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. 9,063,850

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. Pat. No. 9,063,850 to Hyvönen et al.
1002	Declaration of Dr. R. Jacob Baker
1003	CF+ and CompactFlash Specification Revision 3.0
1004	U.S. Pat. No. 7,478,248 to Ziv et al.
1005	U.S. Pat. No. 6,681,304 to Vogt et al.
1006	U.S. Pat. Publ. No. 2004/0243900 to Henkel
1007	U.S. Pat. No. 7,409,489 to Sinclair et al.
1008	File History for U.S. Pat. No. 8,307,180
1009	The MultiMediaCard System Specification, Version 3.31
1010	U.S. Pat. Publ. No. 2008/0080688 to Burgan et al.
1011	U.S. Pat. No. 5,809,340 to Bertone et al.
1012	U.S. Pat. Publ. No. 2006/0280077 to Suwa et al.
1013	File History for U.S. Pat. No. 9,063,850
1014	U.S. Pat. Publ. No. 2002/0087817 to Tomaiuolo et al.
1015	Declaration of Stephen Gross
1016	Affidavit of Christopher Butler, Internet Archive
1017	Appendix A to the Affidavit of Christopher Butler
1018	*CompactFlash Association Announces Availability of Revision 3.0 of the CF+ & CompactFlash Specification; Revision 3.0 Increases CF*

Ex.	Description
	Interface Data Transfer Rate to 66MB/sec, BUSINESS WIRE (Jan. 7, 2005)
1019	U.S. Pat. Publ. No. 2007/079015 to Royer, Jr. et al.
1020	Second Affidavit of Christopher Butler, Internet Archive
1021	Appendix A to the Second Affidavit of Christopher Butler
1022	*MultiMediaCard Association (MMCA) and the JEDEC Solid State Technology Association (JEDEC) Announce eMMC for Embedded Flash Memory Applications*, JEDEC (Dec. 20, 2006)

I. INTRODUCTION

Petitioner SanDisk LLC requests *inter partes* review of claims 10 and 12-18 of U.S. Patent No. 9,063,850 (the "'850 Patent") entitled "Extended Utilization Area for a Memory Device." *See* Ex. 1001.

II. MANDATORY NOTICES, STANDING, AND FEES

Real Parties in Interest: 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, and SanDisk China Limited.

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

The '850 Patent is also subject to an ITC action entitled In the Matter of Certain Flash Memory Devices and Components Thereof, Inv. No. 337-TA-1034 ("ITC Matter"). Petitioner is a respondent in this investigation.

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) and Ebby Abraham (Reg. No. 73,399) 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) 739-7609. Petitioner consents to service by electronic mail at eliot.williams@bakerbotts.com, brian.oaks@bakerbotts.com, and ebby.abraham@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 U.S. Patent No. 9,063,850 is available for *inter partes* review. Petitioner is not barred or estopped from requesting *inter partes* review of any claim of the '850 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. 02-0384, Ref. No. 083480.0106, as well as any additional fees due in connection with this Petition.

II. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

Petitioner requests *inter partes* review and cancellation of Claims 10 and 12-18 (the "Challenged Claims") of the '850 Patent (Ex. 1001) based on the following

grounds.

Ground	'850 Patent Claims	Basis for Rejection
1	10, 12, and 18	§ 102 based on *CompactFlash*
2	14-17	§ 103 based on *CompactFlash* and *Henkel*
3	10, 12, and 18	§ 103 based on *Ziv and Vogt*
4	13	§ 103 based on *Ziv, Vogt, and* eMMC
5	14-17	§ 103 based on *Ziv, Vogt,* and *Henkel*
6	10, 12-13, and 16-18	§ 102 based on *Sinclair*
7	13	§ 103 based on *Sinclair* and eMMC
8	14-15	§ 103 based on *Sinclair* and *Henkel*

III. A PERSON OF ORDINARY SKILL IN THE ART.

Relative to the '850 Patent, a person of ordinary skill in the art is a person having at least a bachelor's degree in Electrical Engineering, Computer Engineering, or equivalent training, with at least two to three years of academic or industry experience in the field of memory system design. Ex. 1002 at ¶65.

IV. BACKGROUND OF THE TECHNOLOGY

A. Technical Background

A memory device is used to store electronic data. Ex. 1002 at ¶68. By 2008, several types of memory devices that use flash or EEPROM memory were in existence, including MultiMediaCard ("MMC") and CompactFlash ("CF"). *Id*. An MMC can communicate with a host device. Ex. 1009 at 19. For example, an MMC receives commands from a host device over a "CMD" bus and communicates data over the "DAT" bus line. Ex. 1002 at ¶74; *see* Ex. 1009 at 142, Fig. 4 (shown

below).

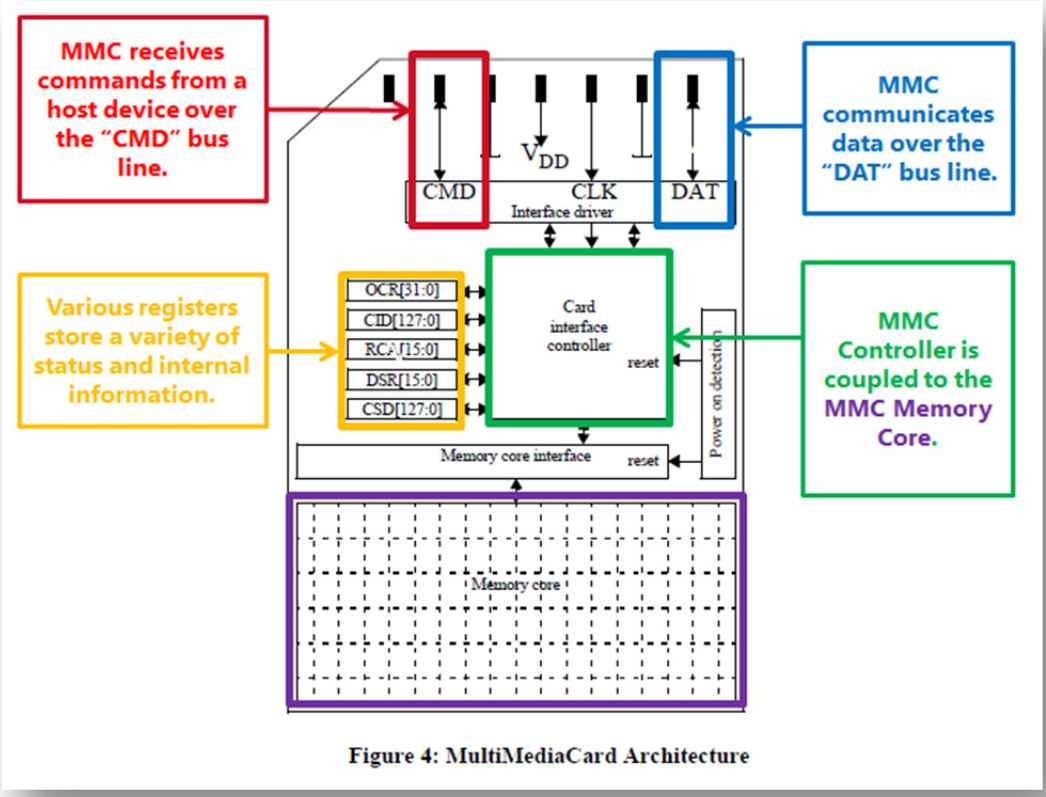


Figure 4: MultiMediaCard Architecture

Ex. 1009 at Fig. 4 (annotated).

V. OVERVIEW OF THE '850 PATENT

The '850 Patent issued on Nov. 6, 2012 from U.S. Patent Application No. 13/951,169 ("'169 Application") filed on Jul. 25, 2008. The '850 Patent claims priority to U.S. Patent Application No. 12/039,672, which was filed on Feb. 28, 2008.

A. Summary of the Claimed Subject Matter

The Challenged Claims generally relate to activating access profiles and

configuring memory devices in accordance with the active access profile. The access profile "governs the current access operations to the memory device." Ex. 1001 at 4:63-64. The access profiles correspond to memory access operations, such as "a read, a write, an erase, and a modify attribute operation." Ex. 1001 at 2:1-3, 5:7-9, 6:25-27.

The '850 Patent specification describes the invention as applicable "in any stand-alone or embedded system that comprises or accesses a memory device." *Id.* at 5:30-33. When connected to such a system, the memory device can "receive one or more commands ... for activating a particular access profile." *Id.* 4:35-38. The system is also described as being able to "issue commands for configuring the memory device in accordance with an access profile." *Id.* at 3:44-50. The portion of the system that issues commands to the memory device is referred to as a "host." *Id.* at 2:65-66.

The '850 Patent specification also describes configurations that may correspond to access profiles. For instance, in a "burst profile mode, [an access profile,] corresponding to fast, contiguous data access," the memory device is configured so that "after receiving all the data" to be written from a host, it may "indicate 'exit busy' and set the transfer mode to 'transfer state,' thus facilitating faster execution of subsequent accesses by the host." *Id.* at 7:21-26.

Other access profiles may cause the device to configure itself such that a

particular profile is "associated with different partitions of the memory device," including either "logical or physical partitions." *Id.* at 7:37-42. Similarly, a profile may result in a device configuration such that access operations are "map[ped] … to certain sections of the physical memory with special characteristics." *Id.* at 7:54-57. Finally, some access profiles may result in a configuration such that the device postpones "management" or "background operations" until after the data transfer. *Id.* at 4:1-4; 4:52-56.

In addition, the memory controller may conduct or interleave simultaneous memory access operations. *Id.* at 6:58-61. In the event that the memory access operations conflict with each other, the controller may designate access priority levels to resolve the conflict. *Id.* at 2:32-34.

B. The '850 Patent Prosecution History

The '850 Patent issued from the '169 Application. Before issuance, Applicant responded to three office actions. Ex. 1013 at 238, 440, 495.

