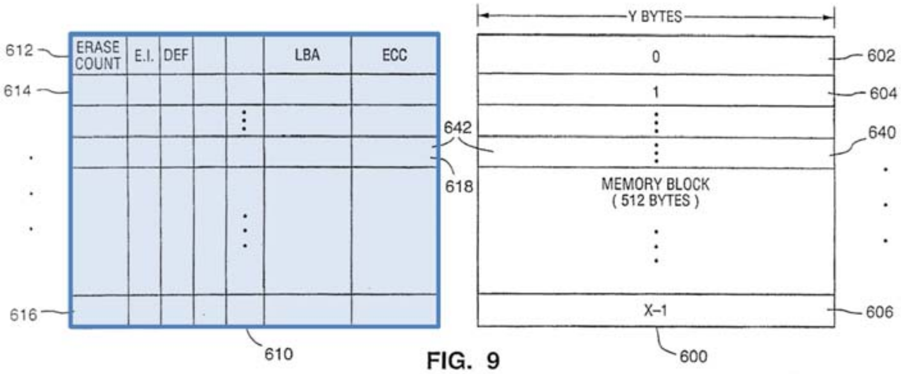


FIG. 9

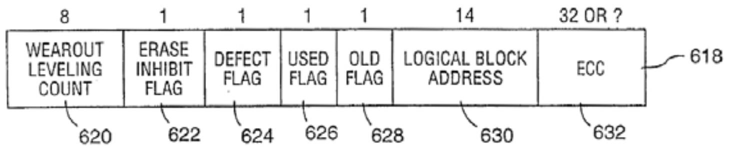


FIG. 10

Figure 10 (above) illustrates the fields of an information block (*i.e.* a row of the table), including an erase count, as well as other information such as a defect flag, used flag, old flag, logical block address and error correction codes ("ECC"). *See id.* at 11:7-9, 11:17-50. As with the '865 patent, Assar uses the information stored in the table for several customary flash memory operations, including for wear leveling. *See e.g.* 11:56-12:3 (programming data); 12:48-58 (reading data); 12:14-17 (erasing blocks); and 12:23-34 (wear leveling).

In an alternative embodiment, as illustrated in Figures 14 and 15, the table is maintained in both the flash memory and in system memory. Ex. 1009 at 14:35-45; 15:1-5 ("The volatile information block **710** and nonvolatile information block **711** contain identical information so long as power is applied to the system… Upon a power-up, or system reset, the information stored in each nonvolatile information block 711 is copied to each volatile information block 710.").

2. Assar anticipates independent claim 1

a. **Claim 1 preamble:** A data structure, the data structure being arranged in a non-volatile memory associated with a non-volatile memory system, the non-volatile memory system including a non-volatile memory which includes a plurality of blocks, the data structure comprising:

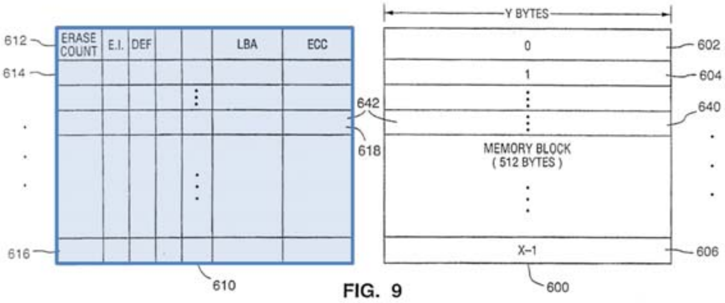


FIG. 9

Patent Owner contends the preamble is not limiting. To the extent the preamble is limiting, Assar discloses the preamble. *See* Ex. 1009 at Fig. 9, annotated on the right, showing the data structure (table 610) highlighted in blue and the plurality of flash memory blocks 602, 604, … 606 in non-volatile memory 600. Assar discloses a data structure in the form of a table of information blocks that each corresponds to a data block of a solid state storage memory. Ex. 1009 at 10:55-66. *See also* Ex. 1009 at 9:1-8; 9:15-16; 10:55-59; 10:66-11:33, showing the table is arranged in the non-volatile memory. *See also* alternative embodiment of FIGS. 14 and 15 described at 14:16-55. *See also* Ex. 1003 at pp. 74-77 (Table 1, claim 1.[Pre]).

b. Claim 1a: a first indicator, the first indicator being arranged to provide an indication of a number of times a first block of the plurality of blocks has been erased;

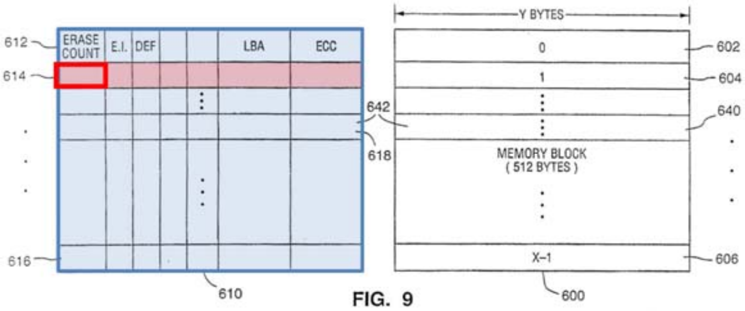


FIG. 9

Assar discloses indicators being arranged to provide an indication of a number of times a first block of the plurality of blocks has been erased. As shown in Fig. 9 on the right, each of the information blocks (*e.g.* a magenta shaded row) within the data structure 610 (highlighted in blue), includes an erase count (outlined in red) for a corresponding "data block" (*e.g.* elements 602, 604, … 606).

This is shown more clearly in Fig. 10, which labels the fields within a representative information block . Each information block includes a wearout leveling count 620 ("a first indicator," highlighted in red below) containing the number of times the corresponding data block ("a first block…") has been erased. Ex. 1009 at 11:7-9. *See also id.* at11:17-18 ("In particular, there is provided an eight bit wear out leveling counter 620…");12:14-17 ("As part of the erase operation, the wear out leveling count 620 for each erase data block 640 and information block 618 will be incremented."). *See also* Ex. 1003 at pp. 77-81 (Table 1, claim 1.[a]).

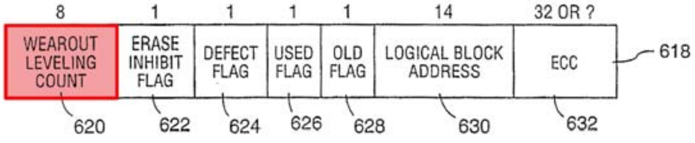
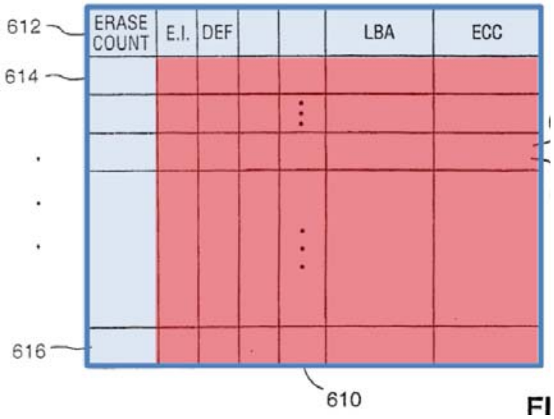


FIG. 10

c. **Claim 1b:** and a header, the header being arranged to contain information relating to the plurality of blocks.

Assar discloses a header being arranged to contain information relating to the plurality of blocks. According to the '865 patent, and as expressly claimed, the claimed header is arranged to contain information relating to the plurality of blocks. Ex, 1001 at 3:35- 38. *See also id.* at 18:65-19:4, stating that the header "may generally be located substantially anywhere." *See also* Ex. 1003 at ¶¶ 86-87.



Assar discloses such a claimed "header" with "information associated with each data block" of the memory within the table. Ex. 1009 at 11:7-11. Such information may include, for example, flags for defective data blocks (*id.* at 11:20-25); flags to mark whether memory is used, free, or old (*id.* at 11:27-29); logical block addresses (*id.* at 11:34-36); and error correcting codes (*id.* at 45-46); *see also* Figs. 9-10, 14-15. In the annotated excerpt of Fig. 9 below, the magenta shaded region of the table of information blocks includes such information and therefore includes the claimed header. *See also* Ex. 1003 at pp. 81-83 (Table 1, claim 1.[b]).

3. Assar anticipates dependent claim 2.

> **Claim 2**: The data structure of claim 1 further including a second indicator, the second indicator being arranged to provide an indication of a number of times a second block of the plurality of blocks has been erased.

