United States Patent No. 7,827,370

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SanDisk LLC
Petitioner
v.

Memory Technologies, LLC
Patent Owner

Patent No. 7,827,370

PETITION FOR *INTER PARTES* REVIEW UNDER 35 U.S.C. § 311, 37 C.F.R. §§ 42.100 *ET SEQ.*

TABLE OF CONTENTS

PETITIONER'S LIST OF EXHIBITS

Ex.	Description
1001	U.S. Patent No. 7,827,370
1002	File History for U.S. Patent No. 7,827,370
1003	U.S. Patent App. Pub. No. 2004/0083346 to Chevallier et al.
1004	U.S. Patent No. 6,279,114 to Toombs et al.
1005	U.S. Patent No. 6,262,918 to Estakhri et al.
1006	Declaration of Dr. R. Jacob Baker
1007	U.S. Patent App. No. 10/279,470 to Chevallier et al.

I. MANDATORY NOTICES, STANDING, AND FEES

Real Parties in Interest: The real parties in interest are: SanDisk LLC, Western Digital Corporation, Western Digital Technologies, Inc., SanDisk, Limited, SanDisk Storage Malaysia Sdn. Bhd., SanDisk Semiconductor (Shanghai) Co., Ltd., and SanDisk Israel (Tefen) Ltd. The following are direct or indirect parents or subsidiaries of the preceding companies: HGST, Inc., Virident Systems International Holdings Ltd., Western Digital International Ltd., SD International Holdings Ltd., SanDisk Technologies LLC, SanDisk International Holdco B.V., SanDisk IL Ltd., SanDisk Bermuda Limited, SanDisk Manufacturing Unlimited Company, SanDisk Bermuda Unlimited and SanDisk China Limited.

Related Matters: *Memory Technologies, LLC v. SanDisk LLC, et al.,* No. 8:16-cv-2163-JLS-DFM (C.D. Cal.).

Lead Counsel and Request for Authorization: Pursuant to 37 C.F.R. §§ 42.8(b)(3) and 42.10(a), Petitioner designates the following: Lead Counsel is Eliot D. Williams (Reg. No. 50,822) of Baker Botts L.L.P.; Back-up Counsel are Brian Oaks (Reg. No. 44,981), Chris Ryan (Reg. No. 54,759), and Jason German (Reg. No. 69,497) of Baker Botts L.L.P.

Service Information: Service information is as follows: Baker Botts L.L.P., 1001 Page Mill Road, Building One, Suite 200, Palo Alto, CA 94304; Tel. (650) 739-7500; Fax (650) 739-7699. Petitioner consents to service by electronic mail at eliot.williams@bakerbotts.com, brian.oaks@bakerbotts.com, chris.ryan@bakerbotts.com, and jason.german@bakerbotts.com. A Power of Attorney is filed concurrently herewith under 37 C.F.R. § 42.10(b).

Certification of Grounds for Standing: Petitioner certifies under 37 C.F.R. § 42.104(a) that the '370 Patent is available for *inter partes* review. Petitioner is not barred or estopped from requesting *inter partes* review of any claim of the '370 Patent on the grounds shown herein.

Fees: Under 37 C.F.R. § 42.103(a), the Office is authorized to charge the fee shown in 37 C.F.R. § 42.15(a) to Deposit Account No. 02-4377, Ref. No. 083480.0106, as well as any additional fees due in connection with this Petition.

II. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

Petitioner challenges claims 1-7, 12-19, and 25 of U.S. Patent No. 7,827,370 ("the '370 Patent"), assigned to the Patent Owner.

A. Patents and Publications Relied Upon

Exhibit 1003—United States Patent Application Publication No. 2004/0083346 to Chevallier et al. ("Chevallier"), entitled "Permanent Memory Block Protection in a Flash Memory Device," filed October 24, 2002 and published April 29, 2004. Chevallier is prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (e). Chevallier was not previously presented to the PTO in the context of the '370 Patent.

Exhibit 1004—United States Patent No. 6,279,114 to Toombs et al.

("Toombs"), entitled "Voltage Negotiation in a Single Host Multiple Cards System," filed November 4, 1998 and issued and published on August 21, 2001. Toombs is prior art under at least pre-AIA 35 U.S.C. §§ 102(a), (b), and (e). Toombs was cited to the PTO during prosecution of the '370 Patent.

Exhibit 1005—United States Patent No. 6,262,918 to Estakhri et al. ("Estakhri"), entitled "Space Management for Managing High Capacity Nonvolatile Memory," filed June 30, 2000 and issued and published on July 17, 2001. Estakhri is prior art under at least pre-AIA 35 U.S.C. §§ 102(a), (b), and (e). Estakhri was not previously presented to the PTO in the context of the '370 Patent.

B. Grounds for Challenge

Petitioner sets forth the following Grounds: **(1)** claims 1-3, 5-6, 12-17, and 25 are anticipated under 35 U.S.C. §§ 102(a) and (e) by Chevallier; **(2)** claims 1-3, 5-6, 12-17, and 25 are rendered obvious under 35 U.S.C. § 103 by Chevallier; **(3)** claims 1-7, 12-19, and 25 are rendered obvious under 35 U.S.C. § 103 by Chevallier in view of Toombs; and **(4)** claim 25 is rendered obvious under 35 U.S.C. § 103 by Chevallier in view of Toombs and Estakhri.

III. BACKGROUND OF THE TECHNOLOGY

Memory cards, such as PC cards, compact flash ("CF") cards, secure digital ("SD") cards, or multimedia cards ("MMC"), are electronic data storage devices used in various portable electronic devices such as digital cameras, mobile phones, laptop computers, tablets, and video game consoles. Ex. 1006 ¶74. Data is stored on a memory device by recording the data in bits in memory cells. *Id.* ¶75. This data can be read by sensing the values of the bits. *Id.*

Memory devices are typically based on a block architecture in which the memory is divided into blocks of memory. Ex. 1006 ¶87. This allows file systems to erase blocks of memory instead of the entire device. Ex. 1003 ¶0005. Memory sectors, memory blocks, and memory groups are units used to describe portions of a memory. Ex. 1004 at 27:37-42; FIG. 66.

MMCs can utilize one or more memory technologies such as ROM, OTP, MTP, or Flash. *Id.* 7:5-8, FIG. 4. An example MMC is shown in the figure below.

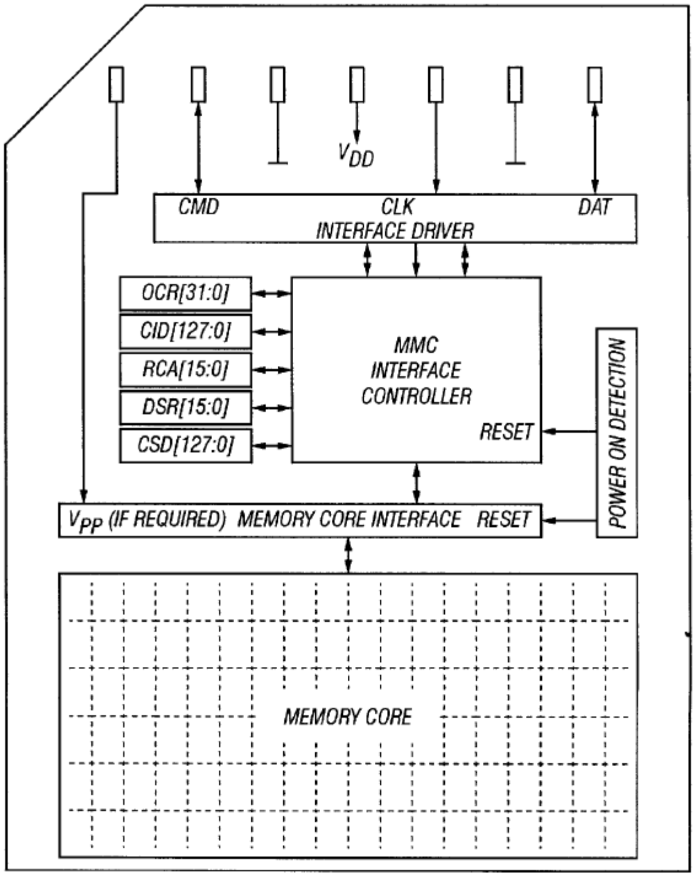


FIG. 14

Id. FIG. 14. As illustrated, a MMC communicates with a host device by a CMD bus line for commands and responses and a DAT bus line for transmission of data. *Id.* 7:57-65. The MMC includes a command set for controlling operations on the MMC such as data read/write or obtaining card information. *Id.* FIGs. 38-44. Each command is identified by a command number. For example, CMD 28 of Fig. 42 identifies the "SET_WRITE_PROT" command. *Id*. FIG. 42.

As illustrated above, the MMC includes an interface controller coupled to the MMC's memory core. Ex. 1006 ¶83. The interface controller also couples to a group of registers (*e.g.*, OCR, CID, CSD, RCA, DSR) that can store information about the memory card. Ex. 1004 at 9:45-59. For example, the CSD register stores card information such as data format, data transfer speed, etc. *Id.* 10:22-33; FIGs. 17A-B. The CSD also contains entries that influence the effect of commands executed by the memory card. For example, the WP_GRP_ENABLE bit of the CSD register controls whether groups of memory in the memory core are protected by execution of a SET_WRITE_PROT command. *Id.* 30:1-12; Fig. 17B. As another example, the WP_GRP_SIZE CSD register bit defines the size of the group to be protected by the SET_WRITE_PROT command. *Id.* CSD register entries may be R=readable, W=writable once, or E=erasable (multiple writable). *Id.* 10:29-31.

As of July 2004, the MMC standard allowed permanent (and temporary) write protection of an entire memory card by setting the PERM_WRITE_PROTECT (or TMP_WRITE_PROTECT) bit in the CSD register. *Id.* 12:56-67. The MMC standard also allowed write protecting memory groups of a memory card using the SET_WRITE_PROT command, which could be cleared by a CLR_WRITE_PROT command. *Id.* 30:9-12.

IV. OVERVIEW OF THE ’370 PATENT

A. Summary of the Claimed Subject Matter

The ’370 Patent relates to permanently write protecting a memory card. The ’370 Patent references the MMC specification in effect at the time. The ’370 Patent notes that it is desirable for some data to be protected from accidental or conscious deletion by a user while other data is alterable. Ex. 1001 at 1:31-49. According to the ’370 Patent, at the time of the invention, “[t]he MMC specification offers one solution to this kind of problem.” *Id.* 1:56-57. The MMC specification provides for write protecting a portion of a MMC using a specific command called SET_WRITE_PROT. *Id.* 1:60-62. However, the ’370 Patent asserts that command does not result in permanent write protection because the write protection can be cancelled using another command called (CLR_WRITE_PROT). *Id.* 1:63-66.

The ’370 Patent recognizes that at the time of the invention the MMC specification did provide permanent write protection (*i.e.*, write protection that was not changeable) by setting a permanent write protection bit called PERM_WRITE_PROTECT in the CSD (Card Specific Data) register of the memory card. *Id.* 1:66-2:2. However, the ’370 Patent asserts such permanent protection could only be applied to the entire card. The ’370 Patent discloses “a definition to the MMC standard for permanently write protecting a portion of a multimedia card.” *Id.* 2:7-8. Specifically, the ’370 Patent discloses “identifying a bit in a specific data register of the memory [and] setting said bit to have a certain predefined value that causes write protection command to mean permanent write protection of part of the memory.” *Id.* 2:12-18. The ’370 Patent discloses that after this bit is set, the command is executed to cause a part of the memory to be permanently write protected. *Id.* 2:19-21.

In one embodiment, the ’370 Patent discloses defining the PERM_WRITE_PROTECT bit of the CSD “in such way that setting of this bit does not as such protect the whole card” but rather “indicate[s] that all the write protect groups protected with SET_WRITE_PROT command…are permanently write protected and cannot be un-protected using CLR_WRITE_PROTECT command.” *Id.* 2:55-62. The segment size of the memory to be protected “is defined in the units of WP_GRP_SIZE groups as known to those skilled in the art.” *Id.* 2:62-64. Part of the CSD fields corresponding to this embodiment is shown in Table 1:

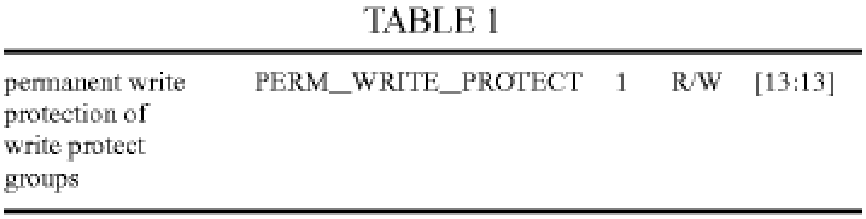


TABLE 1

permanent write protection of write protect groups	PERM_WRITE_PROTECT	1	R/W	[13:13]

Id. Table 1.