The first and second office action rejected a set of claims that Applicant eventually cancelled. Prior to the first office action, Applicants asserted the following exemplary claim:

1. (Currently Amended) A method ~~for configuring access to a memory device,~~ comprising:

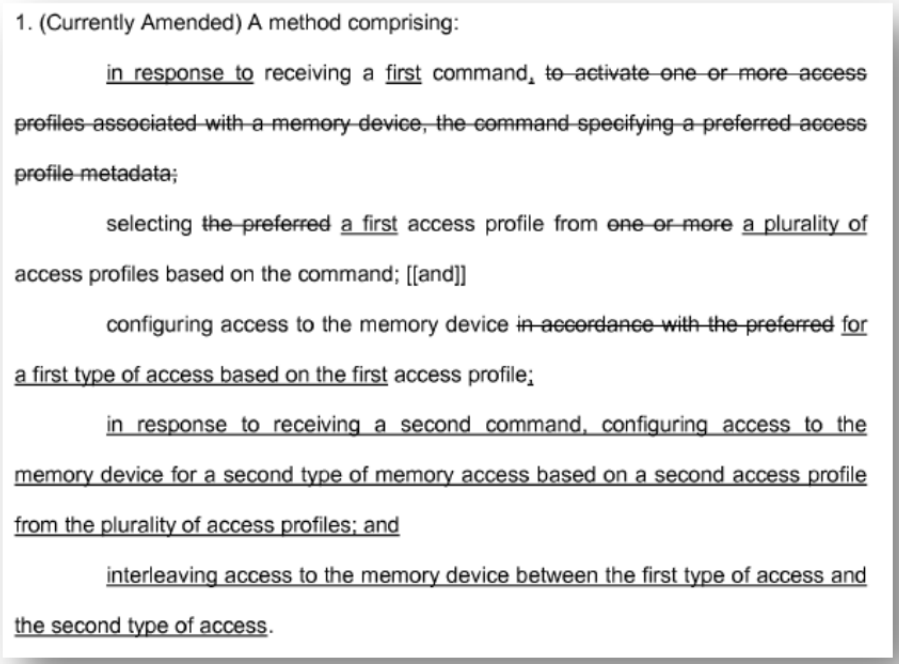
receiving ~~one or more commands for activating~~ a command to activate one or more access profiles associated with [[the]] a memory device, the command specifying a preferred access profile metadata;

selecting the preferred access profile from one or more access profiles based on the command; and

configuring access to the memory device in accordance with ~~at least one of the access profiles~~ the preferred access profile.

Id. at 594 (displaying presented claims after a preliminary amendment).

The Examiner rejected the claims over Suwa (*Suwa*) and Burgan (*Burgan*). In response to the rejection, the Applicant amended the claims and argued that the references do not teach the amendment. The following is an exemplary amendment submitted by the Applicant:



1. (Currently Amended) A method comprising:

in response to receiving a first command, ~~to activate one or more access profiles associated with a memory device, the command specifying a preferred access profile metadata;~~

selecting ~~the preferred~~ a first access profile from ~~one or more~~ a plurality of access profiles based on the command; [[and]]

configuring access to the memory device ~~in accordance with the preferred~~ for a first type of access based on the first access profile;

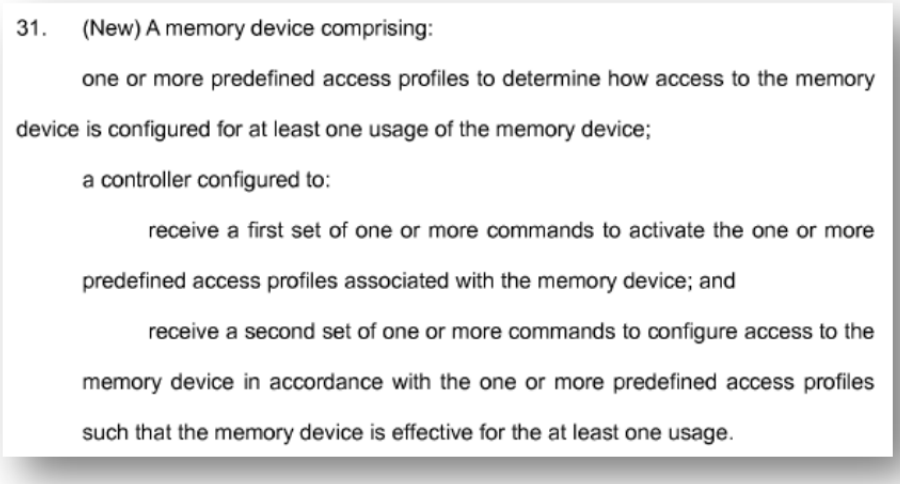
in response to receiving a second command, configuring access to the memory device for a second type of memory access based on a second access profile from the plurality of access profiles; and

interleaving access to the memory device between the first type of access and the second type of access.

Id. at 496.

The Examiner issued a Final Office Action, allowing dependent Claims 12-14 if rewritten in independent form and rejecting the Claims 1-4, 6-11, and 15-21 under 35 U.S.C. §§ 103 and 112. For the § 103 rejection, the Examiner rejected the claims over *Suwa* and Tomaiuolo (*Tomaiuolo*). *Id.* at 461. For the § 112 rejection, the Examiner stated that the claims were not adequately described in the specification. *Id*.

The Applicant subsequently cancelled all pending claims and asserted a new set of claims. The origin of Claim 10 of the ’850 Patent can be found in Claim 31 of the ’169 Application (shown below).



31. (New) A memory device comprising:

one or more predefined access profiles to determine how access to the memory device is configured for at least one usage of the memory device;

a controller configured to:

receive a first set of one or more commands to activate the one or more predefined access profiles associated with the memory device; and

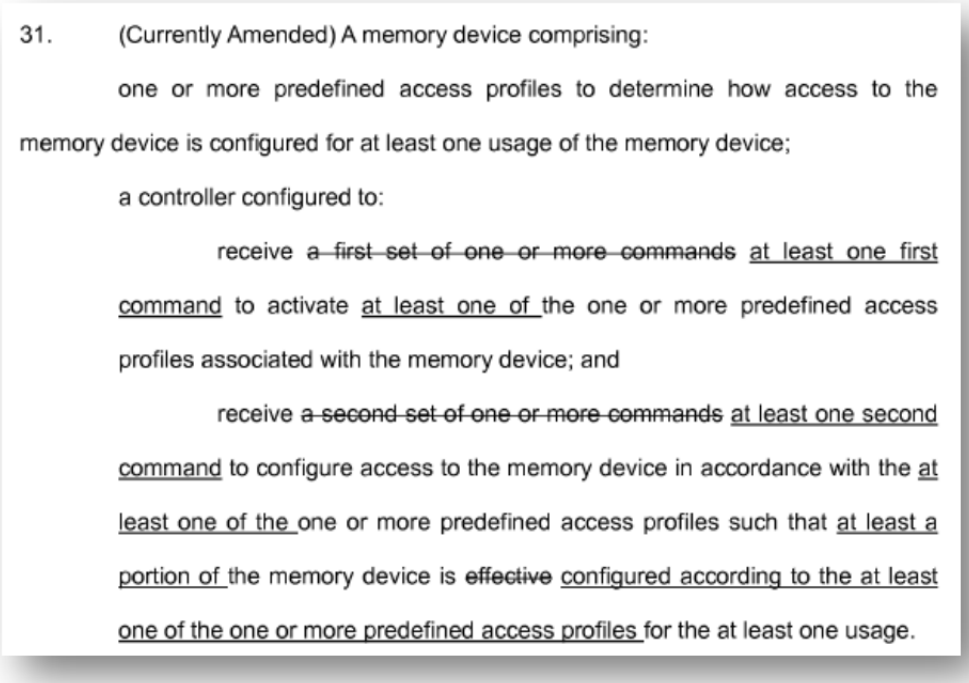
receive a second set of one or more commands to configure access to the memory device in accordance with the one or more predefined access profiles such that the memory device is effective for the at least one usage.

Id. at 443.

The third office action again rejected the claims under 35 U.S.C. § 112. The Examiner stated that the '169 Application did not disclose receiving "a first set of one or more commands" or a "second set of one or more commands." *Id.* at 286.

In response, the Applicants filed a Response and a Supplemental Response[1] amending the claims and arguing that the amended claims render the § 112 rejection moot. *Id.* at 251-52. An exemplary amended claim (that issued as Claim 10 of the '850 Patent) is shown below:

[1] Applicants filed a Supplemental Response to further amend the independent claims. The Supplemental Response removed the claim limitation "for the at least one usage" from the independent claims.



31. (Currently Amended) A memory device comprising:

one or more predefined access profiles to determine how access to the memory device is configured for at least one usage of the memory device;

a controller configured to:

receive ~~a first set of one or more commands~~ at least one first command to activate at least one of the one or more predefined access profiles associated with the memory device; and

receive ~~a second set of one or more commands~~ at least one second command to configure access to the memory device in accordance with the at least one of the one or more predefined access profiles such that at least a portion of the memory device is ~~effective~~ configured according to the at least one of the one or more predefined access profiles for the at least one usage.

Id. at 256.

A Notice of Allowance subsequently issued. *Id.* at 13.

C. The ’180 Patent Prosecution History

The ’850 Patent claims priority to the ‘672 Application, which later issued as U.S. Pat. No. 8,307,180 (“’180 Patent”). As such, the prosecution history of the ’180 Patent is relevant to the ’850 Patent. *Ormco Corp. v. Align Tech., Inc*., 498 F.3d 1307, 1314 (Fed. Cir. 2007) (“When the application of prosecution disclaimer involves statements from prosecution of a familial patent relating to the same subject matter as the claim language at issue in the patent being construed, those statements in the familial application are relevant in construing the claims at issue.”).

Before issuance of the ’180 Patent, Applicant responded to three office actions and submitted a notice of appeal and a pre-appeal brief. Ex. 1008 at 327, 393, 406, 440. The resulting ’180 Patent issued on Nov. 6, 2012.

The first office action rejected the claims as anticipated by Burgan et al. (*Burgan*). *Id*. at 443. *Burgan* teaches smart phones with various profiles, such as a work profile and a family profile, that are accessed in response to caller ID information of a received call. Ex. 1010 at ¶¶5-6.