Assar discloses a second indicator in the information table data structure that indicates the number of times a second block of the plurality of blocks has been erased. As discussed above, Figure 9 depicts a data structure, *i.e.* information block table, comprising multiple information blocks that each includes information corresponding to a data block in the non-volatile memory. *See, e.g.*, Ex. 1009 at

11:7-9; 3:24-26; 6:53-59; 11:17-20; 12:21-23. Because each information block contains an erase count for its corresponding data block in the non-volatile memory, the full table contains multiple indicators arranged to provide indications of erase counts—one for each data block mapped from the table. *See id.* at Fig. 9. *See also* Ex. 1003 at pp. 83-85 (Table 1, claim 2).

4. Assar anticipates dependent claim 3.

> **Claim 3:** The data structure of claim 1 further including a second indicator, the second indicator being arranged to provide an indication that a second block of the plurality of blocks is an unusable block.

Assar discloses an additional field in the information block table that indicates when blocks are unusable. Referring again to Figures 9 and 10, Assar discloses a "defect flag" in element 624 of Figure 10, highlighted in magenta below. When a defect flag is set, it indicates the block is unusable. Ex. 1009 at 6:19-21 ("In the event that a bit fails to be erased or programmed properly, a defect flag 118 in the CAM 106 is set preventing that block from being used again"); 11:20-23 ("A one bit defect flag 624 corresponds precisely to the defect flag 118 of FIG. 2").

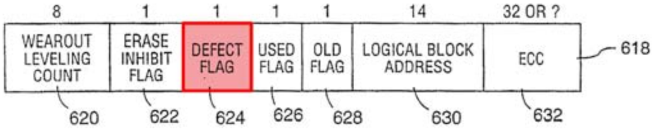


FIG. 10

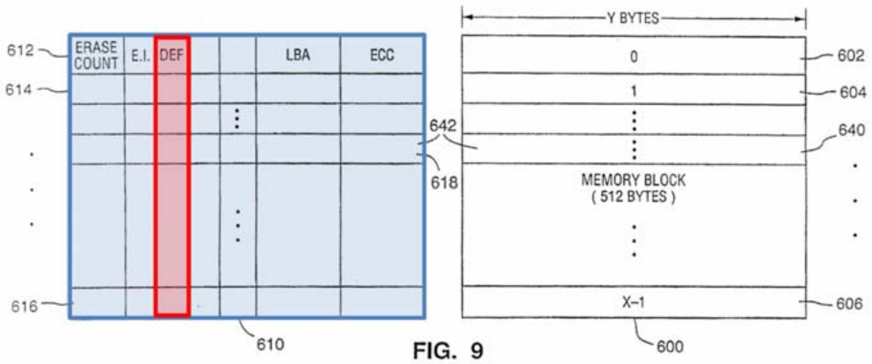


FIG. 9

Moreover, Figure 9 (annotated above) shows that there is at least a *second* indicator of a block being unusable, given the <u>numerous</u> defect flags (outlined in red below) in the information block table. *See also* Ex. 1003 at pp. 85-87 (Table 1, claim 3).

5. Assar anticipates dependent claim 5.

> **Claim 5:** The data structure of claim 3 wherein the second indicator is arranged to indicate that the second block has a growing defect.

A POSITA would recognize that the defect flag in Assar is an indicator to indicate a "growing defect," *i.e.* defects that arise during use and grow over time. *See* Ex. 1003 at ¶40. Here, Assar discloses that when a verification procedure fails, a defect flag is set to prevent the block from being used again. ("In the event that a bit fails to be erased or programmed properly, a defect flag 118 in the CAM 106 is set preventing that block from being used again.") Ex. 1009 at 6:13-21; *see also id.* at 11:20-26. The attempt to write to these memory cells indicates that before this failure, the cells were not recognized as defective, and the defect developed during the normal operation of the device. *See also* Ex. 1003 at pp. 87-88 (Table 1, claim 5).

6. Assar anticipates independent claim 18.

a. **Claim 18 preamble:** A non-volatile memory system comprising:

Patent Owner contends the preamble is not limiting. To the extent the preamble is limiting, Assar discloses a non-volatile memory system. *See* discussion in Section IV.E.2.a; Ex. 1009 at 10:55-66; 1:15-18; 9:12-13; Figs. 9 – 10; *see also* Ex. 1003 at p. 92 (Table 1, claim 18.[Pre]).

b. **Claim 18a:** a non-volatile memory, the non-volatile memory including a plurality of blocks;

As discussed above in Section IV.E.2.a, Assar discloses a non-volatile memory including a plurality of blocks. Ex. 1009 at 10:55-66; 2:66-3:1; Figs. 9 – 10. *See also* Ex. 1003 at pp. 92-93 (Table 1, claim 18.[a]).

c. **Claim 18b:** a system memory; and means for indicating in the system memory a number of times each usable block included in the plurality of blocks has been erased.

Assar discloses a system memory that includes an erase count for the blocks. In the alternative embodiment shown in Figures 14 and 15, the information table, including the erase counts corresponding to each of the plurality of blocks, are stored concurrently in both the non-volatile memory and in volatile system memory. Ex. 1009 at 15:1-3 ("The volatile information block **710** and nonvolatile information block **711** contain identical information so long as power is applied to the system"). *Id.* at 14:43-45. ("This volatile storage is formed of RAM cells. The RAM cells are preferably SRAMs, but DRAMs may also be used.") *See also id.* at 14:35-45. *See also id.* at 13:63-14:15, discussing certain advantages of using RAM-type memories in a separate system for storing the information blocks.

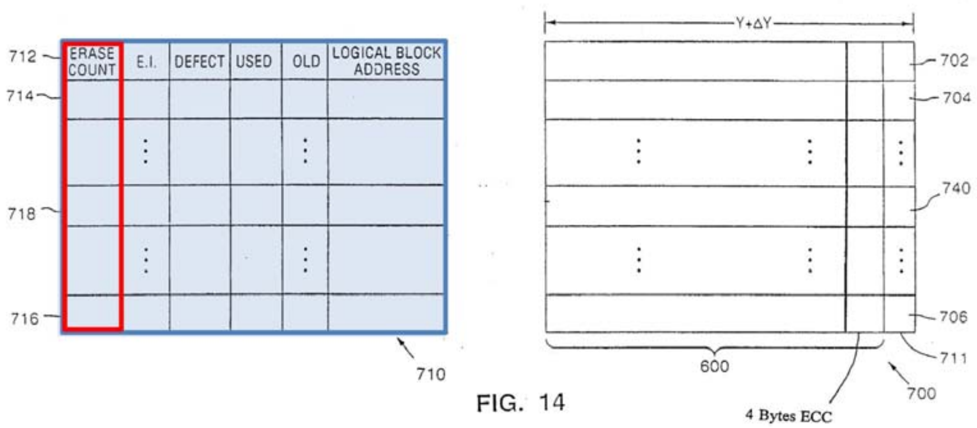


FIG. 14

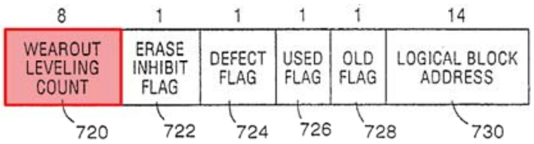


FIG. 15

As shown in the annotations above, the blue information block table (710) in volatile memory also includes an array of erase counts, outlined in red in FIG. 14 and individually highlighted in magenta in FIG. 15 above. These erase counts are identical to the ones described earlier with respect to the embodiments of FIGs. 9 and 10. *See* Ex. 1009 at 14:48-55 (the erase count 720 in FIG. 15 corresponds to erase count 620 in Figure 10); 12:14-17; 11:17-19; 6:53-59. This array of separate erase counts are stored in system memory for each usable block in the non-volatile memory system, and therefore performs the recited function using the same structure as construed. *See* Ex. 1003 at pp. 93-96 (Table 1, claim 18b).

7. Assar anticipates independent claim 25.

a. Claim 25 preamble: A data structure, the data structure being arranged in a physical block of non-volatile memory associated with a non-volatile memory system, the non-volatile memory system including a non-volatile memory which includes a plurality of blocks, the data structure comprising:

Patent Owner contends the preamble is not limiting. To the extent the preamble is limiting, Assar discloses the preamble. *See also* discussion regarding the preamble of claims 1 and 18 in Sections IV.E.2.a and IV.E.6.a. In another embodiment, Assar discloses a data structure in non-volatile memory (information table 711), that is concurrently stored in system memory (block 710), comprising a plurality of blocks, as illustrated below in annotated Figure 14 of Assar.