In another embodiment, one of the unused CSD bits (*e.g.*, named PARTIAL_PERM_WP) may be defined to "indicate that a portion of the multimedia card [] is permanently write protected." *Id.* 3:9-12. The PARTIAL_PERM_WP bit "should be re-programmable" and "could be cleared automatically when SET_WRITE_PROTECT command [] is received." Part of the CSD fields according to this embodiment is shown in Table 2:

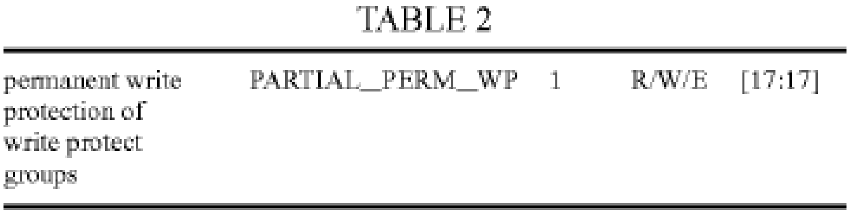


TABLE 2

permanent write protection of write protect groups	PARTIAL_PERM_WP	1	R/W/E	[17:17]

Id. Table 2.

Finally, while not the focus of the challenged claims, the '370 Patent also discloses defining a new command (rather than using a bit to redefine an existing command) to provide permanent write protection. *Id.* 3:58-65; 4:30-37.

B. The '370 Patent Prosecution History

On August 18, 2010, the Office mailed a Notice of Allowance stating that "[t]he primary reasons for allowance of [the] independent claims…is the inclusion in the claims of 'setting at least one bit in a data register configured to indicate that permanent write protection of the at least one part of the memory is allowed in order to redefine the command to allow permanent write protection that cannot be un-protected by a command, of the at least one part of the memory.'" Ex. 1002 at 24.

During prosecution, the applicant acknowledged that the Toombs Publication (U.S. Pub. 2001/0016887) cited by the Office "described that an entire card may be write protected by setting write protect bits in a CSD register" and disclosed that "addressed portions of memory can be write protected." *Id.* However, the applicant argued such write protection was not permanent "because Toombs describes removing/cancelling the write protection via a clear command." *Id.*

V. SUMMARY OF THE PRIOR ART

Chevallier describes temporarily or permanently protecting memory blocks against write and erase operations. Ex. 1003 at Abstract; ¶0008. Ex. 1006 ¶137. For example, Chevallier discloses a Flash memory device having a temporary lock function that temporarily locks memory blocks in response to a lock command. Ex. 1003 ¶¶0008, 0016. Ex. 1006 ¶137. The temporary lock function can be cleared, allowing memory blocks that were protected by the temporary function to be erased or reprogrammed. Ex. 1003 ¶0016. Ex. 1006 ¶137.

Chevallier also discloses a secure function that permanently locks memory blocks in response to a secure command. Ex. 1003 ¶¶0007, 0008, 0016, 0018, 0036. Ex. 1006 ¶138. The memory blocks protected by the secure function are permanently secured against write and erase operations. Ex. 1003 ¶¶0016, 0036. Ex. 1006 ¶138. Notably, Chevallier discloses that *the secure command and the lock command can be the same command*. Ex. 1003 ¶¶0008, 0020. Ex. 1006 ¶138. The secure function can be enabled and disabled by setting a secure function bit in a register. Ex. 1003 ¶0038. Ex. 1006 ¶138. The value of the bit controls whether a lock command will result in temporary write protection or in permanent write protection. Ex. 1003 CL. 18-19. Ex. 1006 ¶138.

Toombs describes write protecting at least portions of groups of a memory of a MultiMediaCard ("MMC"). Ex. 1004 at 29:40-42. Ex. 1006 ¶142. Toombs discloses that the MMC may include a number of data registers used to store information about the card, such as card/content specific information and configuration parameters. Ex. 1004 at 9:51-60. Ex. 1006 ¶142. For example, Toombs discloses a card specific data register (CSD) that contains information about the memory card's characteristics. Ex. 1004 at 10:21-26. Ex. 1006 ¶142.

Toombs discloses permanently write protecting the memory card by setting a non-erasable PERM_WRITE_PROTECT field in the CSD register. Ex. 1004 at

12:56-67; 30:1-3. Ex. 1006 ¶143. Toombs also discloses permanently write protecting the memory card by setting an erasable TMP_WRITE_PROTECT field in the CSD. Ex. 1004 at 12:56-67; 30:1-3. Ex. 1006 ¶143. Finally, Toombs discloses enabling write protection of memory groups (*i.e.*, less than the whole card) by setting a WP_GRP_ENABLE bit in the CSD. Ex. 1006 ¶143. The size of each memory group to be write protected is defined by the WP_GRP_SIZE field in the CSD. *Id*. The addressed group(s) are then write protected by executing a SET_WRITE_PROT command. Ex. 1004 at 30:3-10. Ex. 1006 ¶143.

Estakhri discloses a flash memory device having a microprocessor circuit and a volatile storage unit for executing operations on non-volatile memory. Ex. 1005 at Fig. 1. Ex. 1006 ¶146. The volatile storage unit stores firmware that is executed by the microprocessor. Ex. 1005 at 4:54-59. Ex. 1006 ¶146.

VI. CLAIM CONSTRUCTION

Pursuant to § 42.100(b), a claim in an unexpired patent shall be given its broadest reasonable interpretation ("BRI") in light of the specification in which it appears. Because the '370 Patent will not expire during the pendency of these proceedings, the Board should apply the BRI standard in its review. For terms not specifically listed and construed below, Petitioner interprets them for purposes of this review in accordance with their plain and ordinary meaning. Petitioner reserves the right to seek a different claim construction in litigation.

A. Level of Skill in the Art

At the time of the invention, a person of ordinary skill in the art ("POSITA") would be a person with a bachelor's degree in electrical engineering or a closely related field, and two to three years of experience in the field of memory device circuit design. Ex. 1006 ¶71. A person with less education but more relevant practical experience, or with less experience but more education, may also meet this standard. *Id*.

B. "a data register"

Challenged claims 1, 5, 12, 14, 16 17, and 25 all recite a "data register." A POSITA would have recognized that the broadest reasonable interpretation of this phrase is "a portion of memory containing information about a memory card." Ex. 1006 ¶¶116-18.

The specification refers primarily to a particular kind of data register, the "CSD (Card Specific Data) register." Ex. 1001 at 1:59-60. The CSD consists of fields which have bit strings of varying length. *Id.* 2:50-54. According to the specification, a "CSD provides information on how to access the card contents" (*Id.* 2:49), contains information about the size of memory groups to be write protected (*Id.* 1:58-60), contains a bit enabling permanent write protection on some or all of the memory card (*Id.* 2:65-3:7; 3:38-47), and contains information identifying the type of memory technology on a card and the address space of that memory (*Id.* 5:8-13). In sum, the '370 Patent discloses that a CSD contains information about the memory card. Ex. 1006 ¶117.

Based on the specification, a POSITA would have recognized that the broadest reasonable interpretation of "a data register" is "a portion of memory containing information about a memory card." Ex. 1006 ¶118. Given that the CSD is one type of data register, a POSITA would have recognized that under the broadest reasonable interpretation "data register" may, but is not necessarily required to, refer to some or all of the information in the CSD. *Id*.

C. "an additional data register"

Challenged claim 12 recites "a data register arranged to define at least one bit to indicate [] permanent write protection." Challenged claim 16 depends from claim 12 and recites that "an additional data register is arranged to control existence and characteristics of the at least one part of the memory." A POSITA would have recognized that the broadest reasonable interpretation of "an additional data register" as used in claim 16 is "a portion of memory, distinct from the memory portion containing the bit indicating permanent write protection, containing information about the memory card." Ex. 1006 ¶¶119-21.

The specification discloses that the CSD is made up of fields of bit strings having various lengths. Ex. 1001 at 2:50-54. Each bit string comprises a portion of memory corresponding to its bit string length. *Id.* 2:50-54; Fig. 3. The specification describes that a specific memory address (*e.g.*, bit 13 in the example of Table 1, or bit 17 in the example of Table 2) is dedicated to the bit that indicates permanent write protection. Ex. 1006 ¶120. In other words, a memory portion whose range of memory addresses contains bit 13 contains the bit indicating whether permanent write protection applies. *Id*. The specification discloses additional memory addresses holding bits that provide other information about the memory card. *Id*. For example, Figure 3 and 2:51-54 identify that bit addresses 127:126 may provide information about the structure of the CSD, and bit addresses 125:122 provide information about the specification version used by the card. *Id*. Particular other memory addresses hold the WP_GRP_SIZE bits that define the segments of the card that can be write protected. Ex. 1001 at 3:61-63. Ex. 1006 ¶119. A POSITA would understand that these memory addresses may or may not be contained in the same memory portion as the bit indicating whether permanent write protection applies. Ex. 1006 ¶120. Finally, the specification contemplates that a memory card may have two CSD registers, each describing information about a particular type of memory on the card. Ex. 1004 at FIG. 2 (elements 2 and 7); 4:42-51. Ex. 1006 ¶120. A POSITA would recognize that the two CSD registers may sit on two portions of the memory; one of the two portions that does not contain a bit indicating permanent write protection may be called "an additional data register." Ex. 1006 ¶120.

In sum, the specification describes a specific memory portion containing the bit indicating permanent write protection is allowed (*e.g.*, bit 13 of the CSD). Ex. 1006 ¶121. The specification describes additional portions of memory containing other information about the memory card. *Id.* ¶120. Thus, the broadest reasonable interpretation of "an additional register" as used in claim 16 is "a portion of memory, distinct from the memory portion containing the bit indicating permanent write protection, containing information about the memory card." *Id.* ¶121.

D. "redefine the command to allow permanent write protection"

Challenged claims 1, 12, and 25 each recite this element. A POSITA would have recognized this phrase to mean "cause a command that would not result in permanent write protection to result in permanent write protection." Ex. 1006 ¶¶122-23.

The '370 Patent explains that the MMC specification provides the command SET_WRITE_PROT, which write protects an addressed write-protect group. Ex. 1001 at 1:60-62. However, the specification states that the drawback of using the command is that "it does not allow the portion of the MMC to be permanently write protected" because "[t]he write protection can be cancelled using CLR_WRITE_PROTECT command to the addressed write-protect group." *Id.* 1:63-66. The specification proposes to define a bit that, if set, will "indicate that all the write-protect groups protected with SET_WRITE_PROT command 5 (CMD 28) are permanently write protected and cannot be un-protected using

CLR_WRITE_PROTECT command 5 (CMD29)." *Id.* 2:59-63; 3:27-31. In other words, the SET_WRITE_PROT command does not result in permanent write protection unless the defined bit is set. Ex. 1006 ¶123. Thus, a POSITA would understand from the specification that "redefine the command to allow permanent write protection" means "cause a command that would not result in permanent write protection to result in permanent write protection." *Id.*

E. "wherein said at least one bit has a certain predefined value"

Dependent claim 2 recites "wherein said at least one bit has a certain predefined value." A POSITA would have understood that the '370 Patent specification discloses associating the value of a bit with permanent write protection. Ex. 1006 ¶126. In order to allow permanent write protection, the bit is set to the value associated with permanent write protection. *Id*. Thus, a POSITA would have recognized that phrase to mean that the bit is set to a value associated with permanent write protection. *Id.* ¶¶124-26.

The specification discloses "setting said bit to have a certain predefined value that causes write protection command to mean permanent write protection." Thus, the specification makes clear that the "predefined value" refers to the value of the bit after it has been set. Ex. 1006 ¶125. In several places, the specification discloses embodiments where a bit is defined so that, when set, a command will result in permanent write protection of addressed memory segments. Ex. 1001 at

2:55-62 (defining the PERM_WRITE_PROTECT bit of the CSD such that setting the bit protects groups protected with the SET_WRITE_PROT command); 3:9-30 (defining an unused CSD bit so that, when the bit is set, specified memory groups become permanently write protected); 3:52-57 (defining a PARTIAL_PERM_WP bit such that setting the bit makes write protection permanent). Ex. 1006 ¶125. The specification contemplates defining the bit in the CSD register defined in the MultiMedia Card specification. Ex. 1001 at 2:65-3:8; 3:38-57. Ex. 1006 ¶125.

F. “wherein said at least one bit is reprogrammable”

Dependent Claims 3 recites “wherein said at least one bit is reprogrammable.” A POSITA would have understood that the specification contemplates that a bit set to allow permanent write protection when a command is executed may have its value changed so that permanent write protection does not always occur. Ex. 1006 ¶129. Thus, a POSITA would have recognized that phrase to mean that the value of the at least one bit is changeable. *Id.* ¶¶127-29.