Applicant’s response to the first office action is relevant to the “command” claim element. According to Applicant, “***[a] command in the context of the various embodiments of the present application***, and in any other context for that matter, ***suggests some type of authoritative direction or instruction to do/not to do something***.” *Id*. at 427 (emphasis supplied). In other words, a command must provide a type of instruction. Ex. 1002 at ¶¶114, 136. Using this definition, Applicant submitted that a POSITA would not equate the received Caller ID information in *Burgan* with an actual “command,” because a Caller ID is a “passive” identifier. Ex. 1008 at 427.

Unconvinced by the Applicant’s argument, the Patent Office subsequently issued a final rejection on July 6, 2011, maintaining the rejection. *Id*. at 408. In response, Applicant repeated its arguments in a Reply to the Final Office Action. *Id*. at 401. Applicant thereafter filed a notice of appeal and a pre-appeal brief. *Id*. at

382, 393-96. The Pre-Appeal Panel decided to withdraw the rejection and issue a new office action. *Id*. at 357.

The Examiner again rejected the claims over *Suwa* and Bertone et al. (*Bertone*). *Id*. at 329. Applicant made substantial amendments to the claims to distinguish *Suwa* and *Bertone* by further limiting a predefined access profile and the received one or more commands. *Id*. at 259-268. The amendments to Claim 17 are shown below:

17. (Currently Amended) A memory device, comprising:

one or more registers for storing one or more predefined access profiles associated with said memory device, said predefined access profiles being effective for determining how access to said memory device is configured for at least one usage;

~~receiving means~~ a controller for receiving one or more commands related to said at least one usage of said memory device, said one or more commands for activating said one or more predefined access profiles associated with said memory device, said controller also

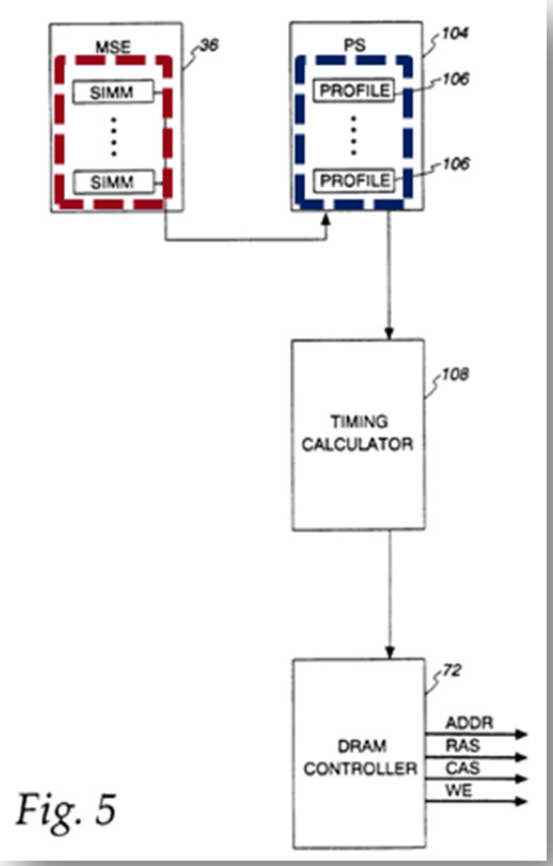
~~; and~~

~~configuring means~~ for configuring access to said memory device in accordance with at least one of said predefined access profiles so that said memory device is effective for said at least one usage.

Id. at 261.

Applicant also clarified the claim limitation "configuring access to said memory device." As illustrated in Fig. 5 of *Bertone* (shown below), *Bertone*

teaches a device that can contain multiple memories (outlined in red) with different characteristics stored in separate profiles (outlined in blue). Ex. 1011 at 16:28-36. The memory device in *Bertone* optimizes the memory device's speed based on the timing characteristics of the multiple memories. *Id*. at 16:28-36. Applicant argued that, unlike the claim requiring "configuring access to said memory device," the memory device in *Bertone* only configures the memory device according to "timing characteristics to control the speed performance of the memory module." Ex. 1008 at 266-67. In other words, because the operating speed does not affect an access operation (*i.e.,* read, write, modify, or erase operation), Applicant argued that merely changing an operating speed for the memory device ***does not*** constitute an access configuration.



Ex. 1011 at Fig. 5 (annotated).

Applicant also contended that an access-mode switcher in *Suwa*, which automatically selects either a random access mode or a sequential mode, fails to configure access to the memory device in accordance with an access profile. Ex. 1008 at 266. Applicant argued that *Suwa* does not configure an access profile, because the switcher in *Suwa* "automatically completes necessary changes to use the selected mode" upon activation. *Id*. at 266. Applicant argued that the claimed invention requires that the activation of an access profile ***occurs separately*** from

the access configuration of the memory device. In other words, the configuration of an access to the memory device and the activation of an access profile ***are distinct steps***. Ex. 1001 at 4:43-46.

A Notice of Allowance subsequently issued. Ex. 1008 at 244.

VI. CLAIM CONSTRUCTION

A. Legal Overview

Because the '850 Patent will not expire during the pendency of these proceedings, the Board should apply the BRI standard in its review. 37 C.F.R. § 42.100(b). For terms not specifically 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.

B. "command"

The Challenged Claims require that a first command is used for activating a predefined access profile associated with the memory device and a second command is used to configure access to the memory device in accordance with the predefined access profile. The Board should construe "command" to mean "*an instruction*." This is consistent with how the term is used in the '850 Patent, the prosecution history of the '850 Patent, its familial prosecution history, and with the ordinary meaning of the term. *See* Ex. 1001 at 3:56-58.

The prosecution history confirms this meaning. *See supra* Section V.B.

Applicant repeatedly stated that "***[a] command*** in the context of the various embodiments of the present application ... ***suggests some type of*** authoritative direction or ***instruction*** to do/not to do something." Ex. 1008 at pp. 401, 427 (emphasis supplied).

The specification also consistently discloses the command as an instruction. The '850 Patent describes the various functions of the command as (1) "activating one or more access profiles associated with said memory device;" (Ex. 1001 at 1:63-65) (2) "designating a preferred access profile;" (*id*. at 2:14-17) (3) "configuring the memory device in accordance with an access profile;" (Ex. 1001 at 3:47-52) (4) "suspend[ing the] current access profile;" (*id*. at 6:9-13) and (5) "revert[ing] back to the access profile" (*id*. at 6:13-18). These cited functions are consistent with the proposed construction. Ex. 1002 at ¶135.

Accordingly, a POSITA would understand that "command" to mean "an instruction."

VII. SUMMARY OF THE PRIOR ART

The prior art teaches the purportedly inventive feature of a memory device receiving a first command to activate a predefined access profile a second command to configure access to the memory device in accordance with the predefined access profile.

A. CompactFlash

A CompactFlash device is a flash memory mass storage device. Ex. 1002 at ¶139. SanDisk first manufactured a CompactFlash device in 1994. Ex. 1014 at 1:56-59. CompactFlash quickly became the go-to portable mass storage device for electornic devices. Ex. 1002 at ¶140. CompactFlash remains popular and is supported by many devices. *Id*.

In 1995, the CompactFlash Association was formed by a group of international companies with the goal of creating an industry standard for flash-based mass storage. *Id*. at ¶141. The CompactFlash specification establishes the methods, processes, and practices for both the CompactFlash device and a host interacting with the device. *Id*. at ¶142.

On December 23, 2004, the CompactFlash Association released CompactFlash Specification Revision 3.0 (*CompactFlash*). *CompactFlash* was made publicly available to any interested member of the public free of charge from the following CFA website prior to 2008. Ex.1015 (Declaration of Stephen Gross) at ¶¶6-10 (attesting to the public availability of *CompactFlash*). The CompactFlash Association publicly announced on January 6, 2005 that *CompactFlash* "is available to download free from the CFA website at http://www.compactflash.org. Ex. 1017 at 9; Ex. 1016 at ¶6 (authenticating page 9 of Ex. 1017 as an accurate representation of the January 6, 2005 announcement by the CompactFlash

Association); *see also* Ex. 1018 (Jan. 7, 2005 article stating that "[t]he CF Specification Revision 3.0 is available to download free from the CFA web site at http://www.compactflash.org"); Ex. 1019 at ¶3 (describing an exemplary hard drive is "described in the CF+ and CompactFlash Specification Revision 3.0, published by the CompactFlash Association, Palo Alto, Calif., Dec. 23, 2004, http://www.compactflash.org.") (published on Apr. 5, 2007).

Moreover, *CompactFlash* was freely and publicly available online from the CompactFlash website as of at least Jan. 13, 2005. Ex. 1015 (Declaration of Stephen Gross) at ¶8 (attesting that a free download of *CompactFlash* was available after completing a registration form); Ex. 1017 at 2-3 (displaying the CompactFlash homepage indicating that *CompactFlash* "is now available to download" on Jan. 13, 2005); Ex. 1017 at 6 (displaying the registration form that once submitted included a link for a free download for *CompactFlash*); Ex. 1016 at ¶6 (authenticating pages 2-3 of Ex. 1017 as an accurate representation of the CompactFlash Association website on Jan. 13, 2005 and page 6 of Ex. 1017 as an accurate representation of the CompactFlash registration form on January 27, 2005). *See Crestron Electronics Inc. v. Intuitive Building Controls Inc.,* Case IPR2015-01460, slip op. at 12-22 (PTAB Jan. 14, 2016) (Paper 14). Accordingly, *CompactFlash* qualifies as prior art at least under pre-AIA 35 U.S.C. §§ 102(a) and 102(b). *CompactFlash* was not previously presented to the PTO in the context of

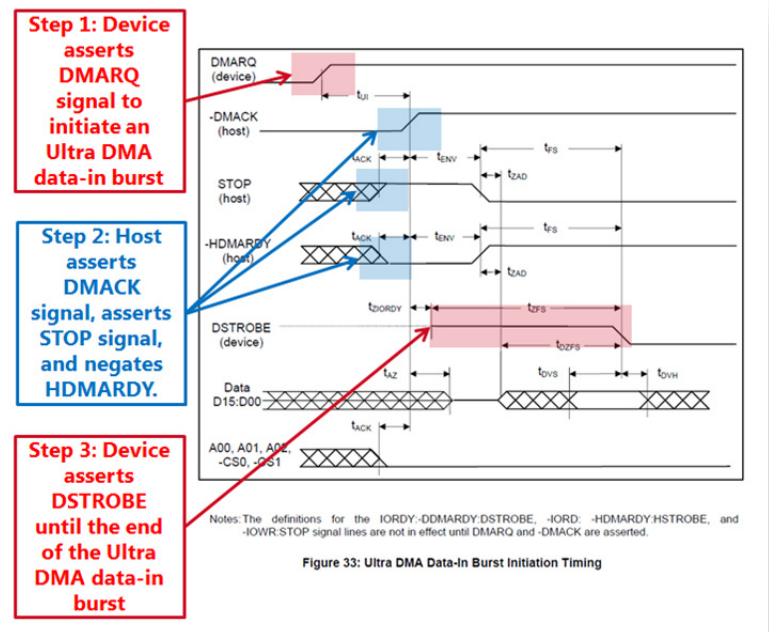
the ’850 Patent.