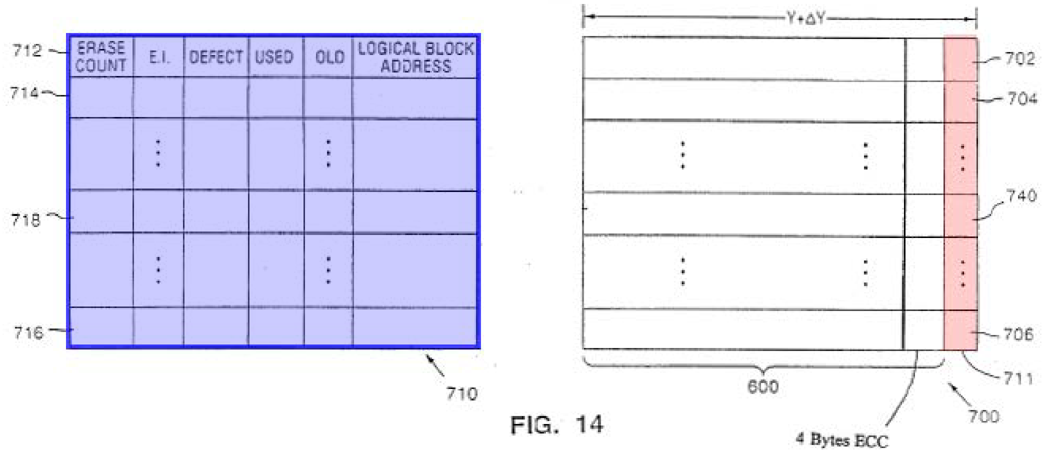


FIG. 14

In the figure above, the table highlighted in blue is also arranged in non-volatile memory highlighted in magenta and labeled element 711. Ex. 1009 at 15:1-3 ("The volatile information block 710 and nonvolatile information block 711 contain identical information so long as power is applied to the system.") Fig. 14. *See also* Ex. 1003 at pp. 100-102 (Table 1, claim 25.[Pre]).

b. Claim 25a: a first plurality of indicators, the first plurality of indicators being arranged to provide indications of numbers of times blocks included in the plurality of blocks have been erased;

Assar discloses indicators being arranged to provide an indication of a number of times blocks in the plurality of blocks have been erased. Ex. 1009 at 14:48-55. For ease of illustration, the erase counts from table 710 (in system memory) are annotated in red outline in FIG. 14 below, with the understanding that table 710 is redundantly stored in the non-volatile memory in element 711:

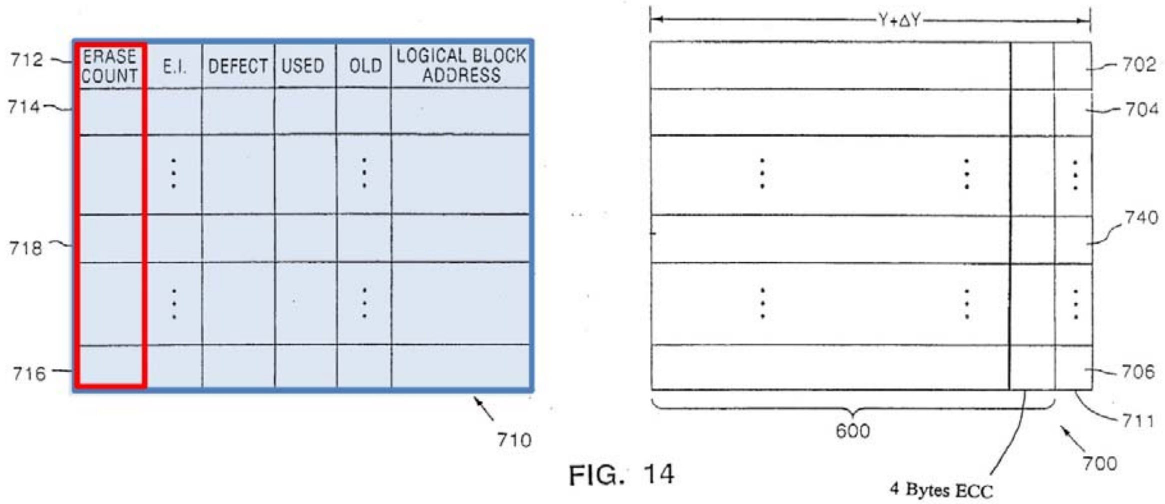


FIG. 14

See also Ex. 1003 at pp. 102-103 (Table 1, claim 25.[a]).

c. **Claim 25b:** a plurality of pages, the pages of the plurality of pages being substantially divided into groups of bytes arranged to contain the first plurality of indicators, wherein a first page of the plurality of pages includes a first group of the groups of bytes that is arranged to contain a first indicator of the first plurality of indicators which is associated with a first block of the plurality of blocks.

Assar discloses a plurality of pages that are arranged into groups of bytes arranged to contain the erase counts, where each erase count corresponds to a block of non-volatile memory. In a preferred embodiment of the'865 patent, pages are the same size as a 512 bytes sector (plus overhead data), and any number of pages may form a block. Ex. 1001 at 8:27-47. This is precisely what Assar discloses in Figure 14 (reproduced with annotations below):

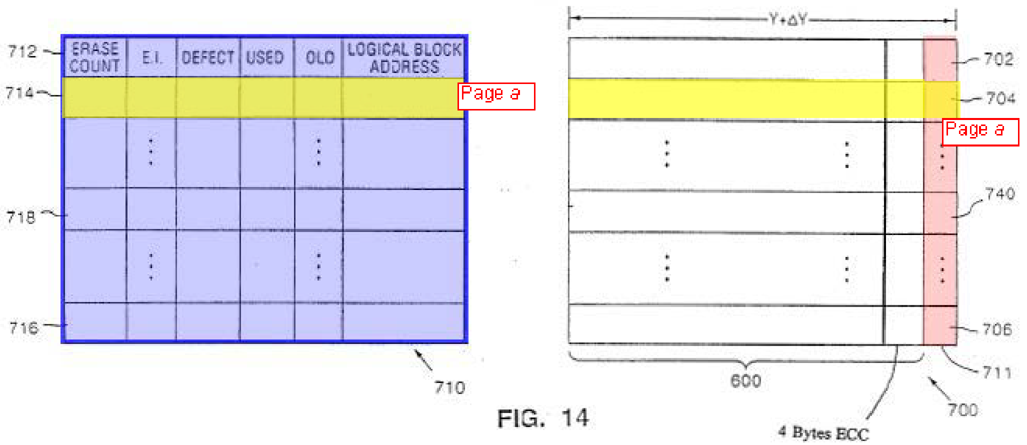


FIG. 14

Assar discloses pages (*e.g.* 702-706) that are 512 bytes wide sectors, with additional bytes for *e.g.* the defect flags and erase counts. Ex. 1009 at 14:23-26 ("Each data block 702 through 706 is Y+ΔY bytes long. The data blocks include

512 bytes for data storage and 8 bytes for storing the appropriate one of the information blocks flags, logical block address and error correction code.") Specifically, Figure 14 shows one page (704, highlighted in yellow) within the physical memory divided into Y and ΔY bytes that contains one row of the information block table 711 (highlighted in magenta). Ex. 1009 at 15:1-3 ("The volatile information block 710 and nonvolatile information block 711 contain identical information so long as power is applied to the system."). In other words, the pages disclosed by Assar are substantially divided into groups of bytes that are arranged to contain the plurality of indicators as this claim limitation requires. *See also* Ex. 1003 at pp. 103-110 (Table 1, claim 25.[b]).

8. Assar anticipates dependent claim 26.

> **Claim 26:** The data structure of claim 25 further including a second plurality of indicators, the second plurality of indicators being arranged to indicate when blocks in the plurality of blocks are substantially unusable.

As discussed above, Assar discloses an additional field in the information block table that indicates when blocks are unusable. In the information block of Figure 15, the defect flag is highlighted in magenta and labeled element 724.

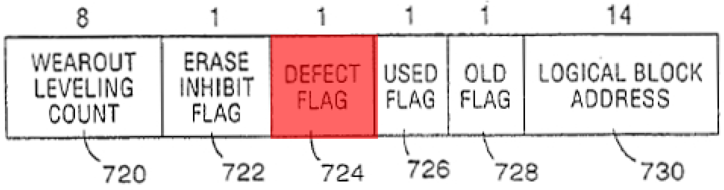


FIG. 15

When these information blocks are arranged in an information block table as annotated below in Figure 14 (and redundantly stored in non-volatile memory as Tabl 711), they contain a plurality of defect indicators that indicate blocks of the memory that are unusable. Ex. 1009 at 6:19-21; 11:20-26.