The specification states that in one embodiment of the invention, an unused bit in a data register can be defined “to indicate that a portion of the multimedia card 1 is permanently write protected.” Ex. 1001 at 3:8-12. The specification refers to that bit as “PARTIAL_PERM_WP.” The specification describes how that bit can be set so that “groups protected with SET_WRITE_PROTECT command 5 (CMD28) become permanently write protected.” The specification then explains the value of that bit can be changed:

> In order to maintain backwards compatibility, the PARTIAL_PERM_WP bit should be re-programmable. Additionally, in order to prevent accidental permanent protection, the PARTIAL_TERM_WP [sic] bit could be cleared automatically when SET_WRITE_PROTECT 5 command is received.

Id. 3:31-36. Ex. 1006 ¶128. In addition, the specification in Table 2 identifies the PARTIAL_PERM_WP bit as "R/W/E," which a POSITA would understand refers to the bit as being readable, writable, and erasable (*i.e.*, capable of being written multiple times). Ex. 1001 at 3:43-48; Ex. 1004 at 10:21-33. Ex. 1006 ¶128.

G. "memory group"

Claims 5, 6, and 13-15 recite a "memory group." A POSITA would have recognized "memory group" to mean "a segment of memory." Ex. 1006 ¶¶130-32.

The specification consistently discusses performing operations on segments of memory. *Id.* ¶131. For example, the specification states that "the segment to be protected is defined in the units of WP_GRP_SIZE groups as specified in the CSD register. The write protection of the addressed write-protect group is then done using the SET_WRITE_PROT command." Ex. 1001 at 1:58-62; 2:60-67 (reiterating that the segment size to be protected by the SET_WRITE_PROT command is identified in the unit of WP_GRP_SIZE groups); 3:61-63 ("The segments of the card that can be write protected are defined using WP_GRP_SIZE bits of CSD 2 as usual.").

The prosecution history further confirms that the reasonable interpretation of "memory group" is "a segment of memory." Ex. 1006 ¶132. During prosecution, the Applicant contrasted the recited "memory group" with the entire memory card, and equated the recited "memory group" with Toombs Publication's disclosure of a write-protected portion of memory. Ex. 1002 at 39.

H. "control existence and characteristics of the at least one part of the memory"

Claim 16 recites "wherein an additional data register is arranged to control existence and characteristics of the at least one part of the memory." In light of the specification, a POSITA would have interpreted that phrase to mean that the additional data register "contains information about the memory." Ex. 1006 ¶¶133-34.

The specification describes a memory card that "contains an additional data register 7 (Extended CSD, EXT_CSD), in which the existence and characteristics" of a memory portion is described. Ex. 1001 at 4:46-49. The specification further states that "this EXT_CSD 7 is similar to the structure of the CSD 2." *Id.* 4:50-51. As explained in Part A of this Section, the CSD contains information about the memory card. The specification further confirms that the CSD and EXT_CSD contain similar information. *Id.* 5:8-10 (noting that that CSD and EXT_CSD both contain register data read by a host). Ex. 1006 ¶134.

VII. A REASONABLE LIKELIHOOD EXISTS THAT THE CHALLENGED CLAIMS ARE UNPATENTABLE.

All of the challenged claims are unpatentable as explained below.

A. Ground 1: Claims 1-3, 5-6, 12-17, and 25 are Anticipated Under 35 U.S.C. §§ 102(a) and (e) by Chevallier.

1. Independent Claim 1

(i) **A method comprising: write protecting at least one part of a memory by a command;**

Chevallier discloses write protecting memory blocks of a Flash memory "against unintended write operations." Ex. 1003 ¶0006. Chevallier discloses "memory blocks that are lockable in response to a lock command [corresponding to a lock function]." *Id.* ¶0008. In particular, Chevallier discloses that "some of the memory blocks have already been temporarily locked with a lock command written to a lock command register." *Id.* ¶0036. Chevallier's disclosure of temporarily locking memory blocks with a lock command discloses "write protecting at least one part of a memory by a command." Ex. 1006 ¶¶147-49.

(ii) setting at least one bit in a data register configured to indicate that permanent write protection of the at least one part of the memory is allowed in order to redefine the command to allow permanent write protection, that cannot be un-protected by a command, of the at least one part of the memory;

Chevallier discloses a secure function that permanently secures "memory blocks specified in [a] control data word" "against write and erase operations." Ex. 1003 ¶0036. Chevallier's secure function is permanent because while the "temporary lock function control can be cleared and the memory blocks erased or reprogrammed," "[t]he permanent secure function…cannot be cleared once it is set." Ex. 1003 ¶0016. Ex. 1006 ¶¶150-52.

Chevallier discloses that "the secure function is enabled by a nonvolatile secure function bit" Ex. 1003 at CL. 19 which is "part of a control register in the memory device." Ex. 1003 ¶0039. Ex. 1006 ¶152. Chevallier discloses "writing a data word that has a '1' in a secure function bit position to [a] control register [to] enable the [secure] function" (*i.e.*, permanent write protection function). Ex. 1003 ¶0039. Figure 5 of Chevallier illustrates a method for setting the secure function bit in order to enable the secure function and thus allow permanent write protection:

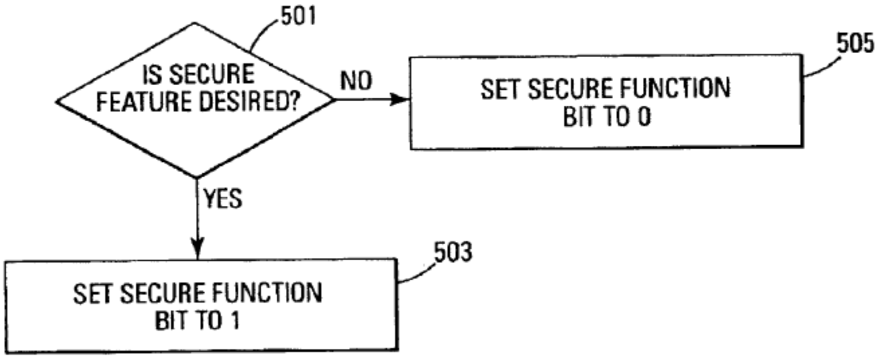

501
IS SECURE FEATURE DESIRED?
NO
SET SECURE FUNCTION BIT TO 0
505
YES
SET SECURE FUNCTION BIT TO 1
503

Fig. 5

Ex. 1003 at FIG. 5; ¶0038. Ex. 1006 ¶¶151-52

Chevallier also discloses "***to redefine the command to allow permanent write protection, that cannot be un-protected by a command***, of the at least one part of the memory." Ex. 1006 ¶¶153-57. As explained in Section VI.D, "redefine the command to allow permanent write protection" would be interpreted by a POSITA to mean "cause a command that would not result in permanent write protection to result in permanent write protection."

Chevallier differentiates between the "regular (temporary) lock function and [the] permanent secure function of the memory device." Ex. 1003 ¶0016. Ex. 1006 ¶156. The "temporary lock function control can be cleared and the memory blocks erased or reprogrammed" while "[t]he permanent secure function of the present invention cannot be cleared once it is set." Ex. 1003 ¶0016. "If the secure command is written to the [memory device] along with the control data word,…those memory blocks specified in the control data word are permanently secured against write and erase operations." *Id.* ¶0036; CL. 18. Chevallier describes that "[t]he permanent secure function of the present invention is an added level of security in addition to the temporary block locking function of the prior art. The secure function overrides the temporary locking function." *Id.* ¶0029.

Chevallier discloses that, in one embodiment, the "secure command … is the same as the lock command." *Id.* ¶¶0008, 0020. Chevallier describes that, when serving as a lock command, this "same" command locks (*i.e.*, temporarily write protects) a plurality of memory blocks of the memory device. Ex. 1003 ¶0008; CL. 18. Ex. 1006 ¶154. In this case, the "temporary lock function control can be cleared and the memory blocks erased or reprogrammed." Ex. 1003 ¶0016.

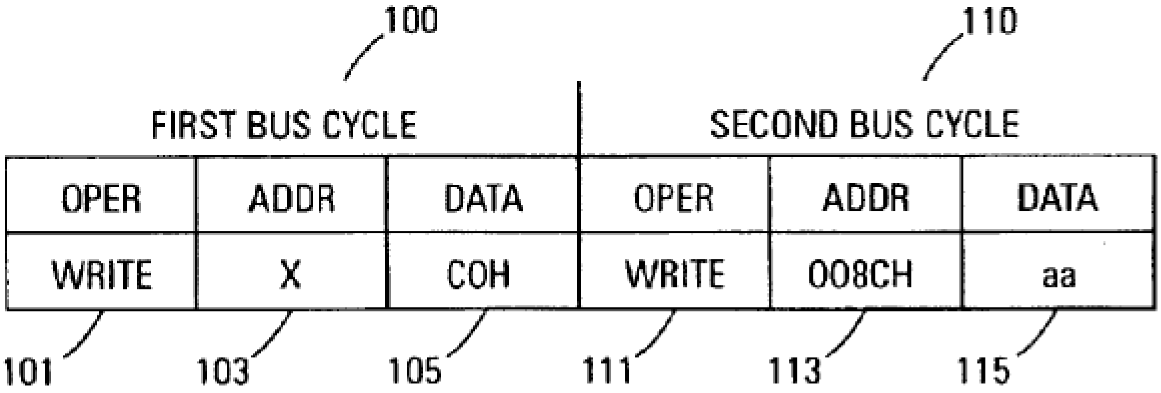
However, Chevallier discloses that when the secure function bit is set, the "same" command "initiates the secure function" which permanently secures "memory blocks specified in [a] control data word" "against write and erase operations." Ex. 1003 ¶¶0008, 0033, 0036. Ex. 1006 ¶154.

Claims 18 and 19 of Chevallier disclose the "lock" command functioning as a "secure" command after enabling the secure function by the secure function bit. Ex. 1003 at CLs. 18-19. Ex. 1006 ¶155. In particular, Claim 18 recites "a plurality of lockable memory blocks" that are "temporarily lockable in response to a lock command." Ex. 1003 at CL. 18. Claim 18 discloses the method involves "enabling a secure function," which Claim 19 discloses "is enabled by a non-volatile secure function bit." *Id.* CLs. 18-19. Claim 18 further discloses "submitting the lock command to the memory device to activate the secure function." *Id.* CL. 18.

In summary, Chevallier discloses that the lock command (temporary protection) and the secure command (permanent protection) may be the same command. Setting the secure function causes the command to operate as the secure command, which when executed results in permanent write protection that is "permanently secured against write and erase operations." Ex. 1003 ¶0036. Ex. 1006 ¶¶153-57. Without enabling the secure function bit, the command provides only temporary write protection, not permanent write protection. Ex. 1003 ¶0036. Ex. 1006 ¶¶153-56.

Chevallier also discloses "to allow permanent write protection, that cannot be un-protected by a command, ***of the at least one part of the memory***." Ex. 1006 ¶158-60. Specifically, Chevallier discloses that the secure function allows permanent write protection of memory blocks that were previously temporarily write protected by the lock command. *Id.* ¶158. For example, Chevallier discloses that "some of the memory blocks have already been temporarily locked with a lock command written to a lock command register." Ex. 1003 ¶0036. However, "if the secure command is written to the unused address along with the control data word, as described above, the lock function is overridden by the secure function and those memory blocks specified in the control data word are permanently secured against write and erase operations." *Id.*

Chevallier details permanently write protecting part of a memory by "set[ting] the command…for the secure function" in a first bus cycle and "set[ting] the particular memory blocks to be permanently disabled" in a second bus cycle. Ex. 1003 ¶0018. Ex. 1006 ¶159. This two-bus-cycle process is illustrated by Figure 1:



FIRST BUS CYCLE			SECOND BUS CYCLE		
OPER	ADDR	DATA	OPER	ADDR	DATA
WRITE	X	COH	WRITE	008CH	aa

Fig. 1

Ex. 1003 at FIG. 1.

Chevallier discloses that "[t]he second bus cycle (110) performs a write (111) operation of a control data word (115) that indicates the memory block or blocks that are to be secured." Ex. 1003 ¶0021. The memory block or blocks indicated by the control data word constitute "at least one part of the memory." Ex. 1006 ¶160.

(iii) executing the command in order to permanently write protect said at least one part of the memory.

As illustrated by Figure 1, Chevallier discloses permanently write protecting part of a memory by "set[ting] the command…for the secure function" in a first bus cycle and "set[ting] the particular memory blocks to be permanently disabled" in a second bus cycle. Ex. 1003 ¶0018. Ex. 1006 ¶¶161-63. Chevallier further teaches that "[i]f the secure command is written to the [memory device] along with the control data word,…those memory blocks specified in the control data word are permanently secured against write and erase operations." Ex. 1003 ¶0036; CL. 18.

2. Dependent Claim 2

(i) A method according to the claim 1, wherein said at least one bit has a certain predefined value.