The CompactFlash Association made improvements to direct memory access (DMA) data transfer in Revision 3. A DMA data transfer occurs directly between the hardware subsystem and the memory device, rather than involving the host’s CPU as an intermediary. Ex. 1002 at ¶145. In this manner, transfer speeds are boosted and the CPU is freed to work on other tasks while the transfer occurs. *Id*. at ¶146. Starting with Revision 3.0, “UltraDMA” was introduced, which boosted DMA transfer speed four-fold from the prior “MultiWord DMA” transfer. Ex. 1018 (“Ultra DMA 33 and UltraDMA66 [sic] interface modes will increase the CompactFlash interface data transfer rate to 66 MB/sec.”).

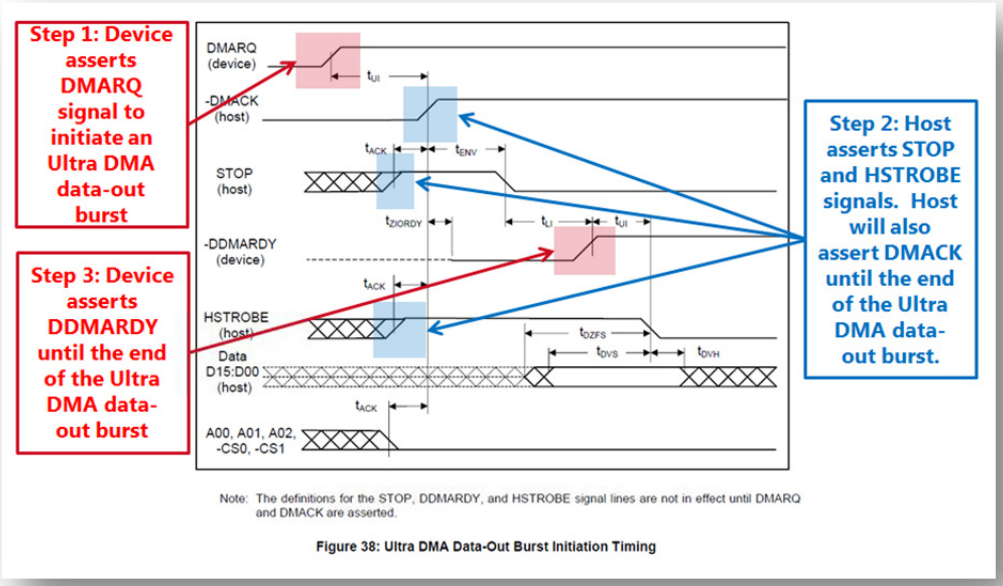
CompactFlash also introduced several Ultra DMA modes. Ex. 1003 at 137. A CompactFlash controller configures a CompactFlash device to perform the Ultra-DMA transfer according to a protocol based on the mode selected. When a host sets an Ultra DMA mode, the memory device automatically disables any enabled MultiWord DMA mode. *Id*. at 158. In response, the memory device sets the selected Ultra DMA mode in a task file register. *Id*. at 120.

The host may subsequently communicate a READ DMA or WRITE DMA command to transfer data to the device. *Id*. at 52. The CompactFlash device, in response, will begin the Ultra DMA-specific initiation protocol to configure the memory device for an Ultra DMA transfer. *Id*. at 68-70. As illustrated for a data-in

burst in Fig. 33 and a data-out burst in Fig. 38 (both shown below), the controller starts the initiation phase of the Ultra DMA burst by asserting a DMARQ signal. *Id*. at 75-76, 81-82. After the host responds by asserting and/or negating several signals, the controller will assert either a DSTROBE signal or DDMARDY signal until the end of the burst. *Id*. at 75-76, 81-82. Finally, unlike the older MultiWord DMA modes, the device will calculate a CRC value to ensure the accuracy of the data transferred in the Ultra DMA modes. *Id*. at 90. If the memory device detects errors, an Error Register is updated to reflect the error. *Id*. at 90.



Id. at Fig. 33 (annotated).



Id. at Fig. 38 (annotated).

B. eMMC

eMMC is an embedded memory module standard promulgated and announced by JEDEC via a press release on December 20, 2006. Ex. 1022. An index describing the press release was publicly available on the JEDEC website no later than June 29, 2007, when the InternetArchive Wayback Machine captured the JEDEC index. Ex. 1021 at 2 (indicating that press release in Ex. 1020 was posted on the JEDEC website on December 20, 2006); Ex. 1020 at ¶6 (authenticating page 2 of Ex. 1021 as an accurate representation of the June 29, 2007 JEDEC index of press releases). The underlying eMMC specification was also made publicly available at the time of the press release. Ex. 1022 (noting that "eMMC™ is the first product standard from the partnership [of JEDEC and the

MultiMediaCard Association]" and that "[a]ll JEDEC standards are available online, at no charge."). *See also* Ex. 1021 at 5 (JEDEC standard policy captured by Wayback backing on July, 3, 2007, noting "JEDEC standards, publications, package outlines and all other documents posted on JEDECs worldwide web site (collectively referred to as the files) may be downloaded free of charge," subject to accepting the terms of JEDEC's copyright statement); Ex. 1020 at ¶6 (authenticating page 5 of Ex. 1021 as an accurate representation of the July, 3, 2007 JEDEC copyright agreement). The public availability of eMMC as prior art against the '850 Patent is corroborated by the file history and the patent specification, where Applicant admitted the prior art status of eMMC. Ex. 1008 at 355; Ex. 1001 2:30-32 (discussing "JESD84 standard for eMMC"); 7:1-4 (discussing "the current JEDEC JC64 eMMC version 4.3 (JESD84) [standard]"). Thus, eMMC is available as prior art under §§ 102(a) and 102(b).

An eMMC module consists of flash memory and a controller, in a small BGA package that can be embedded in a host. Ex. 1022. The eMMC module used the pre-existing MMC standard for host/memory communications. *Id*. This enabled hosts to access "all major classes of mass storage memory subsystems, including embedded memory ... memory cards, or even hard disk drives (via ATA-on-MMC specification) with one common MMC interface…." *Id*.

C. United States Patent Publ. No. 2004/0243900 (*Henkel*)

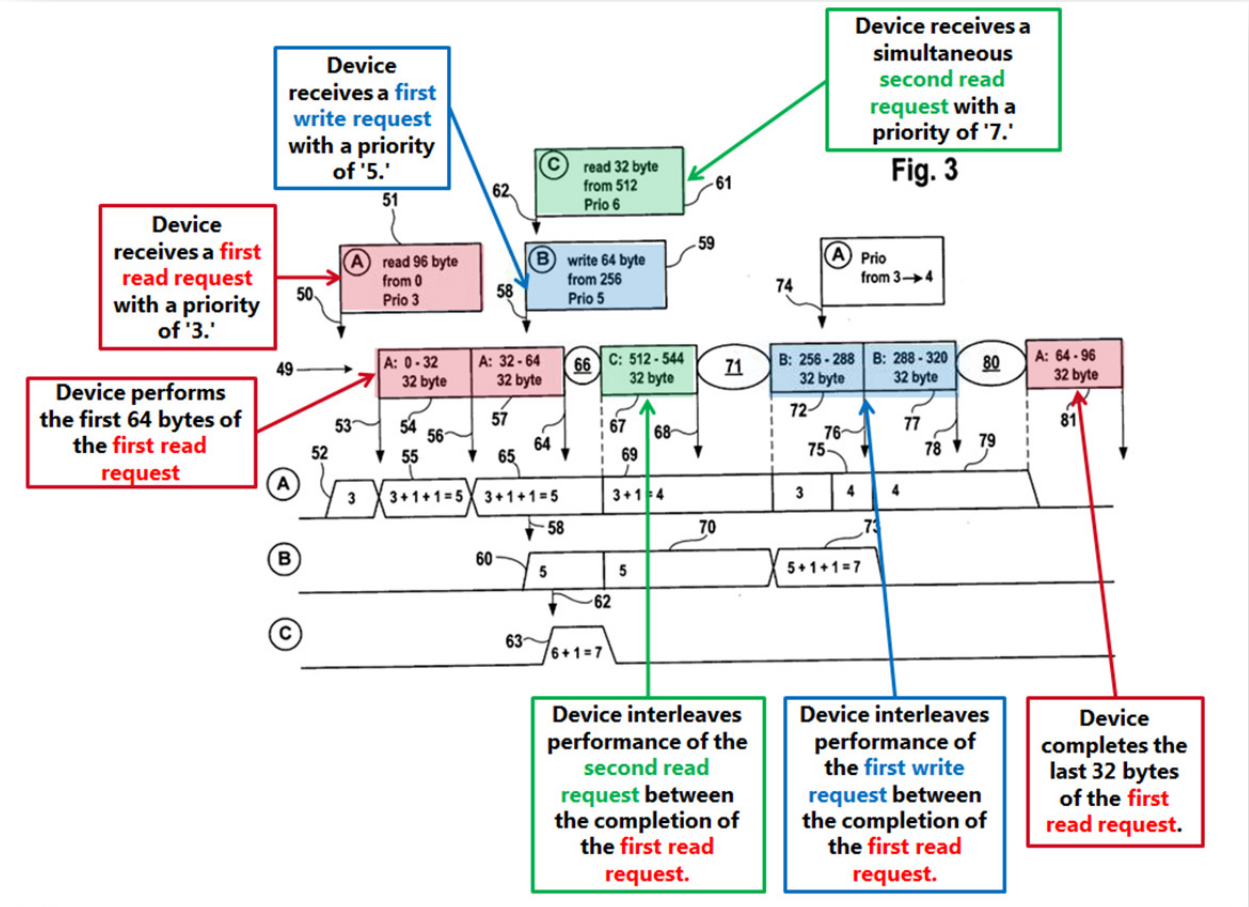
Henkel was filed on Apr. 15, 2004 and published on Dec. 2, 2004. *Henkel* is prior art under at least pre-AIA 35 U.S.C. §§ 102(a), (b), and (e). *Henkel* was not previously presented to the PTO in the context of the '850 Patent.