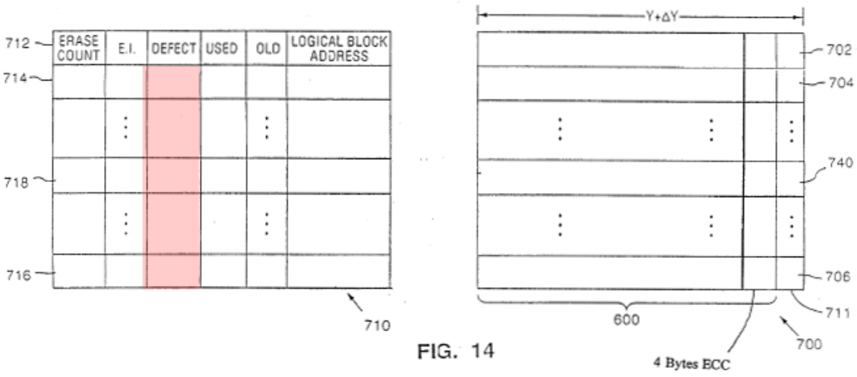


FIG. 14

See also Ex. 1003 at pp. 110-111 (Table 1, claim 26).

9. Assar anticipates claim 27.

> **Claim 27:** The data structure of claim 26 wherein the groups of bytes are further arranged to contain the second plurality of indicators.

As discussed with respect to claim 25b in Section IV.E.7.c, Assar discloses a plurality of pages that are further arranged into groups of bytes that contain erase counts. Ex. 1009 at 14:48-55. In the same way, the groups of bytes also contain the defect indicators as discussed in claim 26 in Section IV.E.8. *See also* Ex. 1003 at pp. 111-114 (Table 1, claim 27).

F. Challenge #2: Claims 8, 24 and 29 are obvious over Assar in view of the knowledge of a POSITA.

1. Assar, in combination with the knowledge of a POSITA

renders obvious claims 8, 24 and 29.

- **Claim 8:** The data structure of claim 1 wherein the non-volatile memory is a NAND flash memory.
- **Claim 24:** The non-volatile memory system of claim 18 wherein the non-volatile memory is a NAND flash memory.
- **Claim 29:** The data structure of claim 26 wherein the non-volatile memory is a NAND flash memory.

These dependent claims relate to NAND flash memory. As discussed with respect to claims 1, 18 and 26 from which these claims depend, Assar anticipates claims 1, 18 and 26. *See* Sections IV.E.2, IV.E.6 and IV.E.8.

Assar does not expressly disclose NAND flash, but does disclose managing flash memory in general using the table of information blocks. However, by 2002, it was well known that NAND-type flash was one of the most common types of flash memory. Ex. 1003 at ¶¶ 33-35. *See also* Ex. 1008 at 36. In addition, it was common knowledge that NAND-type memories have performance advantages for certain applications. Ex. 1003 at ¶ 35. For instance, it was known that NAND flash has higher storage density and much faster erase times, but slower random access read times. *Id. See also* Ex. 1008 at 32, 38 (Tables 3.1 and 3.3) comparing erase times for NOR at 1.6 sec (typ) vs. NAND at 6 ms. As described above, Assar stores at least erase counts and defect indicators within the table of information blocks for managing the flash memory system, including for dealing with memory cell failure. However, as explained in Section IV.B above, memory cell wear down is not unique or limited to any particular type of flash memory, but generally affects all floating gate type flash memory cells - including NAND flash. Ex. 1003 at ¶¶ 36-38. Therefore, to the extent Assar does not expressly disclose that the memory is NAND flash, a POSITA would have been motivated to apply the teachings in Assar to a NAND flash system, because it would be at least equally important to perform wear leveling and defect management in NAND flash memory as it is in the flash memory of Assar. Ex. 1003 at ¶¶ 36-38, 44. *See also id.* at pp. 88-92, 96-100, 114-118 (Table 1, claims 8, 24 and 29).

G. Challenge #3: Claims 4-5 and 28 are obvious over Assar in view of U.S. Patent No. 5,838,614 ("Estakhri").

1. Overview of Estakhri.

U.S. Patent No. 5,838,614 (Ex. 1007), titled "Identification and Verification of a Sector within a Block of Mass Storage Flash Memory," was filed on May 19, 1997 and issued to Estakhri et al. on November 17, 1998. Accordingly, it is prior art to the '865 patent under at least pre-AIA 35 U.S.C. § 102(a), (b), and (e).

Estakhri discloses a flash memory management system that avoids having to perform the undesirable erase-before-write when stored data files are modified or updated by using a non-in-place update scheme. *Id.* at 4:6-7. Ex. 1003 at ¶107. More generally, Estakhri is directed to a method and apparatus for efficiently identifying non-defective blocks in a flash memory. Ex. 1007 at Abstract. The memory system of Estakhri stores "information flags" for each nonvolatile memory location, including flags to identify old/new data, used/free blocks, and defective blocks. *Id.* at 6:7-10. In Estakhri's memory device, modified files may be written to the nonvolatile flash memory without first performing an erase. *Id.* at 6:35-52. Estakhri accomplishes this by designing a memory that searches for a free block, writing the modified file into the new block, storing the address of the new block in RAM, and marking the previous version of the file as "old." *Id.* If the system is unable to locate a free block, it erases the data for all non-defective blocks containing "old" data, and then identifies a free block from the set of blocks that were just erased. *Id.* at 8:12-20.

2. Motivation to Combine Assar and Estakhri.

A POSITA would have been motivated to combine the teachings of Assar and Estakhri to yield the subject matter of the '865 patent. Petro Estakhri is an inventor for both the Assar patent as well as the Estakhri patent. Like Assar, Estakhri relates to a nonvolatile flash memory that tracks defects. Ex. 1007 6:7-10. Estakhri also explicitly incorporates Assar's teachings by reference. Ex. 1007 at 3:24-27; 10:18-24. Therefore, Estakhri expressly directs a POSITA to combine its teachings with Assar's. Estakhri discloses "a novel way to use a defect flag for each block stored within the flash memory device for efficiently identifying non-defective blocks upon system power-up." *Id*. at Abstract. Accordingly, it would have been obvious for a POSITA to apply the general teachings from Assar, with the new teachings disclosed in Estakhri, to arrive at a flash memory device with the capability to track defects with finer granularity, such as tracking manufacturing defects as well as growing defects. Indeed, Estakhri expressly references the Assar scheme for the appropriate details related to the space manager block 544 responsible for tracking and avoiding defective blocks when searching for a free block to program. *See id*. at 10:10-12, 19:2-7; 12:48-57 . *See Ex.* 1003 at ¶¶ 97-103.

3. Assar and Estakhri render obvious Claims 4, 5, and 28.

- **Claim 4:** The data structure of claim 3 wherein the second indicator is arranged to indicate that the second block has a factory defect.

- **Claim 5:** The data structure of claim 3 wherein the second indicator is arranged to indicate that the second block has a growing defect.

- **Claim 28:** The data structure of claim 26 wherein the second plurality of indicators includes a second indicator which is arranged to identify when a second block of the plurality of blocks has a manufacturing defect and a third indicator which is arranged to identify when the second block has a growing defect.

These dependent claims further specify that the unusable block is a factory (or manufacturing) defect and/or a growing defect. It was well known that in 2002, manufacturers shipped and sold flash memory devices that contained defective cells. Ex. 1003 at ¶¶ 41-43. Estakhri expressly teaches that unusable blocks can arise from manufacturing defects, and to include indicators that indicate these defective blocks so they can be avoided in operation:

> During manufacturing of flash memory chips, defects within the memory are commonly identified and marked by the chip manufacturer. This is typically done by writing a predefined pattern (byte-wide) in a predetermined location within a defective block… If a sector or a cell within a block is defective, the manufacturer will set a manufacturing defect flag located somewhere within the defective block… to a predetermined value.

Ex. 1007 at 18:56-66, emphases added. Estakhri also expressly teaches that unusable blocks can arise from growing or grown defects:

> In the event that a bit fails to be erased or programmed properly, a defect flag 148 is set which prevent that block from being used again…
>
> Other than defects detected during manufacturing of flash chips, there may be additional defects developed during operation of the chips due to wearing as discussed earlier. These defects are sometimes referred to as 'grown defects' by the industry at-large and must be accounted for by a system in which using flash memory devices are employed.