As explained in Section VI.E, a POSITA would interpret the phrase "wherein said at least one bit has a certain predefined value" to mean that the bit is set to a value associated with permanent write protection.

Chevallier describes "writing a data word that has a '1' in a secure function bit position to [a] control register [to] enable the [secure] function" (*i.e.*, to enable permanent write protection). Ex. 1003 ¶¶0038-0039; FIG. 5. In other words, when the secure function bit is set to "1," the secure function is enabled and execution of the secure command permanently write protects part of a memory. The value "1" of the secure function bit is therefore associated with permanent write protection. Ex. 1003 ¶¶0036, 0038-0039; FIG. 5. Ex. 1006 ¶166.

3. Dependent Claim 3

(i) A method according to the claim 1, wherein said at least one bit is reprogrammable.

As explained in Section VI.F, a POSITA would have understood this element to mean that the value of the at least one bit is changeable.

Chevallier describes that the value of the secure function bit can be changed based on whether the secure function is desired. *See, e.g.,* Ex. 1003 ¶0037 ("If a customer desires to use the [secure] function for a particular implementation, the customer or the manufacturer can enable it. If the secure function is not required, the feature does not need to be enabled."); *see also* FIG. 5 (disclosing setting the secure bit to "0" if the secure function is not desired and to "1" if the secure function is desired.) Ex. 1003 ¶0039. Ex. 1006 ¶169. A POSITA would recognize that because Chevallier discloses a method for enabling and disabling the secure function bit by changing the bit's value and further discloses that the secure function can be enabled when customer desires and disabled when the customer does not desire the function, then Chevallier discloses that the secure function bit is changeable (*i.e.*, so that the secure function can be enabled or disabled, depending on the situation, to meet the customer's needs). Ex. 1006 ¶169. Thus, the secure function bit is reprogrammable. *Id*.

Moreover, claim 19 of Chevallier teaches that "the secure function is enabled by a nonvolatile secure function bit." Ex. 1003 at CL. 19. By specifying "nonvolatile," claim 19 suggests that its antecedent independent claim 18 covers other possible implementations of the secure function. Ex. 1006 ¶170. As a volatile secure function bit is the only possible alternative to a non-volatile secure function bit, a POSITA would have understood claim 18 to encompass and inherently disclose a volatile implementation of the secure function bit. *Id*. A volatile bit only maintains its data when its device is powered; thus, a volatile secure function bit is necessarily changeable. *Id*.

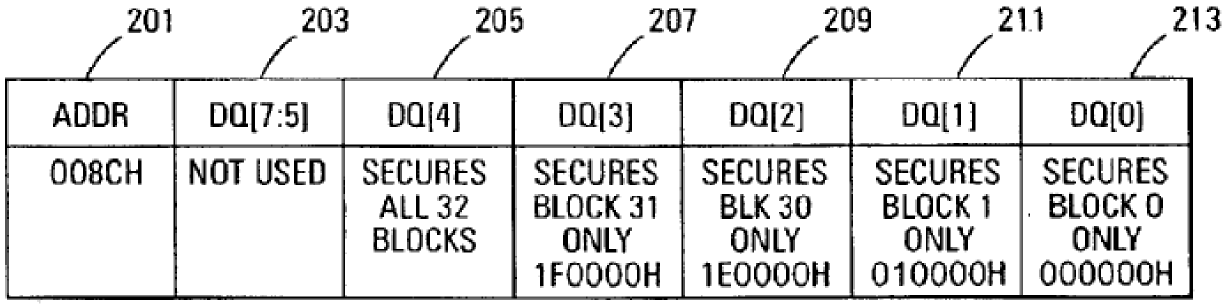
4. Dependent Claim 5

(i) **A method according to claim 1, wherein said at least one part of the memory comprises at least one memory group having a certain memory size defined in the data register.**

As explained in Section VI.G, a POSITA would have recognized "memory group" to mean "a segment of memory." Chevallier discloses all the limitations of claim 5.

As illustrated by Figure 1, Chevallier discloses permanently write protecting part of a memory by "set[ting] the command…for the secure function" in a first bus cycle and "set[ting] the particular memory blocks to be permanently disabled" in a second bus cycle. Ex. 1003 ¶0018. Ex. 1006 ¶173. Chevallier further teaches that "[i]f the secure command is written to the [memory device] along with the control data word,…those memory blocks specified in the control data word are permanently secured against write and erase operations." Ex. 1003 ¶0036. Chevallier's memory blocks are segments of memory. Ex. 1006 ¶173.

Figure 2 illustrates how the control data word is used to "indicate which block or blocks of memory to permanently secure." Ex. 1003 ¶0022.


201
203
205
207
209
211
213

ADDR	DQ[7:5]	DQ[4]	DQ[3]	DQ[2]	DQ[1]	DQ[0]
008CH	NOT USED	SECURES ALL 32 BLOCKS	SECURES BLOCK 31 ONLY 1F0000H	SECURES BLK 30 ONLY 1E0000H	SECURES BLOCK 1 ONLY 010000H	SECURES BLOCK 0 ONLY 000000H

Fig. 2

Specifically, Chevallier discloses one embodiment in which "in order to secure the particular memory block represented by each control bit, a logic 0 is used in that particular control bit location." Ex. 1003 ¶0023. For example, "control bit DQ0 (213) secures memory block 0" while "memory block 1…is represented by bit DQ1 (211)." *Id.* ¶¶0024-25. Memory blocks 30 and 31 may be represented by bit DQ2 (209) and DQ3 (207), respectively. *Id.* Chevallier contemplates that "[a]lternate [sic] embodiments represent other memory blocks by the bits of the control data word." *Id.* ¶0027. For example, Chevallier discloses using the unused control bits DQ5-7 (203) "to represent other memory blocks to secure." *Id.* ¶0026.

Chevallier contemplates protecting any number of memory blocks by specifying in the control data word the corresponding combination of control bits. Ex. 1003 ¶0015 ("provid[ing] a permanent disablement of a write or erase operation to one or more memory blocks of a Flash memory device."); ¶0027 ("different combinations of bits in the control data word indicate different memory blocks."). Ex. 1006 ¶177. Thus, by specifying the number of memory blocks to be secured, Chevallier's control data word defines the size of the memory group to be secured. *Id.*

Chevallier teaches writing the control data word to the data register that has the secure function bit in particular embodiments. Ex. 1006 ¶178. In particular, it teaches that "[t]he [secure] function bit may be part of a control register in the memory device." Ex. 1003 ¶0039. Chevallier also discloses that "[a]n array of control registers (680) store the secure command and the control data word of the present invention." *Id.* ¶0045. Chevallier's control register(s) can be interpreted to be a "data register" because they are a portion of memory containing information (*e.g.*, the secure function bit and the memory blocks (size of memory) to secure) about Chevallier's memory card. Sec.VI.B. Ex. 1006 ¶178.

Therefore, Chevallier discloses that the size of the permanently write protected part of the memory is defined by the memory blocks specified by the control data word written to the control register. Ex. 1006 ¶¶172-179.

5. Dependent Claim 6

(i) A method according to claim 5, wherein redefining the command allows permanent write protection of each memory group individually.

As explained in Section 4(i), Chevallier discloses permanently write protecting part of a memory by "set[ting] the command…for the secure function" in a first bus cycle and setting a control data word specifying "the particular memory blocks to be permanently disabled" in a second bus cycle. Ex. 1003

¶0018; ¶0036. Ex. 1006 ¶182. Chevallier further discloses an embodiment that assigns each memory block a control bit that designates whether that memory block will be secured. Ex. 1006 ¶¶182-183. "In order to secure the particular memory block represented by each control bit, a logic 0 is used in that particular control bit location." Ex. 1003 ¶0023; Fig. 2. Fig. 2 further illustrates that each control bit secures *only* the individual memory block to which it is assigned. Ex. 1006 ¶¶182-183. Chevallier therefore discloses that each of memory blocks can be individually write protected by specifying the appropriate corresponding control bit in the control data word. *Id.* ¶¶181-84.

6. Independent Claim 12

(i) An apparatus comprising: an interface controller arranged to write protect at least one part of a memory of said apparatus by a command;

Chevallier discloses incorporating a Flash memory device in portable computers, personal digital assistants (PDAs), digital cameras, and cellular telephones. Ex. 1003 ¶0004. Figure 6 of Chevallier illustrates a memory device 600 interfacing with a host processors 610. A POSITA would have considered memory device 600, either with or without host processor 610, to be an apparatus. Ex. 1006 ¶186.

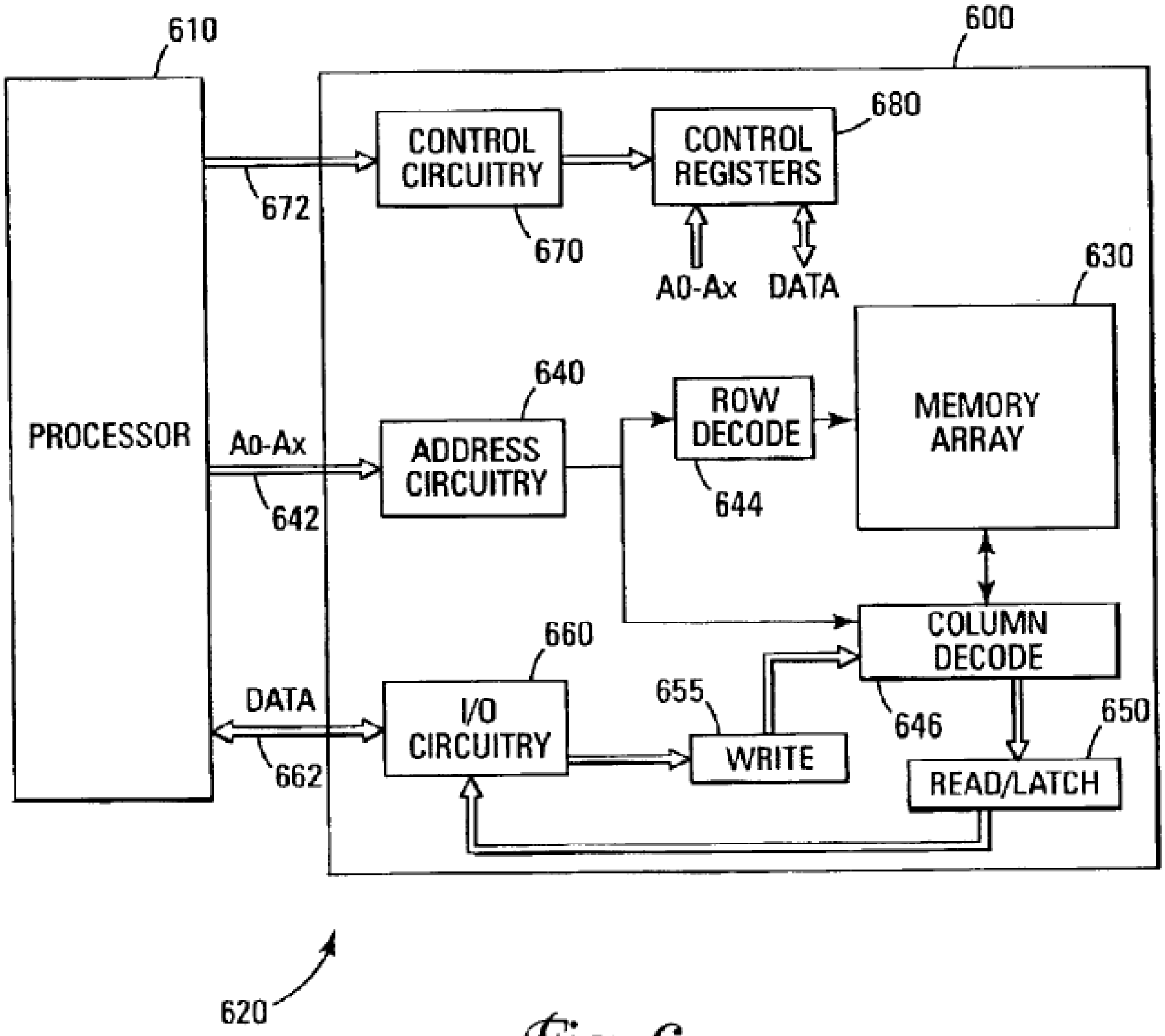


Fig. 6

As illustrated by Figure 6, Chevallier discloses the memory device (600) “includ[ing] an array of memory cells (630).” Ex. 1003 ¶0041. Chevallier also discloses that the memory device (600) includes a “[c]ommand control circuit (670) [that] decodes signals provided on control connections (672) from the processor (610)” and that the “signals are used to control the operations on the memory array (630), including data read, data write, and erase operations.” *Id.*

¶0044. Fig. 6 of Chevallier illustrates control circuitry 600 interfacing with an array of control registers 680. According to Chevallier, an array of control registers 680 "store the secure command and the control data word" and can be "programmed with the appropriate secure command and control data word." *Id.* ¶0045. According to Chevallier, the secure command can be written to the control circuitry of the memory device (*Id.* ¶¶19, 33), and the control data word can be written to an unused memory address in the control register (*Id.* ¶¶21, 34, CL. 12). The secure command and control data word can be written to the same unused address. *Id.* ¶21. The lock command is also written to a control register. *Id.* CL. 4, CL. 9 Finally, the secure function bit can be stored in the control register and can be changed by writing the appropriate data word to the secure function bit position in the register. *Id.* ¶39. Ex. 1006 ¶188

The '370 Patent states that its "interface controller" "handles the accesses according to the address sent be [sic] the host 6." Ex. 1001 at 5:13-15. A POSITA would recognize that the broadest reasonable interpretation of "interface controller" would include Chevallier's command control circuitry; control circuitry, address circuitry, and I/O circuitry interfacing with the host; the processor; or any combination thereof. Ex. 1006 ¶189.