Henkel is primarily concerned with "controlling accesses to a shared storage[,]" like CompactFlash or MMC memory devices. Ex. 1006 at ¶5. Previous solutions to perform access operations as the memory device receives them are inefficient because "time delays" exist when switching between read-and-write operations and also when performing operations in different memory address ranges. *Id*. at ¶¶15, 56. To reduce these time delays, *Henkel* proposes assigning priorities to access operations such that higher-priority access operations are "processed more quickly" than lower-priority access operations. *Id*. at ¶12.

The prioritization scheme takes into account several factors to avoid time delays. *Id*. at ¶13. For example, the controller in *Henkel* prioritizes memory accesses relating to adjacent address ranges in order to avoid "extra time delays due to re-addressing." *Id*. at ¶13. As another example, the controller in *Henkel* prioritizes memory access requests such that switching between read and write accesses are kept minimal. *Id*. at ¶14. In doing so, the controller minimizes extra delays associated with access operations. *Id*. at ¶14. *Henkel* further acknowledges that certain accesses (*e.g*., a maintenance request) take priority over other access

requests. *Id*. at ¶52.

Henkel provides an example of a prioritization scheme for simultaneous access operations. *See id*. at Fig. 3 (shown below); ¶¶53-67. Initially, a memory device receives a first read request, and the controller assigns an initial prioritization of '3' to the request. *Id*. at ¶54. The memory device segments the request into 32 byte chunks, and increases the priority of the first read request as time passes "in order to promote the transmission of an adjacent block of data." *Id*. at ¶57. Before the memory device can finish processing the first read request, a first write request is received with a prioritization of '5' and a second read request is received with a prioritization of '7.' *Id*. at ¶¶60-61. Because the second read request has the highest prioritization of '7,' the memory device immediately interleaves the second read request between the completion of the first read request. *Id*. at ¶62. Following the completion of the second read request, the memory device then completes the write request and then finally completes the first read request due to the updated priorities associated with each access operation. *Id*. at ¶¶64-67.



Id. at Fig. 3 (annotated).

D. United States Patent No. 7,478,248 (*Ziv*)

Ziv was filed Nov. 27, 2002, and published on May 27, 2004. *Ziv* is prior art under at least pre-AIA 35 U.S.C. §§ 102(a), (b), and (e). *Ziv* was not previously presented to the PTO in the context of the '850 Patent.

Ziv discloses a memory controller that allows profile-based access to an encrypted partition of memory. Ex. 1004 at Abstract. A user first selects a password that is hashed and stored in a register. *Id*. at 4:15-20. The system also

generates a cryptographic key, which is used with a stored address offset for accessing the secure area. *Id*. at 5:14-15.

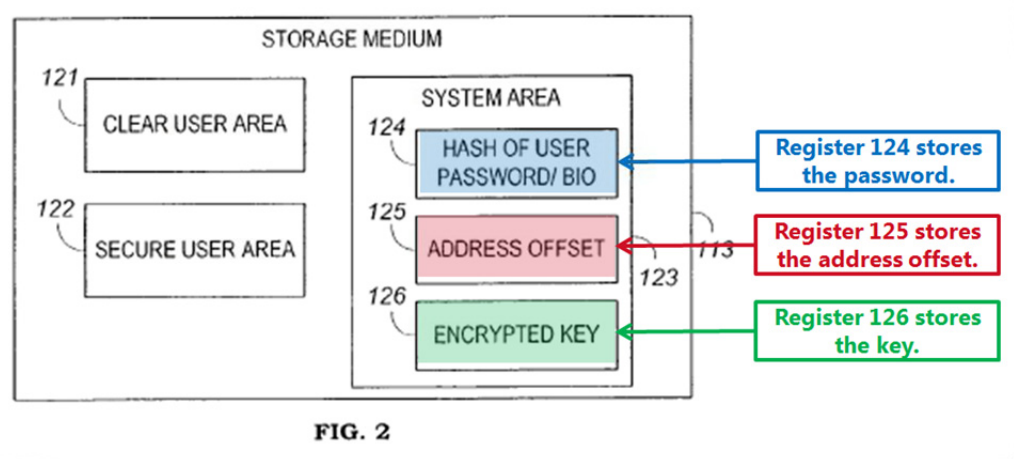
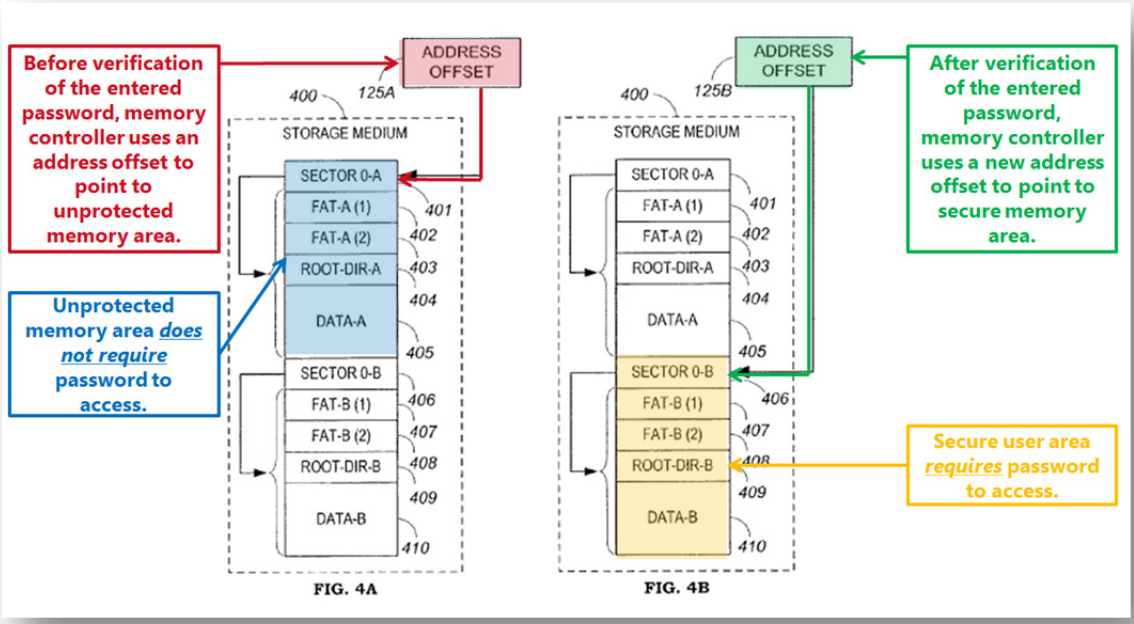


FIG. 2

Id. at Fig. 2 (annotated).

The secure memory area is activated when the host transmits a valid password to the controller. *Id*. at 6:15-22. Then the memory controller remounts the memory by using the stored address offset in the register to point to the secure partition (as illustrated in Fig. 4B). *Id*. at 6:42-49, Fig. 4B (shown below).



Id. at Figs. 4A-B (annotated).

Next, after receiving a subsequent read or write command, the controller must configure the memory device by encrypting/decrypting the data and using the address offset to point to the proper memory address. *Id*. at 7:29-47. Once completed, the memory device can successfully complete the read or write operation to/from the secure memory area. *Id*. at 7:40-47.