Ex. 1007 at 8:42-44; 19:8-13, emphases added. *See also id.* at 8:36-44; 18:56-19:50. Estakhri discloses the common industry-standard practice of identifying the location of these factory and grown defect blocks, and further stresses the importance of having the flash controller (specifically, the "space manager" module as disclosed in Assar) track these blocks so as not to use them when searching for a free block. *Id.* at 10:10-35; 18:63-66, 19:2-7. Lastly, with respect to claim 28, Estakhri expressly discloses that the manufacturing defect flag ("second indicator") is distinct and in addition to the grown defect flag ("third indicator"), although both are tracked by the same space manager block. ("This manufacturing defect flag is not the same as defect flag 1012 …") *Id.* at 18:66-67; *see also* 18:66-19:7, 19:14-18, 19:31-35. A POSITA would have been motivated to modify the data structure of Assar to also include an indicator to indicate these factory/manufacturing defects, and an indicator to indicate these grown defects, such that these blocks are kept out of service when the memory is first initialized. *Id*. at 19:31-35. *See also* Ex. 1003 at pp. 125-127, 127-130, 130-135 (Table 2, claims 4, 5, and 28)

H. Challenge #4: U.S. Patent No. 6,427,186 ("Lin") anticipates Claim 18

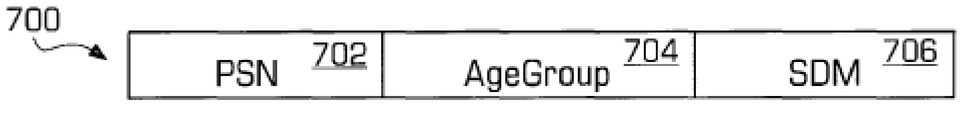
1. Overview of Lin

U.S. Patent No. 6,427,186 (Ex. 1039), titled "Memory, interface system and method for mapping logical block numbers to physical sector numbers in a flash memory, using a master index table and a table of physical sector numbers," was filed on March 30, 1999 and issued to Lin et al. on July 30, 2002. Accordingly, it is prior art to the '865 patent under at least pre-AIA 35 U.S.C. §102 (a) and (e).

Lin relates to defect management in flash memory devices, and describes improved methods to ensure that only non-defective blocks (termed “segments” in Lin) are allocated for programming in a “wear leveling manner.” Ex. 1039 at 1:9-13; 1:53-55; 2:64-67. Lin likewise discloses using indicators to address the problem of manufacture/factory as well as grown defects in flash memory, confirming Estakhri’s disclosure that this was common in the industry. *Id.* at 1:30-34; 5:4-21; 5:22-37; 5:56-6:13. Keeping track of these defects is important to ensure that the system would not attempt to allocate these defective cells for storing data.

In a further aspect, as depicted in Figure 7, Lin discloses a “free list table 700” that stores references to erased segments that are available for use. *Id.* at 8:1-3. Free list table 700 includes an “AgeGroup” field and a “SDM” (“segment defect map”) field that identifies defective memory cells. *Id.* at 8:3-4. The “AgeGroup” field contains that segment’s erase count. *Id.* at 8:9-11 (The AgeGroup field 704 contains all or part of AGE COUNT); *See also id.* at 4:45-53; 8:15-16.

FIG. 7



The AgeGroup (i.e., erase counts) are used to sort the free list table 700 in descending order from “youngest” to “oldest.” *Id*. at 8:17-21. Thus, when the next available block is chosen from the free list table 700, it will readily choose the one that was erased the fewest times, thereby achieving wear-leveling throughout the flash media. *Id*. at 8:24-27. Accordingly, Lin expressly discloses the advantages of tracking both defects and erase counts in order to implement a "wear-level" aware memory allocation scheme that also avoids defective cells.

2. Lin anticipates claim 18

a. Claim 18 preamble: A non-volatile memory system comprising:

To the extent the preamble of claim 18 is limiting, Lin discloses a non-volatile flash memory system. "FIG. 1 schematically illustrates a PC/CF (meaning either PCMCIA or Compact Flash) circuit card 100 having flash memory media 102 and a controller 104 for controlling access to the media." Ex. 1039 at 2:13-16. *See also* Fig. 1; Ex. 1003 at p. 242 (Table 5, claim 18.[pre]).

b. Claim 18a: a non-volatile memory, the non-volatile memory including a plurality of blocks;

Lin discloses a non-volatile memory comprising a plurality of blocks, which Lin calls segments. "The memory cells of the flash memory chips 106a through 106n are grouped into segments (sometimes called blocks), where segments are the base unit for the erase operation, meaning that all the memory cells in a particular segment must be erased at the same time." Ex. 1039 at 2:24-35. *See also* Ex. 1003 at p. 242 (Table 5, claim 18.[a]).

c. Claim 18b: a system memory; and means for indicating in the system memory a number of times

each usable block included in the plurality of blocks has been erased.

Lin describes a system memory in the form of a "local buffer 116" of Figure 1, highlighted in magenta below. "The local buffer serves as the variable and stack space of microcontroller 112 executing the firmware stored in the EEPROM 114, and can also be used for storing bookkeeping information." Ex. 1039 at 3:11-14, emphases added.

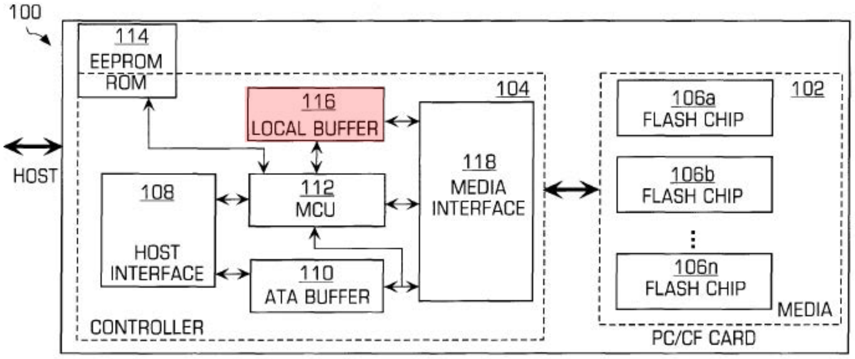
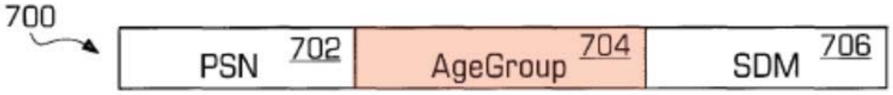


FIG. 1

The local buffer 116 contains a data structure ("free list table 700") that is used to track media usage, including erase count and defect information for free usable blocks (segment). Ex. 1039 at 2:7-10; 8:1-5. This data structure is illustrated in Figure 7, with AgeGroup (indicating erase counts) annotated below.

FIG. 7



Specifically, it stores a separate erase count (AgeGroup) for each free block (segment) that can be used for allocation by the system, which also indicates within the buffer the number of times each of the free blocks has been erased.

> "The free list table 700 is used to store an indication of erased segments that are candidates to be used… Each entry has a PSN field 02, an AgeGroup field 704… The table 700 is held in the local buffer 116. The PSN field is an indication of the first sector of the free segment… The AgeGroup field is part (or, in some embodiments, all) of AGE COUNT described earlier with reference to FIG. 4… The free list table 700 is organized in AgeGroup descending order…" *Id.* at 8:1-18.

> "When a segment is taken from the free list table 700 to store data, the segment is taken from the top of the free list table 700. Thus the free list table 700 indicates a collection of pre-erased segments… such that wear-levelling can be achieved throughout media 102." *Id.* at 8:24-27.

See also id. at 4:45-47 describing AGE COUNT. Accordingly, under either Patent Owner or Petitioner's construction, free list table 700 is corresponding structure that performs the recited function. *See* Ex. 1003 at pp. 242-245 (Table 5, claim 18.[b]).

I. Challenge #5: Claim 24 is obvious over Lin in view of the knowledge of a POSITA.

1. Claim 24: The non-volatile memory system of claim 18 wherein the non-volatile memory is a NAND flash memory.

As discussed in Section IV.F.1, a POSITA would have recognized the advantages of NAND flash memory for certain applications, and therefore would have been motivated to implement the Lin non-volatile memory system (including the use of system memory to store erase counts) with NAND flash memory. Therefore, the analysis in that section applies for Lin as well. *See also* Ex. 1003 at pp. 245-248 (Table 5, claim 24).

J. Challenge #6: Claims 1-3, 8, 18, 24- 27 and 29 are obvious over U.S. Patent No. 6,381,176 ("Kim") in view of the Linux Publication and the knowledge of POSITA

1. Overview of Kim

U.S. Patent No. 6,381,176 (Ex. 1038), titled "Method of Driving Remapping in Flash Memory and Flash Memory Architecture Suitable Therefor," was filed on May 15, 2001 and issued to Kim et al. on April 30, 2002. Kim qualifies as prior art to the '865 patent under at least pre-AIA 35 U.S.C. § 102 (a) and (e).