A POSITA would understand that Chevallier discloses: (1) control circuitry that writes a lock command and/or secure command to a memory address in a control register; and (2) a processor that writes via control connections a lock command and/or secure command to the control circuitry. Ex. 1006 ¶191. As explained in Section A.1.(i) and (ii), the lock command and secure command provide temporary and permanent write protection, respectively. Thus, Chevallier discloses this limitation. *Id.* ¶¶186-91.

(ii) a data register arranged to define at least one bit to indicate that permanent write protection of the at least one part of the memory is allowed;

As explained in Section A.1.(ii), Chevallier's control register is a data register that contains the secure function bit, which enables the secure function and thus indicates that permanent write protection of at least one part of Chevallier's memory is allowed. Ex. 1006 ¶¶151-52,192-95.

(iii) a controller arranged to set the at least one bit in order to redefine the command to allow permanent write protection that cannot be un-protected by a command, of the at least one part of the memory of said apparatus;

As shown in Section A.6.(i), Chevallier discloses a control circuitry that writes to a control register, and further that the control register stores a secure function bit set by writing the appropriate data word to that bit. Ex. 1006 ¶¶188-89, 191, 197-98. As shown in Section A.1.(ii), Chevallier discloses the recited bit. *Id.* ¶¶150-60, 199-205.

(iv) the controller arranged to execute the command in order to permanently write protect said at least one part of the memory.

As shown in Section A.6.(i), Chevallier discloses: (1) control circuitry that writes a secure command to a memory address in a control register; and (2) a processor that writes via control connections a secure command to the control circuitry. Ex. 1006 ¶¶188-89, 191, 207. As shown in Section A.1.(iii), Chevallier discloses that writing the secure command permanently write protects the memory identified by the control data word. *Id.* ¶¶161-63, 208-09.

7. Dependent Claim 13

(i) An apparatus according to claim 12, wherein the memory is arranged to comprise at least one memory group.

As shown in Section A.4.(i), Chevallier's memory is arranged to comprise at least one memory group. *Id.* ¶172-79, 211-16.

8. Dependent Claim 14

(i) An apparatus according to claim 13, wherein the data register is arranged to define a memory size of the at least one memory group.

As shown in Section A.4.(i), Chevallier's data register defines the size of Chevallier's memory groups. *Id.* ¶172-79, 218-21.

9. Dependent Claim 15

(i) An apparatus according to claim 14, wherein the controller is arranged to define the command to allow permanent write protection of the at least one memory group individually.

As shown in Section A.6.(i), Chevallier discloses a control circuitry that writes a control data word to a control register. Ex. 1006 ¶188-89, 191, 224. As explained in Section A.4.(i), the control data word specifies permanent write protection of memory groups individually or in combination. *Id.* ¶172-79, 224-29.

10. Dependent Claim 16

(i) An apparatus according to claim 12, wherein an additional data register is arranged to control existence and characteristics of the at least one part of the memory.

As explained in Section VI.C, a POSITA would have recognized that the broadest reasonable interpretation of "an additional data register" as used in Claim 16 is "a portion of memory, distinct from the memory portion containing the bit indicating permanent write protection, containing information about the memory card." As explained in Section VI.H, a POSITA would have interpreted "control existence and characteristics" to mean that the additional data register "contains information about the memory." Chevallier discloses all the elements of this claim.

Chevallier discloses that the secure function bit "may be part of a control register in the memory device." Ex. 1003 ¶0039. Chevallier also discloses an array of registers—some of which are used for typical control functions and others are reserved for expansion and/or future use. *Id.* ¶0045. Chevallier discloses that the control registers are memory cells (*Id.*); thus, they can be considered to be "data registers" as interpreted in Section VI.B. Ex. 1006 ¶232. Moreover, the "typical control functions" disclosed by Chevallier are distinct from the secure bit, and thus is placed in a memory portion distinct from the memory portion in the control register that the secure bit is part of. *Id*. The portion(s) of the register(s) containing those control functions can therefore be considered to be "a portion of memory distinct from the memory portion containing the bit indicating permanent write protection." *Id*. Finally, even if "additional data register" were interpreted to require a separate register, Chevallier discloses an *array* of control registers. Thus, even under this interpretation, all registers other than the one containing the secure bit would correspond to the "additional data register." *Id*.

Chevallier discloses that some of its control registers contain typical control functions for the memory device. Those control registers therefore contain information about the memory device (*e.g.*, the control functions that can be performed on it). Ex. 1006 ¶233. In addition, Chevallier discloses that writing the secure (or lock) command and control data word to the control register specifies the memory blocks that are write protected. Ex. 1003 at FIG. 2; ¶¶0021-0022. Ex. 1006 ¶233. The memory addresses at which the secure (or lock) command and data word were written can be read with a read query to indicate whether the corresponding memory blocks are not locked, are locked, or are secured. Ex. 1003 at FIG. 3; ¶¶0028-0031. Ex. 1006 ¶233. Thus, at least one control register contains information about the write protection status (read only, or writable and/or erasable) of the memory. Ex. 1006 ¶233. Chevallier therefore discloses "an additional data register [] arranged to control existence and characteristics of the at least one part of the memory." Ex. 1006 ¶¶231-33.

11. Dependent Claim 17

(i) An apparatus according to claim 16, wherein the additional data register is arranged to define access to the at least one part of the memory.

Chevallier discloses that its "array of control registers (680) store the secure command and the control data word." Ex. 1003 ¶0045. Chevallier further discloses that writing the secure (or lock) command and control data word to the control register specifies the memory blocks that are write protected. Ex. 1003 at FIG. 2; ¶¶0021-0022. Ex. 1006 ¶236. The memory addresses at which the secure (or lock) command and data word were written can be read with a read query to indicate whether the corresponding memory blocks are not locked, are locked, or are secured. Ex. 1003 at FIG. 3; ¶¶0028-0031. Ex. 1006 ¶236. Thus, the control register defines access (read only, or writable and/or erasable) to those memory blocks. Ex. 1006 ¶¶235-36.

12. Independent Claim 25

(i) A memory device having stored thereon instructions that, when executed, perform:

Chevallier discloses a memory device, which is a Flash memory device in at least one embodiment (Ex. 1003 ¶0002; FIG. 6.), and also discloses commercial device (portable computers, personal digital assistants (PDAs), digital cameras, and cellular telephones) incorporating memory that could each be considered a memory device." Ex. 1003 ¶0004. Ex. 1006 ¶239. Chevallier teaches that "program code, system data such as a basic input/output system (BIOS), and other firmware can typically be stored in Flash memory." Ex. 1003 ¶0004. A POSITA would have understood that Chevallier's disclosure of a Flash memory device storing program code and firmware teaches "[a] memory device having stored thereon instructions." Ex. 1006 ¶239. Furthermore, a POSITA would have understood that at least part of the program code or firmware stored on the Flash memory device would have stored the instructions necessary to perform the functions Chevallier describes corresponding to elements (ii)-(iv) below. *Id.* In addition, a POSITA would have understood that Chevallier's memory device would necessarily have had to execute instructions to perform the functionality Chevallier discloses, and that an operational memory device would necessarily have stored such instructions, whether a program code, in firmware, or in a state machine implemented as an application-specific integrated circuit (ASIC). *Id.*

(ii) write protecting at least one part of a memory by a command;

As explained above in Section A.1.(i), Chevallier discloses this limitation. Ex. 1006 ¶¶149-56, 158-60, 241.

(iii) setting at least one bit in a data register configured to indicate that permanent write protection of the at least one part of the memory is allowed in order to redefine the command to allow permanent write protection, that cannot be unprotected by a

command, of the at least one part of the memory; and

As explained above in Section A.1.(ii), Chevallier discloses this limitation. Ex. 1006 ¶¶149-56, 158-60, 242-52.

(iv) executing the command in order to permanently write protect said at least one part of the memory.

As explained above in Section A.1.(iii), Chevallier discloses this limitation. Ex. 1006 ¶161-63, 253-55.

B. Ground 2: Claims 1-3, 5-6, 12-17, and 25 are obvious under 35 U.S.C. § 103 over Chevallier in view of the knowledge of a POSITA

1. Independent claim 1, 12, and 25

To the extent Chevallier does not explicitly recite that the "secure function bit" is set in order to "redefine" the lock command to become the (permanent) secure command, it would be obvious to use the bit in that manner, for example to eliminate any ambiguity as to whether the command meant "lock" or "secure." Ex. 1006 ¶¶259, 270, 276.

As shown above in Section A.1.(i)-(ii), Chevallier describes that a lock command used for temporary write protection and a secure command used for permanent write protection may be the *same* command. *Id*. ¶¶153-57, 259, 270, 276. A POSITA would have understood that when the lock function and the secure function result from the same command, it would be beneficial to unambiguously specify which function results from execution of the command. *Id*. ¶259, 270, 276.

A POSITA would understand that Chevallier's secure function bit, which enables and disables the secure function, could provide the beneficial specificity—if the secure function is not enabled, then the command results in the lock function; if the bit is enabled then the command results in the secure function. *Id.*

2. Dependent claim 3

(i) Wherein said at least one bit is reprogrammable

To the extent Chevallier does not disclose that the secure function bit was reprogrammable, it would have been an obvious matter of design choice to implement the register address for the secure function bit to be reprogrammable. Ex. 1006 ¶262.

Moreover, it would have been obvious to try implementing a reprogrammable secure function bit given the small number of options (*i.e.*, reprogrammable or not reprogrammable). *Id.* ¶262.

Additionally, a POSITA would have been motivated to implement the secure function bit to be reprogrammable rather than non-reprogrammable to ensure flexibility of the device. *Id.* ¶263. As shown above in Sections A.1.(ii) and B.1, Chevallier discloses that the secure function bit determines whether temporary or permanent write protection occurs. *Id.* ¶263. A POSITA would have understood that it is beneficial for a memory device to sometimes apply temporary write protection and other times apply permanent write protection. *Id.* ¶263. Notably, Chevallier contemplates providing such flexibility to a user. *Id.* ¶264. *See* Ex. 1003

¶0037 ("If a customer desires to use the function for a particular implementation, the customer or the manufacturer can enable it. If the secure function is not required, the feature does not need to be enabled."). A POSITA would have recognized that by making the secure function bit changeable, a user could enable the secure function when it is desired and disable it when it is not. Ex. 1006 ¶264. In fact, Chevallier provides a method for doing just that. Ex. 1003 at FIG. 5; ¶0038. Ex. 1006 ¶264. A POSITA would have found it obvious to implement the memory device of Chevallier in a way such that the customer can use the method illustrated by Figure 5 as desired to switch between permanent and temporary write protection. Ex. 1006 ¶264.

In addition, if the secure function bit was made non-reprogrammable, setting the secure function bit would render inaccessible Chevallier's temporary lock function when the lock and secure commands are the same. *Id.* ¶263.

3. Dependent claims 5 and 14

(i) "at least one memory group having a certain memory size defined in the data register"/ "data register…define[s] a memory size…."

As shown in Section A.4.(i), Chevallier discloses a memory based on a block architecture and protecting blocks of the memory specified by a control data word, where the memory blocks are located at particular memory addresses. Ex. 1006 ¶¶172-79, 267, 273. A POSITA would have understood that securing specific memory block as Chevallier discloses requires information about the structure of the memory, including the size of each memory block. *Id*. ¶¶267, 273.

In addition, it was known that the size of each memory block may be directly stored in the control register or indirectly indicated by addresses of the memory blocks stored in the control register, and storing memory size information in one memory location versus another memory location was simply a design choice. *Id*. It would therefore be obvious to define the size of the memory group to be secured in a register. *Id.*

4. Dependent claims 2, 6, 13, and 15-17

Each of these claims depends from at least one claim that this section shows is obvious over Chevallier in view of the knowledge of a POSITA. Section A shows that Chevallier discloses the limitations introduced by these claims. Thus, Chevallier renders these claims obvious.