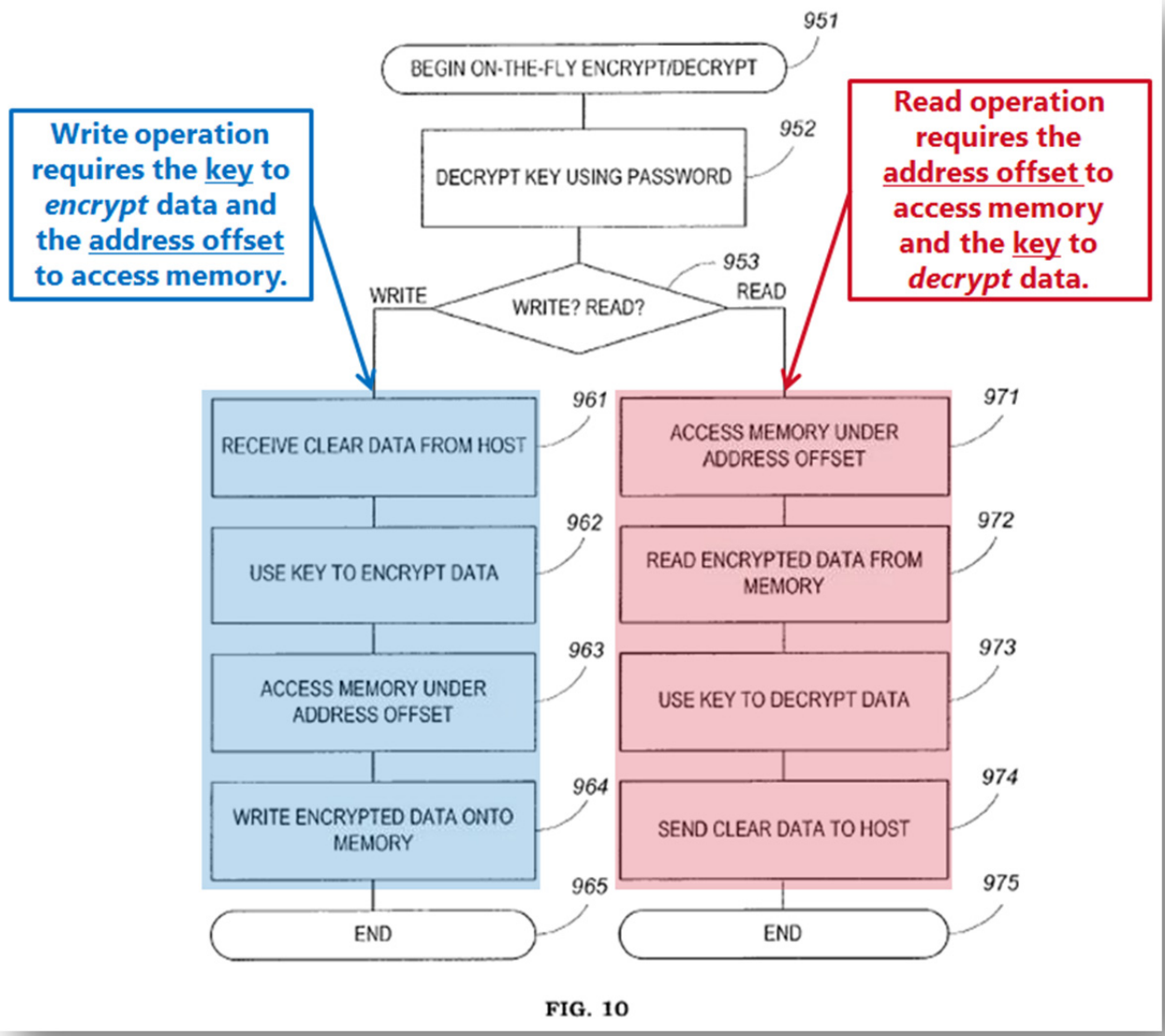


FIG. 10

Id. at Fig. 10 (annotated).

E. United States Patent No. 6,681,304 (*Vogt*)

Vogt was filed Jun. 30, 2000 and issued on Jan. 20, 2004. *Vogt* is prior art under at least pre-AIA 35 U.S.C. §§ 102(a), (b), and (e). *Vogt* was not previously presented to the PTO in the context of the '850 Patent.

Vogt teaches a "password verify" command to access hidden storage in a

memory device. Ex. 1005 at 6:35-37, 8:11-30, 11:53-55. *Vogt* also teaches implementing the hidden storage in a flash memory embedded in a device. *Id*. at 11:16-19.

Vogt further discloses a specific configuration for a read/write operation accessing a hidden storage area in the memory device. Upon receiving a memory read or write signal that attempts to access a hidden storage area address, the controller must then create a hidden storage read signal or write signal. *Id*. at 4:7-10. To create the hidden storage read/write signal, a Valid_HS_Access signal is logically "ANDed" with the memory read/write signal. *Id*. at 4:7-10. The Valid_HS_Access signal accounts that a valid user password has been entered for that specific password-protected hidden storage area. *Id*. at 3:45-4:6. Moreover, in read operations accessing a hidden storage, data is uniquely sent to a "hidden storage bus out" before transferring to the external data bus. *Id*. at 5:24-28.

F. United States Patent No. 7,409,489 (*Sinclair*)

Sinclair was filed on Oct. 25, 2005, claims priority to U.S. Provisional App. No. 60/705,388 filed on Aug. 3, 2005, and was published on Feb. 8, 2007. *Sinclair* is prior art under at least pre-AIA 35 U.S.C. §§ 102(a), (b), and (e). *Sinclair* was not previously presented to the PTO in the context of the '850 Patent.

Sinclair discloses selectable reclaim operation modes that provide optimizations to the rate that memory reclaim operations (*i.e*.,

background/management operations) occur. Ex. 1007 at Abstract. Reclaim operations convert memory containing obsolete data into writeable memory. *Id*. at 2:9-11. An exemplary reclaim operation mode is the Reclaim Normal mode, which can be activated by the host sending a "Reclaim_normal" command to the memory controller. *Id*. at 23:27-30; 23:47-51. This changes the configuration of the memory device to interject reclaim operations between write commands received from the host. *Id*. at 23:47-51, 2:50-57. The device calculates an optimal interleave ratio of reclaim operations to write commands such that the memory card will run out of writeable memory only when no reclaimable memory remains. *Id*. at 2:50-57.

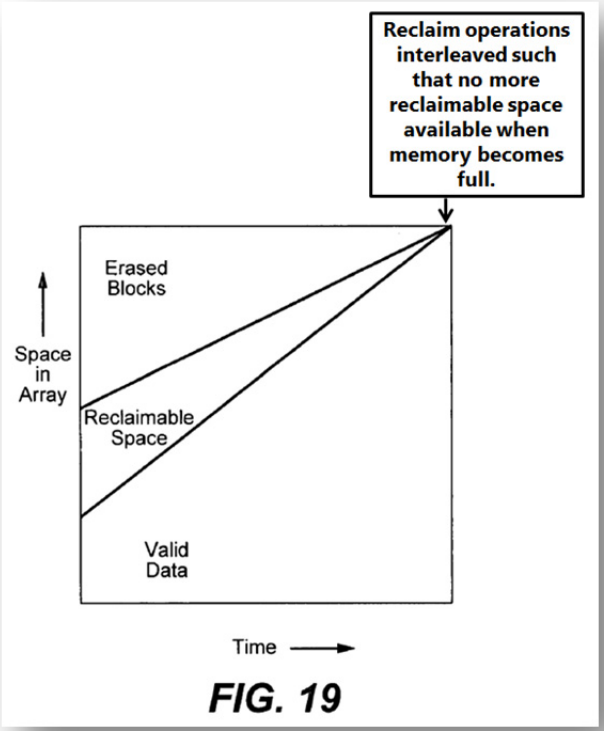


FIG. 19

Id. at Fig. 19 (displaying a time graph illustrating an optimal interleave of reclaim operations to write operations) (annotated).

The interleave ratio can be recalculated periodically or when triggered by a host command. *Id*. at 18:17-26. As shown in Fig. 20, when host issues a delete command to the memory device at time t10, the system alters the interleave ratio such that the time when no more reclaimable space is available shifts to the newly anticipated time when the memory will be filled to capacity. *Id*. at 18:12-26.

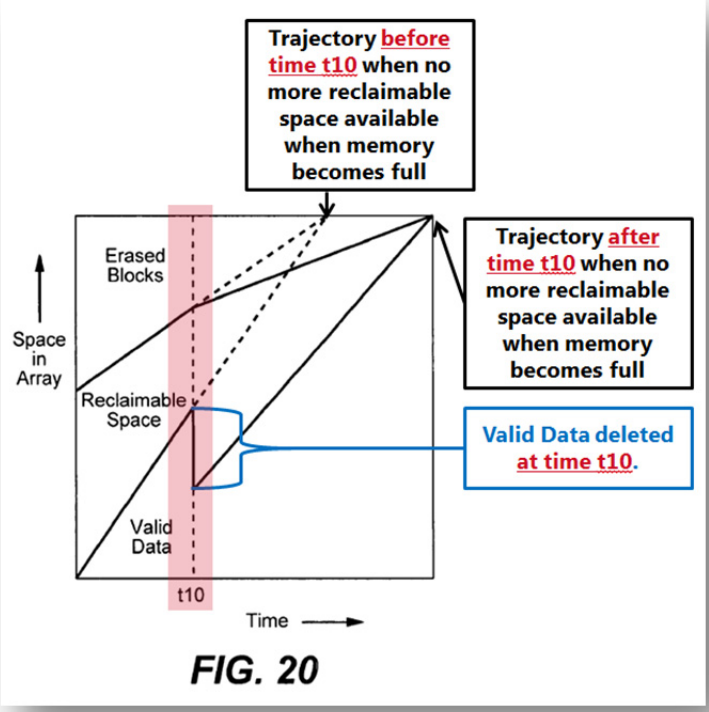


FIG. 20

Id. at Fig. 20 (annotated).

Sinclair also describes other reclaim modes selectable by the host, such that that "the host may have commands to select the appropriate reclaim mode based on

present host activity or expected host activity." *Id*. at 23:27-30, 23:36-59.

VIII. GROUNDS FOR CHALLENGE

Inter partes review of Claims 10 and 12-18 of the '850 Patent is requested as follows.

A. GROUND 1: Claims 10, 12, and 18 are unpatentable under 35 U.S.C. § 102 over *CompactFlash*.

1. Independent Claim 10

(i) A memory device, comprising:

The CompactFlash device is a memory device and contains a "controller and flash memory module(s)." Ex. 1003 at 19.

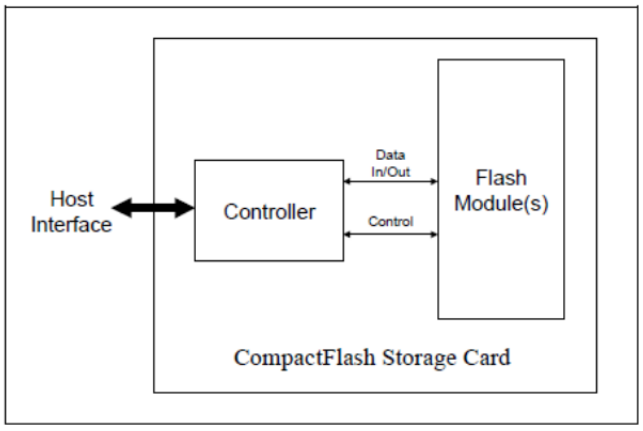


Figure 1: CompactFlash Storage Card Block Diagram

(ii) one or more predefined access profiles to determine how access to the memory device is configured for at least one usage of the memory device;

CompactFlash discloses MultiWord and Ultra DMA modes that determine how access to the memory device is configured for a usage. As shown in Table 53,

various MultiWord DMA and Ultra DMA modes are supported, each of which constitutes a pre-defined access profile.