Kim was cited by the Examiner during the prosecution of the '865 patent. Ex. 1002 at 242-243 (Apr. 13, 2004 Office Action at 3-4). In response, Applicants argued that the element-at-issue in Kim was incorrectly identified by the Examiner as an "erase count," when in fact, it is a "wrap count." *Id.* at 290 (June 11, 2004 Remarks at 8). A Notice of Allowance was subsequently issued, without an expressed reason for allowance. *Id.* at 300 (Aug. 6, 2004 Notice of Allowance). However, Kim does in fact disclose erase counts, and the prosecution history record shows the Examiner may not have fully appreciated the proper context of some of the terms used in Kim, particularly with respect to the PC Card Standard.

See Ex. 1003 at ¶¶117, 120-125. This Petition therefore presents Kim in a new light, *i.e.* in view of the Linux and/or Lin implementations of the PC Card Standard, that was not previously considered by the Examiner during the original prosecution.

Kim describes the use of a data structure with erase counts and defect indicators for operating a NAND flash memory. Ex. 1038 at 2:64-67. Specifically, Kim describes the data structure within an "erase unit," which contains "wear level" counts (cnt and xcnt) as well as a "bad block" table (x/bb tbl). These are annotated in Figure 7 below:

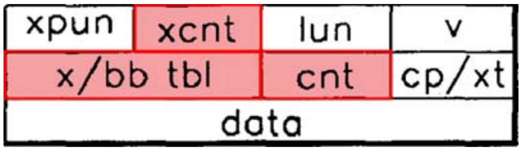


Figure 7 shows some of the data stored in a block of flash memory, including "the physical unit number of a previous unit xpun, a wear level of a reclaimed (previous) unit xcnt, a logical unit number lun, a valid flag v, a bad block table bb tbl, a wear level cnt, copying/transferring cp/xf, and data." Ex. 1038 at 7:53-57, emphases added. The wear level cnt indicates the number of erase cycles for that unit, which is increased by one each time the unit is erased. *See id.* at 3:27-30; 8:13-16. Each erase unit also includes a variety of information about the memory, such as a table of bad blocks, an Erase Unit Header (EUH), and a Block Allocation Map (BAM). *Id.* at 4:31-32; 8:66-9:1. A POSITA would have recognized the EUH and BAM as terms of art consistent with how they are used in the PC Card standard. Ex. 1003 at ¶¶124-125. Such data as organized allow the memory system to maintain information necessary for, *e.g.*, wear leveling and efficient data usage.

2. Overview of the Linux Publication

The Linux publication was authored by Dr. David Hinds, packaged into a single .tar file ("pcmcia-cs-3.1.21.tar") and published online and made available to the public on October 3, 2000. Ex. 1025 at ¶10. Therefore, the Linux Publication qualifies as prior art to the '865 patent under at least pre-AIA 35 U.S.C. §§ 102 (a) and (b). For the purpose of this *inter partes* review, Petitioner has submitted only selected excerpts from the .tar container file as separate Exhibits (Ex. 1026-1036), although a true and correct copy of the full .tar file can still be accessed publicly at the following URL: http://sourceforge.net/projects/pcmcia-cs/files/pcmcia-cs/3.1.21/. Ex. 1025 at ¶10. However, Petitioner submits that collectively, Exhibits 1026-1036 constitute excerpts from a single piece of prior art publication.

A few of these exhibits are discussed specifically below. In particular, Ex. 1026, Ex. 1030 and Ex. 1031 are true and correct copies of the files SUPPORTED.CARDS, ftl.h and ftl_cs.c, respectively, within the pcmcia-cs-3.1.21.tar file. Ex. 1025 at ¶13. Dr. Hinds' began working on earlier versions of this publication in as early as 1995. His goal was, in part, to enable products that comply with the "PCMCIA Standard" to work with computers that run the Linux operating system. *See* Ex. 1025 at ¶ 4. The Linux publication specifically discloses support for a wide variety of flash memory storage cards, such as Smartmedia flash, Compact flash cards, Intel's Series 2 Flash Memory Cards. *See* Ex. 1026 at lines 337-346, 465-468. *See also* Ex. 1023, a data sheet for the Intel Series 2 product.

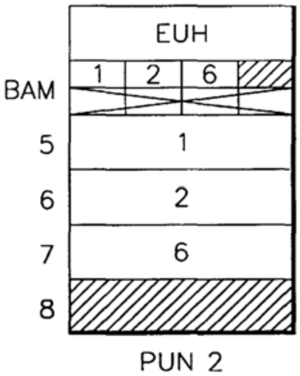
Dr. Hinds' publication describes a driver implementation that is compatible with the PC Card Standard (Ex. 1010), and utilizes the standardized Flash Translation Layer "FTL" data structures that allow a flash memory card to be used "as if it were an ordinary disk device." Ex. 1029 at 37. Ex. 1025 at ¶ 4. It also includes a wear leveling algorithm for managing these flash memory PC cards. Ex. 1025 at ¶ 7. The algorithm and associated structures are described within ftl.h (Ex. 1030) and ftl_cs.c (Ex. 1031) of the Linux Publication. Ex. 1031 and 1032 are source code files written in the C programming language, and contain plain English comments, in addition to human-readable code for implementing the Linux card services driver. Ex. 1003 at ¶ 137. A person of ordinary skill in the art, reading the Linux publication, would be able to appreciate the teachings of not only the comments written in plain English, but also the lines of C code themselves, and glean relevant disclosures and teachings regarding implementation of a wear leveling algorithm for flash memory devices. *Id.* Further, a person of ordinary skill in the art would also readily recognize the relationship between the Linux Publication and the PC Card Standard, and would possess at least some working familiarity with the PC Card Standard. *Id.* at ¶ 73, 138.

3. Independent claim 1 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

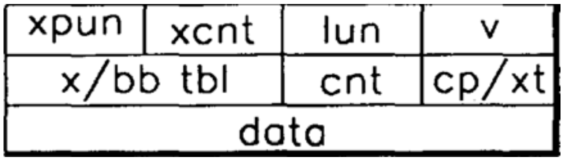
a. **Claim 1 preamble:** A data structure, the data structure being arranged in a non-volatile memory associated with a non-volatile memory system, the non-volatile memory system including a non-volatile memory which includes a plurality of blocks, the data structure comprising:

Patent Owner contends in the Related Litigation that the preamble is not limiting. To the extent the preamble is limiting, Kim discloses it. *See* Ex. 1038 at Figures 1-3, showing multiple erase units (*i.e.* "blocks") as PUN 1 and PUN 2. *See also id.* at 1:8-9; 5:39-43; claim 1. *See also* Ex. 1003 at ¶¶120-124.

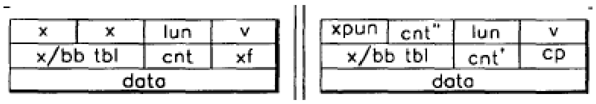
The Kim data structure is illustrated in Figs 3, 7 and excerpt of 8:



Ex. 1038 at Figure 3.



Ex. 1038 at Figure 7.



Ex. 1038 at Figure 8 (excerpt)

See also Ex. 1038 at 7:38-42, 7:51-66.

As illustrated, the data structure includes an EUH and a BAM, arranged in flash memory. *Id*. at Figure 3; 4:14-40. Though Figs. 7 and 8 do not expressly identify the EUH and BAM, a POSITA would understand the EUH and BAM to be included in those embodiments as well. *Id.* at 7:38-42, 7:64-66.

To the extent the Board finds that Kim does not expressly or inherently disclose a "data structure" that includes both the EUH and the BAM, the Linux Publication expressly discloses a data structure (a struct defined in C) named "partition_t" (defined at Ex. 1031 lines 154-180) that comprises both an EUH and BAM. *Id.* at lines 172, 175. Further, the partition_t struct includes an array of indicators for erase counts of all erase units in the partition. *Id.* at lines 162, 168.

```
154    typedef struct partition_t { …
160            struct eun_info_t { …
162            u_int  EraseCount; …
165            } *EUNInfo;
166            struct xfer_info_t { …
168            u_int  EraseCount; …
170            } *XferInfo;
171            u_short        bam_index
172            u_int          *bam_cache; ...
175            erase_unit_header_t        header; …
180    } partition_t;
```

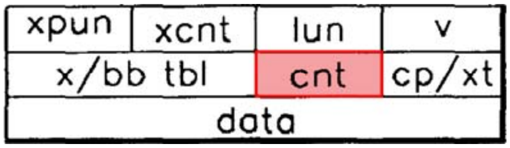
Excerpt from Ex. 1031. *See also* Ex. 1030 at line 37 within the definition of the erase_unit_header_t structure at lines 33-53.