C. Ground 3: Claims 1-7, 12-19, and 25 are obvious under 35 U.S.C. § 103 over Chevallier in view of Toombs

Chevallier discloses the elements of claims 1-3, 5, 6, 12-17 and 25. *See* Ground 1. Additionally, those claims as well as claims 4, 7, 18 and 19 are obvious over Chevallier and Toombs, as explained below with reference to particular claim elements.

1. Independent claim 1

(i) Setting at least one bit in order to redefine the command to allow permanent write protection that cannot be unprotected by a command

Toombs discloses register fields that provide supplemental information controlling the meaning of particular commands (*i.e.*, redefining those commands). Ex. 1006 ¶280. For example, Toombs discloses a WP_GRP_ENABLE bit in the CSD register that "is used to indicate whether the write protection group is enabled." Ex. 1004 at 12:25-28. A SET_WRITE_PROT command can be used to "set[] the write protection of [] addressed write-protect group" only if the WP_GRP_ENABLE bit is set to 1 in the CSD register. *Id.* at 30:3-12. As another example, a READ_BL_LEN field in the CSD register controls the maximum size of a block of data that can be read using a command READ_SINGLE_BLOCK command. *Id.* at 20:5-17.

Toombs also discloses a PERM_WRITE_PROTECT bit that, when set, permanently protects the memory card such that "all write and erase commands for this card are permanently disabled." Ex. 1004 at 12:56-61. Toombs' permanent write protection thus cannot be cleared by a command, such as Toombs' CLR_WRITE_PROT command that clears temporary write protection. Ex. 1004 at 30:8-12. Ex. 1006 ¶282.

It would have been obvious to a POSITA to apply Toombs' technique of using register bits to control a command's functionality to Chevallier's system wherein a single lock/secure command can have two different functions, in order to control the meaning/functionality of the lock/secure command. Ex. 1006 ¶283. *See*

Section A.1.(ii). A POSITA would have understood that it is necessary to unambiguously specify, whenever the command is executed, which of the two functions is to be initiated. *Id*. This would motivate a POSITA to combine the teachings of Chevallier and Toombs to solve this problem and use the existing secure function bit in Chevallier to control the meaning of the lock/secure command. *Id*.

It would also have been obvious to a POSITA to implement Chevallier's secure function bit and secure/lock command in the memory card of Toombs. Ex. 1006 ¶285. For example, a POSITA would have been motivated to introduce the lock/secure function of Chevallier to Toombs in order to provide the memory card of Toombs with the additional functionality of flexibly invoking permanent write protection of Chevallier, and a POSITA would have further been motivated to store Chevallier's secure function bit for controlling that command in Toombs' data structure storing information command functionality, *e.g.* the CSD register, to keep such information organized in one data structure. *Id*.

This combination would also yield a predictable result—that the command initiates the secure function when the binary secure function bit is set to enable the secure function, and the command initiates the lock function when the secure function bit is set to disable the secure function. *Id*.

(ii) A data register

Toombs discloses that "each of the cards of the MultiMediaCard system comprises a group of registers for storing a variety of status and internal information." Ex. 1004 at 9:46-48. Toombs discloses an embodiment, in which the information is stored in five registers including OCR, CID, CSD, RCA, and DSR. *Id.* 9:51-53. In particular, Toombs teaches that "[t]he CSD register is responsible for providing information to the MultiMediaCard host on how to access the card content" and that "the CSD register stores values defining the data format … data transfer speed… etc." *Id.* 10:24-29. Toombs discloses an embodiment where the CSD register contains information about the card's write protection, including the WP_GRP_ENABLE bit that enables the SET_WRITE_PROT function, and also including the PERM_WRITE_PROTECT and TMP_WRITE_PROTECT bits. *Id.* FIG. 17B; 12:56-67; 30:1-12.

It would also have been obvious to a POSITA to implement Chevallier's secure function bit using the CSD of Toombs, as explained above. *See* Section C.1.(i).; Ex. 1006 ¶285.

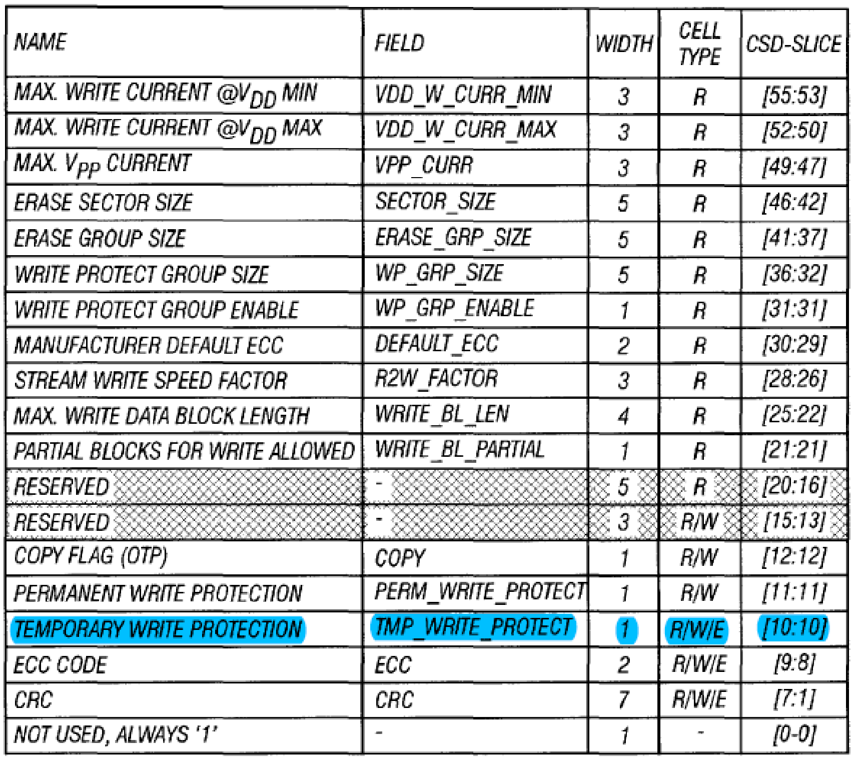
2. Dependent claim 2

Chevallier discloses all the limitations of claim 2. *See* Section A.2.(i); Ex. 1006 ¶287.

3. Dependent claim 3

(i) Wherein said at least one bit is reprogrammable

Toombs discloses a TMP_WRITE_PROTECT field in the CSD register that "temporarily protects the whole card content against overwriting or erasing (all write and erase commands for this card are permanently disabled)" and that "can be set and reset." Ex. 1004 at 12:62-66. Here, setting the TMP_WRITE_PROTECT bit applies temporary write protection because the TMP_WRITE_PROTECT field can be reset by a user. Ex. 1006 ¶289. This register field is illustrated below:



NAME	FIELD	WIDTH	CELL TYPE	CSD-SLICE
MAX. WRITE CURRENT @V_{DD} MIN	VDD_W_CURR_MIN	3	R	[55:53]
MAX. WRITE CURRENT @V_{DD} MAX	VDD_W_CURR_MAX	3	R	[52:50]
MAX. V_{PP} CURRENT	VPP_CURR	3	R	[49:47]
ERASE SECTOR SIZE	SECTOR_SIZE	5	R	[46:42]
ERASE GROUP SIZE	ERASE_GRP_SIZE	5	R	[41:37]
WRITE PROTECT GROUP SIZE	WP_GRP_SIZE	5	R	[36:32]
WRITE PROTECT GROUP ENABLE	WP_GRP_ENABLE	1	R	[31:31]
MANUFACTURER DEFAULT ECC	DEFAULT_ECC	2	R	[30:29]
STREAM WRITE SPEED FACTOR	R2W_FACTOR	3	R	[28:26]
MAX. WRITE DATA BLOCK LENGTH	WRITE_BL_LEN	4	R	[25:22]
PARTIAL BLOCKS FOR WRITE ALLOWED	WRITE_BL_PARTIAL	1	R	[21:21]
RESERVED	-	5	R	[20:16]
RESERVED	-	3	R/W	[15:13]
COPY FLAG (OTP)	COPY	1	R/W	[12:12]
PERMANENT WRITE PROTECTION	PERM_WRITE_PROTECT	1	R/W	[11:11]
TEMPORARY WRITE PROTECTION	TMP_WRITE_PROTECT	1	R/W/E	[10:10]
ECC CODE	ECC	2	R/W/E	[9:8]
CRC	CRC	7	R/W/E	[7:1]
NOT USED, ALWAYS '1'	-	1	-	[0-0]

FIG. 17B

Ex. 1004 at FIG. 17B (emphasis supplied).

The TMP_WRITE_PROTECT field is marked "R/W/E," which indicates that the field is readable, writable, and erasable (multiple writable). Ex. 1004 at 10:29-32. Toombs thereby teaches using a reprogrammable register field to enable or disable write protection for an entire memory.

It would have been obvious to a POSITA to combine the teachings of Toombs and Chevallier to make the secure function bit disclosed by Chevallier erasable. Ex. 1006 ¶291. Such a combination would have been a simple design choice to allow a user to flexibly choose whether to temporarily or permanently write protect part of the memory by controlling the value of the secure function bit. *Id*.. Notably, Chevallier contemplates providing such flexibility to a user. *Id*. Ex. 1003 ¶0037. In addition, a POSITA would have been motivated to introduce the R/W/E feature to Chevallier as one way to implement Chevallier's method of Figure 5, *i.e.*, to enable and disable the secure function as desired. Ex. 1006 ¶291. Finally, introducing the known R/W/E feature disclosed in Toombs to the secure function bit of Chevallier would have the predictable result of making Chevallier's secure function bit R/W/E. *Id*.

Alternatively, a POSITA would have been motivated to improve the functionality of permanently write protecting an entire memory using the reprogrammable TMP_WRITE_PROTECT bit, as disclosed in Toombs, by using this bit in conjunction with the control data word, as disclosed in Chevallier. Ex. 1006 ¶292. This control data word may be written in one or more unused addresses in the CSD register or another suitable register. *Id*. Chevallier suggests making this improvement by stating that the block architecture of "[n]ewer memory devices" "allows the file system to erase blocks of Flash memory instead of the entire device" and that "critical system code can be stored in a lockable block of memory while other blocks are allocated to other portions of code or data." Ex. 1003 ¶0005. Ex. 1006 ¶292. This combination would allow a user to specify a particular portion of the memory (*e.g.*, critical system code, data necessary for the operation of an application, or important records) to permanently write protect while keeping other portions (*e.g.*, system data that can be updated or user files that are overwritten, such as drafts of documents) available for writing. Ex. 1006 ¶292.

4. Dependent claim 4

(i) Wherein executing the command clears automatically said at least one bit

Toombs discloses erasing data stored in memory groups by first tagging each group to be erased and then using one command to erase all tagged groups. Ex. 1004 at 28:10-25. Ex. 1006 ¶296. According to Toombs "[a]ll tag bits are cleared by each command except a tag or untag command." Ex. 1004 at 28:37-39. Toombs therefore discloses setting one or more bits (*e.g.*, group tags) to specify a functionality of a command (*e.g.*, particular memory groups to be erased by the erase command) and executing the command, which automatically clears the one or more bits. Ex. 1006 ¶296. Toombs further discloses "issuing [a] status command [] to read [] bits" in a status register and that some of the bits are cleared after "reception of a valid command." (Ex. 1004 at 24:55-25:6.)

It would have been obvious to a POSITA to combine the feature where a command automatically clears related register bits, as disclosed in Toombs, with the use of the secure function bit to specify the meaning of the lock/secure command, as disclosed in Chevallier, such that executing the secure command automatically clears the secure function bit. Ex. 1006 ¶297. A POSITA would have been motivated to do so to avoid inadvertently and permanently securing part of the memory following a permanent secure command (e.g., where the next secure command was intended to only be temporary). *Id.* In fact, Chevallier recognizes that inadvertent securing of blocks is a problem to be avoided. Ex. 1003 ¶0035 ("It is desirable to use a voltage so that the memory blocks cannot be inadvertently secured.") Ex. 1006 ¶297. Toombs also teaches that irreversible write protection is not always desirable, as Toombs discloses both a permanent write protection function and a temporary write protection function. Ex. 1004 at 12:56-67. Ex. 1006 ¶297.

5. Dependent claims 5, 13, and 14

(i) "At least one part of the memory comprises a memory group having a certain memory size defined in the data register" / "wherein the memory is arranged to comprise at least one memory group" / "wherein the data register is arranged to define a memory size of the at least one memory group"

The '370 Patent acknowledges that the concept of a "memory group having a certain memory size" is known to a POSITA. Ex. 1006 ¶300, 332. The '370

Patent states that "the segment size to be protected is defined in the units of WP_GRP_SIZE groups as specified in the CSD…register" and that "[t]he write protection of the addressed write-protect group is then done …." Ex. 1001 at 1:58-62.