Table 53: Transfer mode values

Mode	Bits (7:3)	Bits (2:0)
PIO default mode	00000b	000b
PIO default mode, disable IORDY	00000b	001b
PIO flow control transfer mode	00001b	Mode
Reserved	00010b	N/A
Multiword DMA mode	00100b	Mode
Ultra DMA Mode	01000b	Mode
Reserved	10000b	N/A
Mode = transfer mode number		

Ex. 1003 at 158 (equating MultiWord DMA and Ultra DMA "Mode[s]" as "transfer mode number[s]"). For example, the following Ultra DMA Modes are possible:

6.2.1.6.26 Word 88: Ultra DMA Modes Supported and Selected

* * *

Bit 4: 1 = Ultra DMA mode 4 and below are supported. Bits 0-3 Shall be set to 1.

Bit 3: 1 = Ultra DMA mode 3 and below are supported, Bits 0-2 Shall be set to 1.

Bit 2: 1 = Ultra DMA mode 2 and below are supported. Bits 0-1 Shall be set to 1.

Bit 1: 1 = Ultra DMA mode 1 and below are supported. Bit 0 Shall be set to 1.

Bit 0: 1 = Ultra DMA mode 0 is supported

Id. at 137; 132-33.

Each DMA mode is an access profile, because the selected mode is used to determine the memory device configuration for the subsequent DMA access

operations to the memory device. *See* Ex. 1001 at 3:56-58 ("***This profile***, which may be any one of the supported predefined profiles, ***governs the current access operations to the memory device***.") (emphasis supplied); Ex. 1002 at ¶¶210-14. For instance, if an Ultra DMA mode is activated (unlike MultiWord DMA modes) the device configures itself to perform a CRC check calculation after the data transfer. Ex. 1003 at 90 ("CRC errors are detected and reported only while operating in Ultra mode."). In addition, unlike the MultiWord DMA modes, each Ultra DMA mode utilizes both the rising and falling edge of the clock signal to transfer data and utilizes HDMARDY, DDMARDY, and DSTROBE signals, which are not part of a MultiWord DMA access operation. Ex. 1002 at ¶213-14; Ex. 1003 at 44; Fig. 32 (illustrating the MultiWord DMA transfer initialization process).

Furthermore, each Ultra DMA mode is different. As an example, after a pause in an Ultra DMA data burst, a memory device will be prepared to receive up to two additional data words for Ultra DMA Modes 0-2 and up to three additional data word for DMA Modes 3-5. Ex. 1003 at 70. Finally, each mode is associated with different maximum transfer rates. *Id*. at Tables 21-22; Ex. 1002 at ¶213.

These DMA modes are access profiles because use of each DMA Mode requires the use of a specific DMA *protocol*, not merely the selection of particular timing characteristics. Ex. 1003 at 68 ("[T]he *Ultra DMA protocol* shall be used

instead of the *Multiword DMA protocol.…* This protocol applies to the Ultra DMA data burst only."). *See* Ex. 1008 at 266-67. Moreover, in the example of Ultra DMA, "[s]everal signal lines are redefined to provide different functions during an Ultra DMA burst." *Id*. at 68.

Each MultiWord and Ultra DMA mode is effective for determining access configuration for a usage. The "usage" in *CompactFlash* relates to the host activity in accordance with the selected DMA mode (*e.g.*, a READ DMA or WRITE DMA operation). *Id*. Specifically, the controller configures the CompactFlash device for a host-initiated access operation according to the host-selected DMA mode. *Cf.* Ex. 1003 at Fig. 32 *with id*. at Figs. 32, 38 (illustrating the difference in initialization process between a MultiWord DMA transfer and an Ultra DMA transfer). For example, enabling an Ultra DMA mode causes the controller to initiate the Ultra DMA protocol (rather than the MultiWord DMA protocol) when receiving a READ DMA or WRITE DMA command from the host. *Id*. at 68. Similarly, enabling a MultiWord DMA mode causes the controller to initiate the MultiWord DMA protocol (rather than the Ultra DMA protocol) when receiving a READ DMA or WRITE DMA command from the host. *Id*. at 68.

Furthermore, unlike the Patent Owner's argument in the Prosecution History that changing the operating speed does not affect an access operation, the DMA modes ***directly affect*** access operations. *See* Ex. 1008 at 266-67. The selected

DMA mode dictates the specific DMA protocol that the memory device utilizes. Moreover, the selection of an Ultra DMA profile rather than a MultiWord DMA profile requires the device to perform a CRC comparison to ensure the accuracy of the data transferred. Ex. 1003 at 90 ("CRC errors are detected and reported ***only while operating in an Ultra mode***.") (emphasis supplied); Ex. 1002 at ¶213.

(iii) a controller configured to receive at least one first command to activate at least one of the one or more predefined access profiles associated with the memory device; and

CompactFlash has a CompactFlash controller, which receives a SET FEATURES command from the host. *See* Ex. 1003 at Fig. 1 (shown below). The SET FEATURES command is communicated from the host to the CompactFlash controller to select and activate one of the MultiWord or Ultra DMA modes (*i.e*., a predefined access profile) from among the available modes supported by the device. *Id*. at 15, 156-58, Table 53 (indicating bits representing a host-selected DMA mode) (shown below). Specifically, the SET FEATURES command instructs the controller to set the DMA mode to the selected MultiWord DMA mode or Ultra DMA mode for subsequent DMA access operations (*i.e*., the usage), such as READ DMA and WRITE DMA. *Id*. at 15, 157-58.

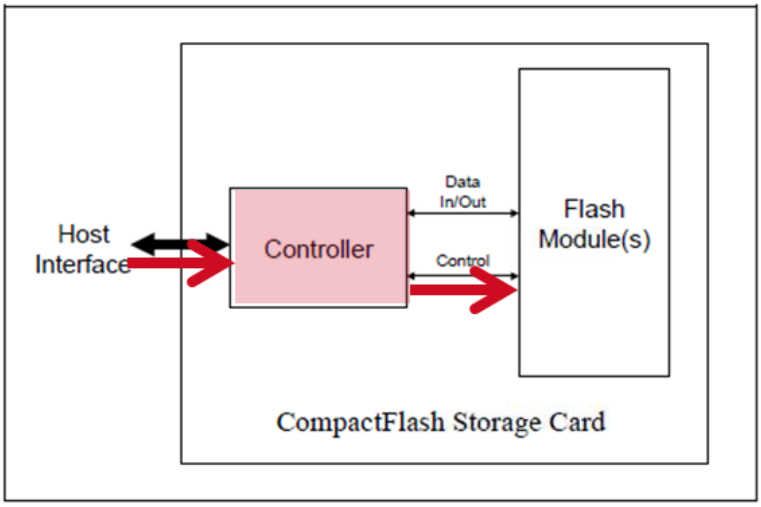


Figure 1: CompactFlash Storage Card Block Diagram

Id. at Fig. 1 (annotated).

Table 53: Transfer mode values

Mode	Bits (7:3)	Bits (2:0)
PIO default mode	00000b	000b
PIO default mode, disable IORDY	00000b	001b
PIO flow control transfer mode	00001b	Mode
Reserved	00010b	N/A
Multiword DMA mode	00100b	Mode
Ultra DMA Mode	01000b	Mode
Reserved	10000b	N/A
Mode = transfer mode number		

Id. at Table 53 (annotated).

For example, for Ultra DMA modes, the Set transfer mode subcommand (using the transfer mode values in Table 53) in the SET FEATURES command allows "a host to select the Ultra DMA mode at which the system operates." *Id*. at 69.

(iv) receive at least one second command to configure access to the memory device in accordance with the at least one of the one or more predefined access profiles such that at least a portion of the memory device is configured according to the at least one of the one or more predefined access profiles for the at least one usage.

CompactFlash teaches the CompactFlash controller receiving a READ DMA or WRITE DMA command (*i.e.*, a second command) to configure access to the memory device for a read or write operation in accordance with the activated MultiWord or Ultra DMA mode (*i.e*., a predefined access profile). Ex. 1003 at 52. As noted above, the configuration of the memory device, including the protocol used to carry out the READ DMA or WRITE DMA command, will depend on which DMA mode was previously set by the host. *Id*. at 68 (noting that when the Ultra DMA protocol is enabled, it is "used instead of the Multiword DMA protocol" when the "READ DMA, and WRITE DMA commands" are "issued by the host"); *see supra* Section VIII.A.1(ii).

After a host communicates a READ DMA or WRITE DMA command following the selection of a DMA mode, the controller will begin the DMA-specific initiation protocol to configure access to a portion of the memory device

(*i.e.*, the readable and writeable area for the memory device) for either a MultiWord DMA transfer or an Ultra DMA transfer in the selected mode. Ex. 1003 at Figs. 32-33, 38. Specifically, the controller configures the CompactFlash device to perform a DMA transfer under the selected DMA mode on specified memory sectors in the CompactFlash memory (*i.e.*, a portion of the memory device). *Id*. at 145 ("[The Read DMA] command uses DMA mode to read from 1 to 256 sectors as specified in the Sector Count register.... The transfer begins at the sector specified in the Sector Number Register"), 164 ("[The Write DMA] command uses DMA mode to write from 1 to 256 sectors as specified in the Sector Count register.... The transfer begins at the sector specified in the Sector Number Register."); Ex. 1002 at ¶¶221-22. Immediately following the configuration of the device, a data transfer will occur between the host and the memory device under the host-activated MultiWord or Ultra DMA mode. Ex. 1003 at 76-77, Figs. 32, 33, 38; Ex. 1002 at ¶223.

In a MultiWord DMA mode, the controller starts the initiation phase of the MultiWord DMA burst by asserting a DMARQ signal (Step 1 of Fig. 32). The host, in response, asserts a DMACK signal (Step 2 of Fig. 32), and, in turn, the controller response by asserting the IORD signal (Step 3 of Fig. 32).