A POSITA would have understood the general importance of organizing all the relevant bookkeeping information (such as the partition header information, erase counts of the erase units, and locations of bad blocks) for a partition in the same data structure for the purpose of managing the memory in the partition. For example, on a host write request, the FTL needs to ensure there is sufficient free space in the partition to accommodate the request. This requires knowledge of *e.g.* the number and size of blocks in the partition. If there is insufficient free space, blocks will need to be reclaimed as part of the allocation process – at a minimum, erase counts would therefore need to be updated accordingly (as disclosed in Kim, *see* Ex. 1038 at 3:14-29). The importance of such information is confirmed by noting the repeated reference to the partition_t data structure (generally given the name "part") within numerous function calls (algorithms) in the Linux Publication's implementation of the PC Card standard-compliant driver. *See* ftl_write() defined at Ex. 1031 lines 1279-1358, especially lines 1293-1295, 1311, 1321-1329, 1340; *see also* reclaim_block() at lines 880-966; find_free() at 968-1041. Because these algorithms require the use of this information, it is natural and obvious to organize this information into a data structure. *See* Ex. 1003 at ¶84;

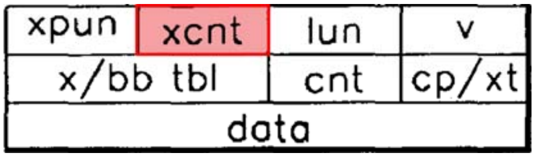
Id. at pp. 153-154. To the extent this is not already encompassed by the knowlegde of a POSITA, this is expressly disclosed in the Linux Publication. *See also* Ex. 1003 at pp. 148-154 (Table 3, claim 1.[Pre]).

b. Claim 1a: a first indicator, the first indicator being arranged to provide an indication of a number of times a first block of the plurality of blocks has been erased;

Kim discloses a first indicator, arranged to provide an indication of a number of times a first block of the plurality of blocks has been erased. For example, Kim discloses a wear level, stored in the EUH of each erase unit. "[I]nformation required for management of a corresponding unit are written in the EUH of the erase unit… may include a LUN, and a wear level." Ex. 1038 at 4:33-39, emphasis added. The wear level stores the number of erase cycles of a unit and increases by one when its corresponding block is erased. *Id.* at 3:27-29, 8:13-16. This is also shown as wear level "cnt" in Figure 7, below, and described at 7:56:



In another aspect, wear level "xcnt" is another indicator indicating the erase count for a different block. Ex. 1038 at 7:51-58.



See also Ex. 1003 at pp. 154-159 (Table 3, claim 1.[a]).

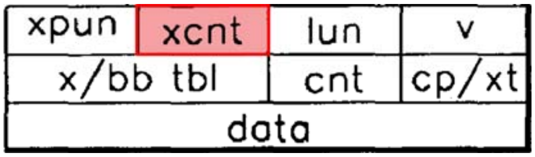
c. Claim 1b: and a header, the header being arranged to contain information relating to the plurality of blocks.

Kim discloses a header being arranged to contain information relating to the plurality of blocks. Kim describes an Erase Unit Header (EUH) within each erase unit, which contains "information about the whole flash memory." Ex. 1038 at 4:33. That information may include "indications of the size of the blocks in the erase unit and a bad block map." *Id.* at 4:33-40. The embodiment shown in Figures 7 and 8 also includes an EUH. Ex. 1038 at 7:65-66 ("the EUH is written over two blocks as shown in FIG. 8"). *See also* Ex. 1003 at pp. 159-165 (Table 3, claim 1.[b]).

4. Dependent claim 2 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

Claim 2: The data structure of claim 1 further including a second indicator, the second indicator being arranged to provide an indication of a number of times a second block of the plurality of blocks has been erased.

Kim discloses a second field in the data structure that indicates the number of times a second block of the plurality of blocks has been erased. Fig. 7 depicts a data structure with indicators for erase counts for two different erase units, with a first indicator being wear level "cnt" (as discussed above), and a second indicator "xcnt" (annotated in Fig. 7 to the right) representing the wear level of a reclaimed (previous) erase unit. Ex. 1038 at 7:51-58.



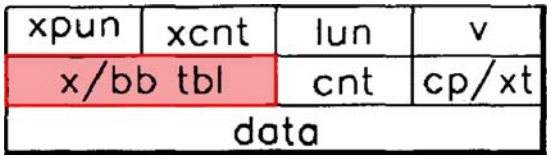
xpun	xcnt	lun	v
x/bb tbl		cnt	cp/xt
data			

To the extent the Board finds that Kim does not expressly disclose a “data structure” that includes all of the recited limitations of claim 2, *see* discussion regarding the “partition_t” data structure as disclosed in the Linux Publication in Section IV.J.3.a *supra*. *See also* Ex. 1003 at pp. 165-167 (Table 3, claim 3).

5. Dependent claim 3 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

> **Claim 3**: The data structure of claim 1 further including a second indicator, the second indicator being arranged to provide an indication that a second block of the plurality of blocks is an unusable block.

Kim discloses additional indicators that identify unusable erase units. The EUH of each erase unit contains a “bad block map.” Ex. 1038 at 4:35-37. Additionally, as discussed above, the data structure shown in Figure 7 contains a bad block table, which table in turn contains indicators arranged to provide an indication that an erase unit is unusable. *Id.* at 7:56; 8:66-67 and at Figure 7. *See also* Ex. 1010 at 28-29, where the PC Card standard expressly reserves fields for storing indications of unusable blocks.



xpun	xcnt	lun	v
x/bb tbl		cnt	cp/xt
data			

To the extent the Board finds that Kim does not expressly disclose the second indicator being in the same data structure as identified in claim 1, the Linux Publication teaches a single partition_t data structure that contains all such information about a partition. *See* discussion in Section IV.J.3.a. *See also* Ex. 1003 at pp. 167-173 (Table 3, claim 3).

6. Claim 8 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

> **Claim 8:** The data structure of claim 1 wherein the non-volatile memory is a NAND flash memory.

Kim discloses this limitation. The objective of the Kim invention is expressly "…suitable for a NAND-type flash memory." Ex. 1038 at 2:65-66. *See also id.* at 5:31-32. *See also* Ex. 1003 at pp. 173-174 (Table 3, claim 8).

7. Independent claim 25 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

a. Claim 25 preamble: A data structure, the data structure being arranged in a physical block of non-volatile memory associated with a non-volatile memory system, the non-volatile memory system including a non-volatile memory which includes a plurality of blocks, the data structure comprising:

Patent Owner contends in the Related Litigation that the preamble is not limiting. To the extent the preamble is limiting, Kim, in view of the Linux Publication, discloses it. *See* discussion in Sections IV.J.3.a *supra*. *See also* Ex. 1003 at pp. 174-175 (Table 3, claim 25.[Pre]).

b. Claim 25a: a first plurality of indicators, the first plurality of indicators being arranged to provide indications of numbers of times blocks included in the plurality of blocks have been erased;

As discussed above with respect to claim 1a and claim 2, Kim discloses indicators being arranged to provide an indication of a number of times a first block of the plurality of blocks has been erased, as well as indicators being arranged to provide an indication of a number of times a second block of the plurality of blocks has been erased. These indicators constitute a plurality of indicators arranged to provide indications of numbers of times blocks included in the plurality of blocks have been erased. *Id.* at Figure 7; 7:54-55; 8:13-16; 4:35-37; 3:27-29. *See also* discussion in Sections IV.J.3.b and IV.J.4 *supra*.

To the extent the Board finds that Kim does not expressly disclose a "data structure" that includes the plurality of indicators, *see* discussion regarding the "partition_t" data structure as disclosed in the Linux Publication in Section IV.J.3.a *supra*. *See also* Ex. 1003 at pp. 175-180 (Table 3, claim 25.[a]).

c. **Claim 25b:** a plurality of pages, the pages of the plurality of pages being substantially divided into groups of bytes arranged to contain the first plurality of indicators, wherein a first page of the plurality of pages includes a first group of the groups of bytes that is arranged to contain a first indicator of the first plurality of indicators which is associated with a first block of the plurality of blocks.

Kim discloses a plurality of pages that are arranged into groups of bytes arranged to contain the erase counts, where each erase count corresponds to a block of non-volatile memory. Specifically, Kim discloses a NAND-type flash memory. Ex. 1038 at 2:64-67. It is known to POSITA that NAND flash is divided into a plurality of pages. Ex. 1008 at 168. Kim also expressly discloses that the physical units ("PU") comprises a plurality of pages of memory cells. Ex. 1038 at 5:31-43. Note that Kim uses the term "block" to refer to a "page." ("the block is also called a page") Ex. 1038 at 5:37. *See also* Ex. 1003 at ¶¶120-125 for a discussion of how a POSITA would understand the terminology used in Kim in relation to the terms in the '865 patent.