Toombs discloses a MMC card comprising memory groups:

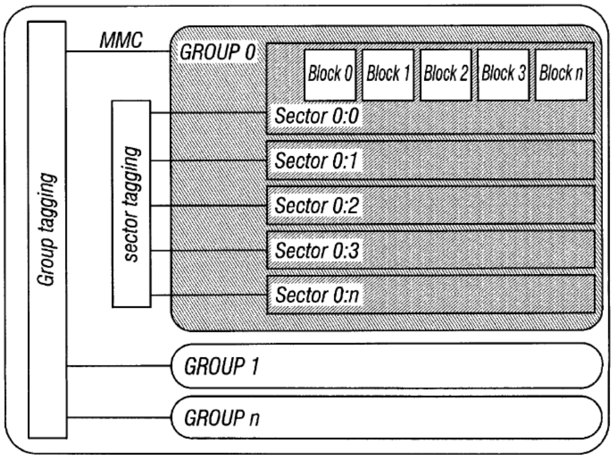


FIG. 66

Ex. 1004 at FIG. 66.

According to Toombs, "the MultiMediaCard card is divided into n memory groups." *Id.* 27:37-38. "Each of the memory groups is subdivided into a plurality of sectors." *Id.* 27:38-39. "Further, each of the sector [sic] comprises of a plurality of memory blocks." *Id.* 27:39-40. Toombs further discloses that "the group size is a configurable parameter" and that "[t]he actual size is stored in the CSD register." *Id.* 27:57-60.

Toombs further discloses write protection being "applied to memory groups." *Id.* 29:45-46. Toombs shows a memory hierarchy for a write protection mechanism:

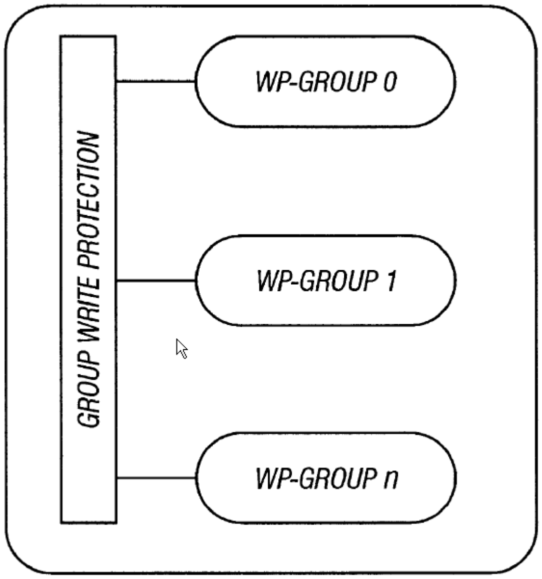


FIG. 69

Id. FIG. 69. Toombs explains: "[t]he WP-Group is the minimal unit which may have individual write protection" and that "[i]ts size is the number of groups which will be write protected by one bit." *Id.* 29:58-60. Toombs further discloses that "[t]he size of a WP-group is a configurable parameter" and that "the actual size [of a WP-group] is stored in the CSD register." *Id.* 29:60-62. According to Toombs, "[f]or cards which support write protection of groups of sectors…portions of the data may be protected (in units of WP_GRP_SIZE sectors as specified in the CSD)." *Id.* 30:3-7.

It would have been obvious to a POSITA to combine the memory hierarchy of Toombs with the use of register bits to define the meaning of a write protection command of Chevallier to achieve permanent write protection of memory groups.

Ex. 1006 ¶305, 335. One would have been motivated to do so because the introduction of Toombs' memory hierarchy to Chevallier would reduce the number of bits in Chevallier's control register required to specify each portion of the memory to protect. *Id.* ¶306, 336. For example, for the memory illustrated by Figure 2 of Chevallier, which comprises thirty-two memory blocks, specifying each part of the memory to write protect would require a control data word comprising at least thirty-two bits, each corresponding to one memory block. *Id.* On the other hand, if for example the thirty-two memory blocks are grouped into four memory groups each comprising eight memory blocks, the control data word would only need to be four-bits long to specify all possible combinations of the groups. *Id.*

Alternatively, a POSITA would have been motivated to improve Toombs' teaching of write protecting memory groups with Chevallier's disclosure of using a register bit to redefine a command so that the command applies permanent write protection to specified memory groups. Ex. 1006 ¶307, 337. The secure function bit, as disclosed by Chevallier, may be placed in one or more unused addresses of the CSD register of Toombs or another suitable register. *Id.* The resulting combination would introduce the permanent write protection of Chevallier to the memory groups and their size definitions specified in the CSD of Toombs. *Id.* Moreover, the combination would achieve permanent write protection of specified memory groups rather than of an entire card. *Id*.

Finally, it would have been obvious to a POSITA to store the secure function bit and the size of the memory group to be write protected in the same register. Ex. 1006 ¶308, 338. That data relates to the memory to be write protected, and thus it would be an obvious design choice to store the same type of data in the same register. *Id*. It would also provide the benefit of keeping information about write protection organized in the same data structure. *Id.* In fact, both Chevallier and Toombs suggest storing write-protection information and memory-size information in the same register (*e.g.*, Ex. 1003 ¶0039; CL. 4; Ex. 1004 at Fig. 17B; 2:50-51). Ex. 1006 ¶308, 338.

6. Dependent claim 6

(i) Wherein redefining the command allows permanent write protection of each memory group individually

Toombs discloses dividing a memory into memory groups. Section C.5; Ex. 1006 ¶¶301-04, 311. Toombs teaches "a method of providing write protection to any combination of memory groups and sectors in the MultiMediaCard system." Ex. 1004 at 1:57-59. Toombs further discloses that "[e]ach WP-group has an additional write protection bit," which "can be programmed via special commands." *Id.* 29:65-67. The "special command[]" is a SET_WRITE_PROT command, which "sets the write protection of the addressed write-protect group." *Id.* 30:9-10. Toombs therefore discloses temporarily write protecting each memory group individually. Ex. 1006 ¶311.

As shown above in Section C.5, a POSITA would have been motivated to improve the system disclosed in Chevallier with the group-based memory hierarchy disclosed in Toombs or to introduce to Toombs' system the ability to permanently write protect individual memory groups. Ex. 1006 ¶¶306, 312. Moreover, a POSITA would have been motivated to combine Chevallier and Toombs in order to allow a user to selectively identify which memory groups to permanently write protect. *Id.* ¶307, 312.

7. Dependent claim 7

(i) A method according to claim 1, wherein the memory is included on a multimedia card (MMC).

Toombs shows a MMC bus system with MMC cards:

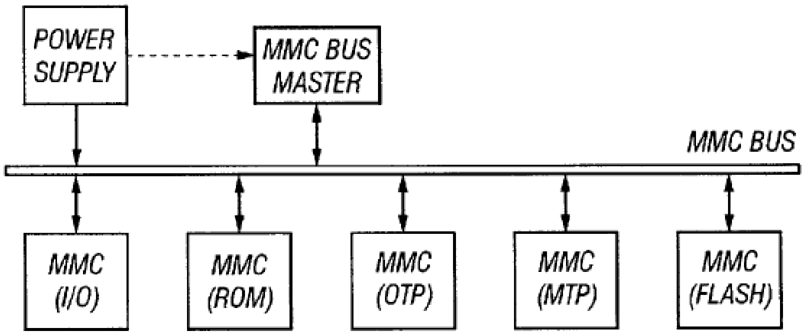


FIG. 4

Ex. 1004 at FIG. 4.

According to Toombs, "the MultiMediaCard bus connects the MultiMediaCard cards each comprising various solid-state mass storage devices, or I/O devices." *Id.* 6:61-65. Toombs discloses "allow[ing] the MultiMediaCard card to write protect any combination of groups of the memory." *Id.* 29:40-42.

Chevallier contemplates Flash memory devices. Ex. 1003 ¶0002. As Toombs discloses, a MMC can comprise Flash memory. Ex. 1006 ¶318. A POSITA would have found it obvious to include the Flash memory of Chevallier in a MMC form factor disclosed by Toombs. *Id.* A POSITA would be motivated to include the Flash memory in a memory card compliant with the MMC standard since such cards are in demand for various consumer applications such as "PDAs, cameras, smart phones…" Ex. 1004 at 1:20-23. Ex. 1006 ¶318. Indeed, Chevallier also identifies "[c]ommon uses for Flash memory" including "(PDAs), digital cameras, and cellular telephones." Ex. 1003 ¶0004. Ex. 1006 ¶318. The teachings of Chevallier and Toombs would yield the predictable result of: (1) the Flash memory of Toomb's MMC being permanently write protectable; and (2) Chevallier's memory device being part of a MMC. Ex. 1006 ¶318. Moreover, a MMC including Flash memory provides the benefit of allowing "software and data [to] be preloaded and changed by the [] host." Ex. 1004 at 1:31-33. Ex. 1006 ¶318.

8. Independent claim 12

Toombs discloses an interface controller that interfaces with an MMC's data registers, memory core, and a host. *Compare* Ex. 1004 Fig. 3 and Ex. 1001 Fig. 1. Ex. 1006 ¶188.

(i) An interface controller arranged to write protect at least one part of a memory of said apparatus by a command

Fig. 14 and 23:46-51 of Toombs illustrate that Toombs' interface controller processes the commands executable on Toombs' MMC (which are sent by a host over a CMD line), including write protect commands. Ex. 1004 Fig. 4 (CMD 28), 30:8-12 (SET_WRITE_PROT command). Ex. 1006 ¶322. One of ordinary skill would have been motivated to introduce Toombs' interface controller to the memory device of Chevallier in order to have a built-in controller that could interface with a host and execute Chevallier's write protection commands (*i.e.*, the lock command and secure command). Ex. 1006 ¶322. Moreover, introducing Toombs' interface controller to Chevallier's memory device would have had the predictable result of a controller that interfaces with a host to process commands for performing functions on an array of memory. Ex. 1003 ¶0044, Fig. 6. Ex. 1006 ¶322.

(ii) A controller arranged to set the at least one bit in order to redefine the command to allow permanent write protection that cannot be un-protected by a command, of the at least one part of the memory of said apparatus

Figs. 17A-B and 10:1-4 of Toombs illustrate that the data registers of Toombs (specifically, the CSD register) contains the bit(s) controlling permanent and temporary write protection of the MMC (Ex. 1004 at Figs. 17A-B, 12:56-67 (PERM_WRITE_PROTECT and TMP_WRITE_PROTECT bits)). Ex. 1006 ¶323.

One of ordinary skill would have recognized that Fig. 14 of Toombs illustrates that the MMC interface controller is the only controller that interfaces with the CSD, and thus the interface controller set the bits related to write protection. Ex. 1006 ¶324. One of ordinary skill would have been motivated to introduce Toombs' interface controller to the memory device of Chevallier in order to have a built-in controller that could interface with a host and set Chevallier's secure function bit (*i.e.*, execute the steps of Chevallier's method in Fig. 5). *Id.* ¶323. Moreover, introducing Toombs' interface controller to Chevallier's memory device would have had the predictable result of a controller that interfaces with a host to control operations and settings of a memory array, including setting a secure function bit in a data register. Ex. 1003 ¶¶0039, 0044, Fig. 6. Ex. 1006 ¶323.

(iii) **Setting at least one bit in order to redefine the command to allow permanent write protection** and/or **a data register**

These limitations would have obvious over the Chevallier-Toombs combination. Ex. 1006 ¶¶280-85, 324-27. Sec. C.1.

9. Dependent claim 15

Wherein the controller is arranged to define the command to allow permanent write protection of the at least one memory group individually would have been obvious over the Chevallier-Toombs combination. *See,* Sec. C.5-6 and Ex. 1004 Fig. 4 CMD 28 (explaining that the SET_WRITE_PROTECT command is defined to protect the memory group(s) identified by the

WP_GRP_SIZE bit in the CSD, and explaining rationales for combining Toombs with Chavllier) and Sec. C.8 (explaining that execution of the SET_WRITE_PROTECT command and interfacing with the CSD functions are performed by Toombs' interface controller). Ex. 1006 ¶341. In sum, Toombs' interface controller defines the value of the WP_GRP_SIZE bit in the CSD. The WP_GRP_SIZE bit is referenced by the SET_WRITE_PROTECT command executed by Toombs' controller, and thus the Toombs' controller defines the SET_WRITE_PROTECT command (via the WP_GRP_SIZE bit) to protect the specified memory group(s), including each WP-group individually. *Id.*

10. Dependent claim 16

(i) Wherein an additional data register is arranged to control existence and characteristics of the at least one part of the memory

Toombs discloses that "each of the cards of the MultiMediaCard system comprises a group of registers for storing a variety of status and internal information." Ex. 1004 at 9:46-48. Toombs discloses an embodiment, in which the information is stored in five registers including OCR, CID, CSD, RCA, and DSR. *Id.* 9:51-53. Different registers store different information—some registers carry out card/content specific information, while other store configuration parameters. *Id.* 9:56-59.

Toombs also teaches that "[t]he CSD register is responsible for providing information to the MultiMediaCard host on how to access the card content" and that "the CSD register stores values defining the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used, etc." Ex. 1004 at 10:24-29. Furthermore, Toombs shows what when the memory is divided into blocks, sectors, and groups, the sizes of those units are stored in the CSD. *Id.* 27:43-62; FIG. 66. Toombs discloses that each field is assigned its own memory addresses. *Id.* FIGs. 17A-B, "CSD_SLICE" column. Thus, each field can be assigned to "a portion of memory containing information about the memory card" that is "distinct from" a "memory portion containing the bit indicating permanent write protection," *i.e.*, PERM_WRITE_PROTECT at memory bit 11 in Figure 17B. Ex. 1006 ¶345.

The '370 Patent asserts that its EXT_CSD register (which is arranged to "control existence and characteristics of [] at least one part of the memory") has a similar structure to that of the CSD register. A POSITA would understand that Toombs therefore teaches a register so arranged, for example by disclosing that the CSD register stores information related to access to the memory, data format, and sizes of subparts of the memory. Ex. 1006 ¶346.

It would have been obvious to a POSITA to combine the teachings of the CSD register in Toombs with the teachings of the functionalities of the control register in Chevallier to result in one or more data registers having part or all of the functionalities disclosed. Ex. 1006 ¶347. The combination would have produced a predictable result since Chevallier discloses that "[s]ome of the control registers are used for typical control functions and others are reserved for expansion and/or future use." Ex. 1003 ¶0045. Ex. 1006 ¶347. A POSITA would have recognized that one or more functionalities of the CSD register in Toombs would fall under the typical control functions of the control registers in Chevallier. Ex. 1006 ¶347. It would be obvious to implement one or more other functionalities of the CSD register in the registers reserved for expansion and/or future use in Chevallier. *Id*. A POSITA would have been motivated to combine Toombs with Chevallier because information related to, for example, memory access, data format, or block size would be necessary for a host to access and control the memory device disclosed in Chevallier. *Id*.

A POSITA would have found it obvious to implement the control register storing the secure function bit, the secure command, and the control data word separately from a data register identifying other information about the memory card. Ex. 1006 ¶348. Both Toombs and Chevallier disclose multiple data registers Ex. 1004 at 9:51-53; Ex. 1003 ¶0045. Ex. 1006 ¶348. A POSITA would have found it an obvious matter of design choice to implement multiple registers to store different control information. Ex. 1006 ¶348. Moreover, a POSITA would have recognized that doing so brings organization to the system structure by permitting some registers to have certain functions (*e.g.*, identifying the write protection to be applied to memory groups) while other registers are devoted to other functions (*e.g.,* configuration parameters). *Id.* In addition, it would have been obvious to a POSITA to store information related to write protection in one data register and other information about the memory in another data register (as Toombs discloses at, *e.g.*, Ex. 1004 at 9:51-53.) in order to organize the information. Ex. 1006 ¶348. Thus, the combination of Toombs and Chevallier would result in "an additional data register" that is "arranged to control existence and characteristics of the at least one part of the memory." *Id.*

It would also have been obvious to a POSITA to implement more than one data registers controlling "existence and characteristics" of different parts of the memory. Ex. 1006 ¶349. Toombs discloses "multi type MultiMediaCards (*e.g.*, a ROM—Flash combination)." Ex. 1004 at 30:20-22. A POSITA would have understood that different memory technologies may have different characteristics (*e.g.*, access method, data format, data transfer speed) and recognized that different CSD registers could be used for different memory technologies. Ex. 1006 ¶349. For example, a POSITA would have recognized that a dedicated CSD for each memory technology would provide organizational structure for the CSDs. *Id.* In addition, a POSITA would have recognized that implementing a separate CSD for each type of memory technology would have the predictable result of each CSD containing the information for that particular memory technology. *Id.*

11. Dependent claim 17

(i) Wherein the additional data register is arranged to define access to the at least one part of the memory

Toombs teaches that "[t]he CSD register is responsible for providing information to the MultiMediaCard host on how to access the card content." Section C.10. In particular, Toombs discloses a "supported Card Command Classes (CCC) [field] coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card." Ex. 1004 at 10:63-11:4. In addition, Toombs discloses that the RCA register (which is separate from the CSD register) identifies the address of the memory card so that the host can recognize the card. *Id.* at 13:17-26. A POSITA would recognize that access to a portion of memory card requires access to the card itself. Ex. 1006 ¶354. Finally, as shown above, it would have been obvious to a POSITA to combine the teachings of Chevallier and Toombs to implement a separate data register defining access to at least part of the memory. *Id*. Section C.10. For example, a POSITA would have been motivated to do so in order to provide a host access to Chevallier's memory device. *Id*.

12. Dependent claim 18

(i) An apparatus according to claim 12, wherein the memory is arranged to implement different memory technologies.

Figure 4 of Toombs shows a MMC bus system. Ex. 1004 at FIG. 4.

According to Toombs, "the MultiMediaCard bus connects the MultiMediaCard cards each comprising various solid-state mass storage devices, or I/O devices." *Id.* 6:61-65. Toombs explains that "the MultiMediaCard system is a single master bus with a variable number of slaves" and that "each slave is either a single mass storage card (with possible different technologies such as ROM, OTP, Flash etc) or an I/O card." *Id.* 7:3-8. Furthermore, Toombs discloses that "[t]he write protection may also be useful for multi type MultiMediaCards (*e.g.*, a ROM—Flash combination)." *Id.* 30:20-22. Therefore, Toombs teaches both a memory comprising multiple memory cards with different memory technologies and a memory card comprising memory based on different technologies. Ex. 1006 ¶358.

A POSITA would have been motivated to improve the permanently write protectable memory device disclosed in Chevallier using the technique of implementing a memory using different memory technologies disclosed in Toombs. Ex. 1006 ¶359. For example, Chevallier discloses that a memory device can be used to store both long-term data (*e.g.*, program code, BIOS, firmware) and upgradeable data. Ex. 1003 ¶0004. Ex. 1006 ¶359. A POSITA would have recognized that particular memory technologies (*e.g.*, ROM) may work better with long-term data while other technologies (*e.g.*, Flash) work better with upgradeable data. Ex. 1006 ¶359. Combining the teachings of Chevallier and Toombs would improve the versatility of the memory device for different uses. Ex. 1006 ¶359. In addition, introducing the teachings of Chevallier to Toombs would provide the benefit of permanently write protecting some or all of the memory types on the cards Toombs discloses as having multiple types of memory. Ex. 1006 ¶359. *See, e.g.*, Ex. 1003 ¶0007 ("There is a resulting need in the art to permanently lock memory blocks in a Flash memory device.") Moreover, a POSITA would have recognized that the benefit of Chevallier's solution is not limited to Flash memory. Ex. 1006 ¶359.

Alternatively, a POSITA would have been motivated to improve Toombs' teaching of temporary write protecting memory groups in a multi-type MMC (which includes different memory technologies) with Chevallier's disclosure of using a register bit to redefine a command so that the command applies permanent write protection. Ex. 1006 ¶360.

13. Dependent claim 19

(i) An apparatus according to claim 12, wherein the apparatus is a multimedia card (MMC).

As explained in Section C.7.(i), the Chevallier-Toombs combination renders these limitations obvious. Ex. 1006 ¶¶318, 362-66.

14. Independent claim 25

(i) A memory device having stored thereon instructions that, when executed, perform [the steps of Claim 25]

Toombs discloses an MMC card (*e.g.*, Fig. 14) that includes firmware that stores information necessary for operation of the card. Ex. 1004 at, *e.g.*, 7:59-61. A

POSITA would have understood that Toombs disclosure of a firmware teaches "[a] memory device having stored thereon instructions" for performing the functions of the memory card. Ex. 1006 ¶368. Furthermore, a POSITA would have understood that the firmware would have stored the instructions necessary for its MMC card (*e.g.*, the interface controller) to perform operations on the memory block. *Id.* A POSITA would have been motivated to implement the permanent-write-protection method of Chevallier on the MMC card (which includes the firmware) of Toombs in order to enable permanent write protection of segments of memory on the MMC. *Id.* A POSITA would have recognized that the MMC would still perform the predictable function of storing the instruction necessary for the interface controller to control operation of the MMC. *Id.*

(ii) Setting at least one bit in order to redefine the command to allow permanent write protection and/or **a data register**

These limitations would have been obvious over the Chevallier-Toombs combination. Section C.1. Ex. 1006 ¶¶369-72.

D. Ground 4: Claim 25 is obvious under 35 U.S.C. § 103 over the Chevallier-Toombs-Estakhri combination

To the extent Chevallier and/or Chevallier-Toombs does not teach a memory device storing instructions that perform the function of a card controller, it was obvious to introduce that feature to Chevallier and/or Chevallier-Toombs from Estakhri's disclosure of a memory device having a flash controller (microprocessor circuit) and a storage unit containing the controller's firmware (instructions) . Ex. 1005 Fig. 1; 4:54-59; Ex. 1006 ¶377.

A POSITA would have understood that Estakhri's storage unit could store the instructions for executing the functionality (including the steps of Claim 25) of Chevallier and/or Chevallier-Toombs, and the microprocessor circuit could execute those functions, providing the predictable and beneficial result of a card controller operable to perform its functions. Ex. 1006 ¶378. Sections A.1.(ii)-(iv), C.10; Ex. 1003 Fig. 6, ¶0004.

February 9, 2017

Respectfully submitted,
BAKER BOTTS L.L.P.

/Eliot D. Williams/
Eliot D. Williams (Reg. No. 50,822)
eliot.williams@bakerbotts.com
Jason German (Reg. No. 69,497)
jason.german@bakerbotts.com
1001 Page Mill Road
Building One, Suite 200
Palo Alto, CA 94304
Phone: (650) 739-7500
Facsimile: (650) 739-7699

Brian W. Oaks (Reg. No. 44,981)
brian.oaks@bakerbotts.com
Chris V. Ryan (Reg. No. 54,759)
chris.ryan@bakerbotts.com
98 San Jacinto Boulevard
Suite 1500
Austin, TX 78701
Phone: (512) 322-2500
Facsimile: (512) 322-2501

ATTORNEYS FOR PETITIONER
SANDISK LLC

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition, exclusive of the exempted portions as provided in 37 C.F.R. § 42.24(a), contains no more than 13,999 words and therefore complies with the type-volume limitations of 37 C.F.R. § 42.24(a).

February 9, 2017

/Eliot D. Williams/
Eliot D. Williams

CERTIFICATE OF SERVICE ON PATENT OWNER UNDER 37 C.F.R. § 42.105

Pursuant to 37 C.F.R. § 42.105, the undersigned certifies that on the 9th day of February 2017, a complete and entire copy of this Petition for *Inter Partes* Review under 35 U.S.C. § 311 and 37 C.F.R. § 42.104, and all supporting exhibits were provided via Federal Express, postage prepaid, to the Patent Owner and its known representatives by serving the correspondence address of record for the '370 Patent holder and the patent holder's counsel:

LEE & HAYES, PLLC
601 W. RIVERSIDE AVENUE
SUITE 1400
SPOKANE WA 99201

The undersigned further certifies that a courtesy copy of the complete and entire Petition was provided by electronic service to counsel retained by Patent Owner in the Related Matter identified herein:

MATTHEW D. POWERS
matthew.powers@tensegritylawgroup.com

AARON M. NATHAN
aaron.nathan@tensegritylawgroup.com

STEFANI C. SMITH
stefani.smith@tensegritylawgroup.com

ROBERT L. GERRITY
robert.gerrity@tensegritylawgroup.com

JONATHAN TAMIMI
jonathan.tamimi@tensegritylawgroup.com
TENSEGRITY LAW GROUP, LLP
555 Twin Dolphin Drive, Suite 650
Redwood Shores, CA 94065
Telephone: (650) 802-6000

Respectfully submitted,
BAKER BOTTS L.L.P

/Eliot D. Williams/

Eliot D. Williams (Reg. No. 50,822)
eliot.williams@bakerbotts.com
Jason German (Reg. No. 69,497)
jason.german@bakerbotts.com
1001 Page Mill Road
Building One, Suite 200
Palo Alto, CA 94304
Phone: (650) 739-7500
Facsimile: (650) 739-7699

Brian W. Oaks (Reg. No. 44,981)
brian.oaks@bakerbotts.com
Chris V. Ryan (Reg. No. 54,759)
chris.ryan@bakerbotts.com
98 San Jacinto Boulevard
Suite 1500
Austin, TX 78701
Phone: (512) 322-2500
Facsimile: (512) 322-2501

ATTORNEYS FOR PETITIONER SANDISK LLC