Further, Kim discloses that these pages are divided into bytes, as is customary in the art. ("A block [*i.e.* page] in a flash memory is composed of bytes…") Ex. 1038 at 1:24-26. *See also* Ex. 1010 at 28 (noting that BAI is stored in groups of bytes), at 34-37 (showing that each field of the EUH is stored in a group of bytes). Each of the indicators identified in the discussion for limitation 25a (as discussed in relation to claims 1 and 2) reside in these Kim physical units comprising pages divided into bytes. *See* discussion in Sections IV.J.3 and IV.J.4 *supra*. *See also* Ex. 1003 at pp. 180-186 (Table 3, claim 25.[b]).

8. Dependent claim 26 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

> **Claim 26**: The data structure of claim 25 further including a second plurality of indicators, the second plurality of indicators being arranged to indicate when blocks in the plurality of blocks are substantially unusable.

As discussed above with respect to claim 3, Kim discloses an additional indicator ("bad block table") in the data structure shown in Figure 7 that indicates when blocks are substantially unusable. That data structure contains a plurality of indicators arranged to indicate when blocks in the plurality of blocks are substantially unusable. *See id.* at Figure 7; 4:35-37; 7:56; 8:66-67. *See also* Ex. 1010 at 28-29. *See also* Section IV.J.5 *supra*.

To the extent the Board finds that Kim does not expressly disclose a "data structure" that includes the first and second plurality of indicators, *see* discussion regarding the "partition_t" data structure as disclosed in the Linux Publication in Section IV.J.3.a *supra*. *See also* Ex. 1003 at pp. 186-191 (Table 3, claim 26).

9. Claim 27 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

> **Claim 27**: The data structure of claim 26 wherein the groups of bytes are further arranged to contain the second plurality of indicators.

As discussed with respect to claim 25b, Kim discloses a plurality of pages that are further arranged into groups of bytes that contain erase counts. *Id.* at Figures 1-3 and 7; 4:32-40; 7:56. In the same way, Kim discloses a group of bytes that contain the bad block indicators shown in Figure 7. *See also* Ex. 1003 at pp. 191-195 (Table 3, claim 27).

10. Dependent claim 29 is obvious over Kim in view of the Linux Publication and the knowledge of POSITA.

> **Claim 29:** The data structure of claim 26 wherein the non-volatile memory is a NAND flash memory.

Kim discloses this limitation. The objective of the Kim invention is expressly "…suitable for a NAND-type flash memory." Ex. 1038 at 2:65-66. *See also id.* at 5:31-32. *See also* Ex. 1003 at p. 195 (Table 3, claim 29).

K. Challenge #7: Claims 4, 5, 24 and 28 are obvious over Kim in view of Lin.

1. Kim and Lin render obvious Claims 4, 5, and 28.

- **Claim 4:** The data structure of claim 3 wherein the second indicator is arranged to indicate that the second block has a factory defect.
- **Claim 5:** The data structure of claim 3 wherein the second indicator is arranged to indicate that the second block has a growing defect.
- **Claim 28:** The data structure of claim 26 wherein the second plurality of indicators includes a second indicator which is arranged to identify when a second block of the plurality of blocks has a manufacturing defect and a third indicator which is arranged to identify when the second block has a growing defect.

These dependent claims further specify that the unusable block is a factory defect and/or a growing defect. To the extent a POSITA would not have understood the Kim disclosure of bad blocks to inherently indicate factory or growing defects, it would have been obvious to POSITA given the known state of the art regarding flash memory cells and defects. *See* Ex. 1003 at pp. 203-204, 212-213.

Moreover, these claims are obvious over Kim in view of Lin. Both Kim and Lin are directed to flash memory management operations, and more specifically directed to PC-Card Standard-compliant flash memory systems. *See* Ex. 1003 at ¶¶124, 146. Lin expressly teaches maintaining tables (data structures) to track unusable blocks that arise from manufacturing defects ("factory defects") as well as from defects grown during operation of the chips. Ex, 1039 at 4:54-6:13.

> "In general, memory cells of the media 102 may be defective as a result of either a manufacturing defect or as a result of simply wearing out. Those memory cells which are defective … are marked as defective by the manufacturer… Then, in operation, after certain erase-program cycles, more defective memory cells may result." Ex. 1039 at 5:4-15, emphasis added.

Lin expressly discloses the industry-standard practice of manufacturers identifying the location of these factory defect blocks.

> **"**In accordance with an embodiment, manufacturing defects are recorded in a manufacture defect list (MDL), constructed at the first low-level format of the media 102 (usually at the first power up of the card 100) …A hard defect table (HDT) is also maintained. The HDT originates from the MDL, and is a "working copy" for segment-level defect management." Ex. 1039 at 5:59-6:5, emphasis added.

Lin discloses that the MDL and HDT are stored in the flash memory media itself. *Id.* at 6:23-24. Lin also expressly discloses a segment defect map ("SDM") to track growing defects.

> "The SDM field 408 is used to hold a segment defect map, which is a one-byte bit map of the sectors within a segment… When a program failure in a particular sector occurs, the SDM field 408 is updated to indicate the newly-defective sector." Ex. 1039 at 4:54-55, 5:1-3, emphasis added. *See also* FIGs. 4, 5A and 5B.

The SDM is included in the first non-defective page of every erase unit. *Id.* at 2:30-39, 4:64-5:1. Additionally, Lin expressly notes that the manufacturing defect flag is not the same as the grown defect flag, and each type of defect is tracked in a different table. *Id.* at 5:4-13; 5:55-6:13.

Accordingly, to the extent the bad block identifiers in Kim do not expressly indicate a "factory defect" or a "growing defect," a POSITA would have been motivated to apply the teachings in Lin to specifically include indicators to indicate these factory and grown defects in the bad block table area in Fig. 7 of Kim, or in the blocks allocated for storing information about bad areas in accordance with the PC Card standard. Ex. 1010 at 29. Indeed, a POSITA would recognize that the standard expressly reserved these fields within the data structure for defect management, with special defined indicators for such indication.

Accordingly, in an exemplary aspect, combining the teachings of Lin and Kim would yield an erase unit, formatted according to the specifications of the PC Card Standard (*i.e.* with an EUH and BAM) and storing at least one of those defect tables as well as the segment defect map within the erase unit (with the appropriate page marked with the xxxxxx10H indicator in the BAM). That erase unit would therefore contain the header and other associated information of Kim as discussed with respect to claims 1, 2, 3, and 25 as well as the factory and/or growing defect indicators of claims 4, 5, and 28. *See also* Ex. 1003 at pp. 203-212, pp. 212-221, pp. 221-241(Table 4, claims 4, 5 and 28).

2. Kim and Lin render obvious Claims 24.

> **Claim 24:** The non-volatile memory system of claim 18 wherein the non-volatile memory is a NAND flash memory.

Petitioner submits that Lin in combination of the knowledge of a POSITA renders claim 24 obvious, as discussed in Section IV.I *supra*. To the extent the Board finds that the use of NAND flash memory is not within the general knowledge of a POSITA, this is expressly disclosed by Kim, as discussed with respect to claim 8 in Section IV.J.6 *supra.* It would have been obvious to modify the teachings of Lin regarding the use of the free list table 700 in system memory for a NAND flash memory system when the storage application calls for the use of NAND flash memory as discussed with respect to claim 24 in Section IV.I *supra*.

A POSITA would have been motivated to utilize the Lin free list table 700 in system memory in order to perform wear leveling. Ex. 1039 at 8:24-26. *See also* Ex. 1003 at pp. 245-248 (Table 5, claim 24).

V. CONCLUSION

Claims 1-5, 8, 18, and 24-29 of the ’865 patent are anticipated or rendered obvious by prior art discussed *supra*. There is a reasonable likelihood that Petitioner will prevail as to each of the claims. Petitioner respectfully requests that the Patent Office initiate an *inter partes* review of claims 1-5, 8, 18, and 24-29, that it find those claims invalid in light of the prior art, and that it cancel those claims.

Respectfully submitted,

Dated: September 21, 2015

/s/ Brent Yamashita

Brent Yamashita
Registration No. 53,808

DLA PIPER LLP (US)
2000 University Avenue
East Palo Alto, CA 94303-2214
Telephone: 650.833.2348
Facsimile: 650.833.2001

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. 37 C.F.R. §§ 42.6(e) and 42.105(b) on the Patent Owner by courier of a copy of this Petition for *Inter Partes* Review and supporting material at the following correspondence address of record for the ’865 Patent:

Brinks Gilson & Lione (BGL)
P.O. Box 10395
Chicago, IL 60610

Dated: September 21, 2015

/s/ Brent K. Yamashita
Brent K. Yamashita
Registration No. 53,808

DLA PIPER LLP (US)
2000 University Avenue
East Palo Alto, CA 94303-2215
Telephone: 650.833.2348
Facsimile: 650.687.1206