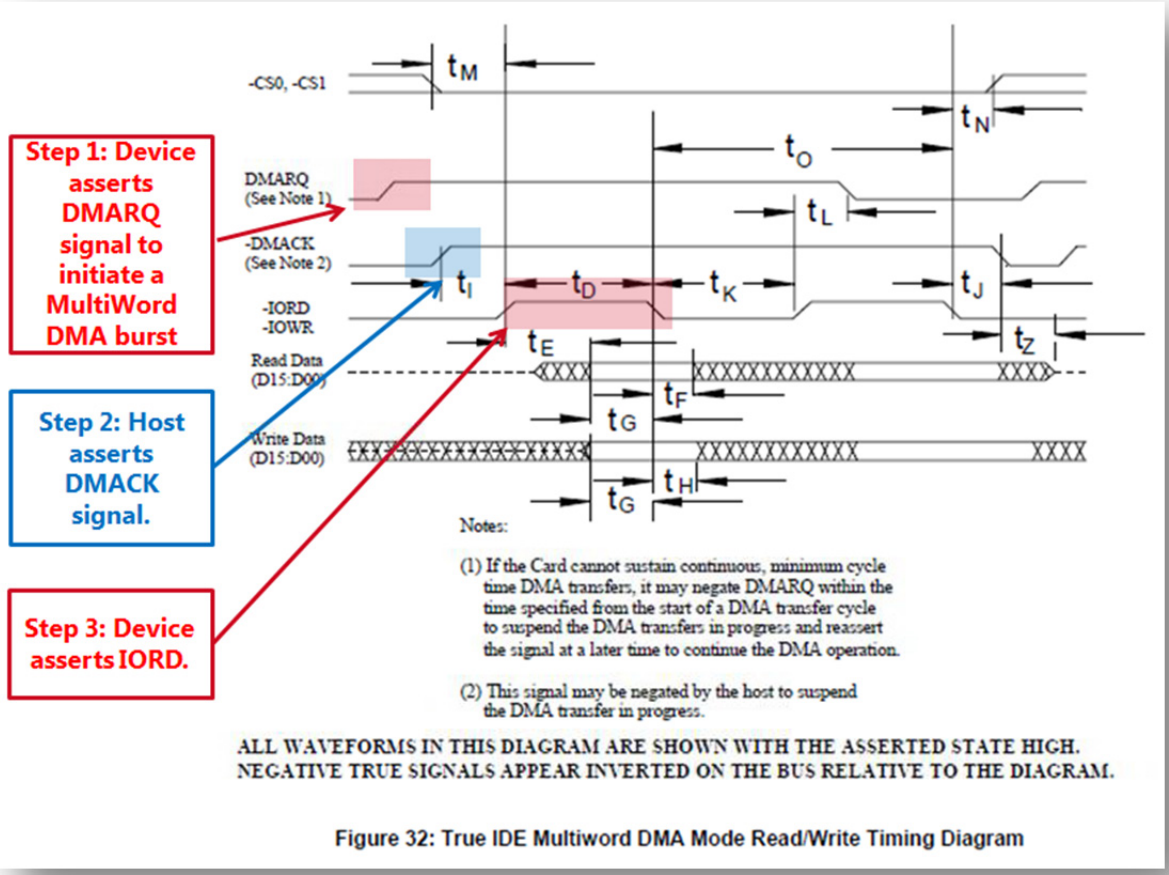
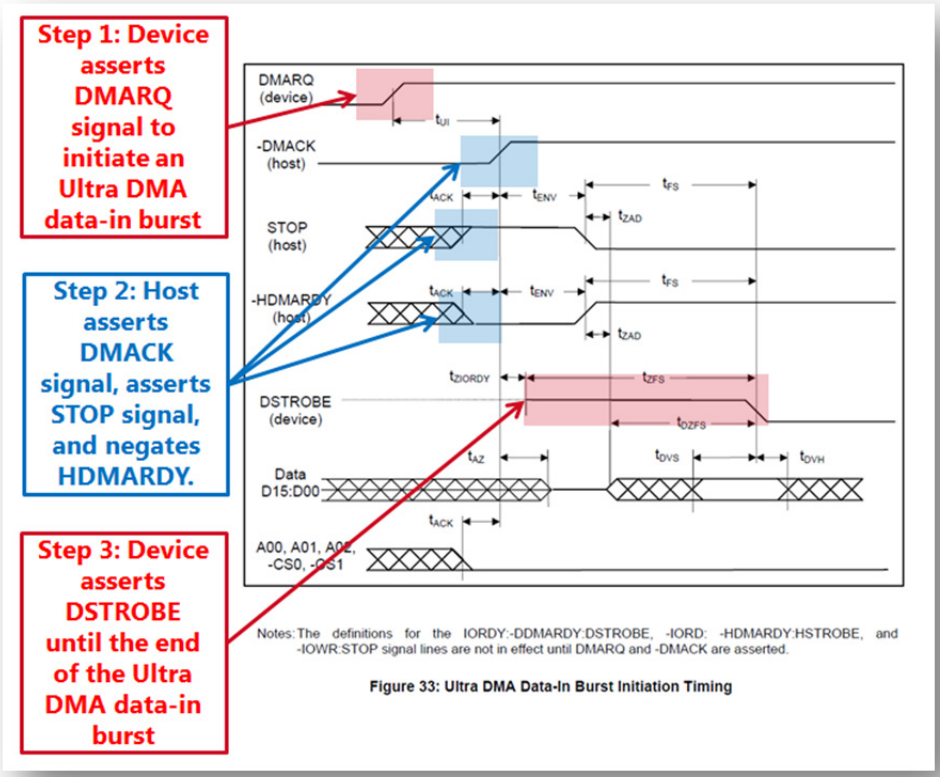


Figure 32: True IDE Multiword DMA Mode Read/Write Timing Diagram

Ex. 1003 at Fig. 32 (annotated).

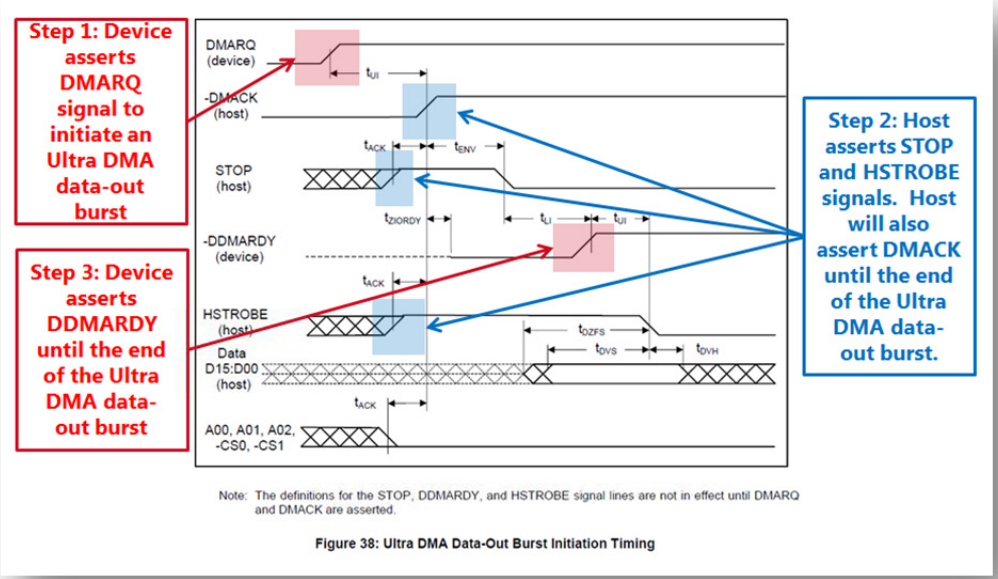
As another example, when an Ultra DMA mode is activated, the controller starts the initiation phase of the Ultra DMA burst by asserting a DMARQ signal (Step 1 of Fig. 33). *Id*. at 76. The host then asserts a DMACK signal (Step 2 of Fig. 33). *Id.* at Fig. 33. Only at that point do the other signal lines become effective, permitting the transfer. *Id.* ("The definitions for the …DSTROBE… and -IOWR:STOP signal lines are not in effect until DMARQ and -DMACK are asserted). The device will assert a DSTROBE signal line to start the data-in burst

(Step 3 of Fig. 33). *Id.* at Fig. 33.



Id. at Fig. 33 (annotated).

For a data-out burst, the device will wait to negate any signal until generating a STROBE edge (*see* Step 3 of Fig. 38). *Id.* at 70. At this point, the memory device is now configured for usage (*i.e.*, to perform a data transfer under the selected Ultra DMA mode).



Id. at Fig. 38 (annotated).

Finally, as opposed to a READ SECTOR and WRITE SECTOR command in *CompactFlash* that causes the CompactFlash device to only perform a data transfer, the READ DMA and WRITE DMA command causes the CompactFlash device to first configure the memory device and then perform a data transfer. After receiving a READ SECTOR or WRITE SECTOR command, the host specifies the memory sectors for the data transfer and the CompactFlash device performs the data transfer on the specified memory sectors. *Id*. at 148; Ex. 1002 at ¶¶224-25. On the other hand, after receiving a READ DMA and WRITE DMA command, the CompactFlash controller must first configure the CompactFlash device for a DMA transfer and only then may the CompactFlash device perform the DMA transfer. Ex. 1003 at 76-77, Figs. 32, 33, 38; Ex. 1002 at ¶224.

2. Dependent Claim 12

(i) The memory device of claim 10, wherein at least one of the one or more predefined access profiles comprises a default access profile.

When executing a power-on or hardware reset, *CompactFlash* discloses that the device may revert to a default non-Ultra DMA mode, like a MultiWord DMA mode. Ex. 1003 at 69. As discussed above, each MultiWord DMA mode may be an access profile. Ex. 1002 at ¶¶210-14; *see supra* VIII.A.1(ii).

3. Dependent Claim 18

(i) The memory device of claim 10, wherein the one or more predefined access profiles comprise a plurality of predefined access profiles.

As discussed above, each DMA mode is a predefined access profile. Consequently, *CompactFlash* discloses a plurality of predefined access profiles as each MultiWord and Ultra DMA Mode is a predefined access profile. Ex. 1003 at 137, 132-33, 158

Table 53: Transfer mode values

Mode	Bits (7:3)	Bits (2:0)
PIO default mode	00000b	000b
PIO default mode, disable IORDY	00000b	001b
PIO flow control transfer mode	00001b	Mode
Reserved	00010b	N/A
Multiword DMA mode	00100b	Mode
Ultra DMA Mode	01000b	Mode
Reserved	10000b	N/A
Mode = transfer mode number		

Id. at Table 53 (annotated).

B. GROUND 2: Claims 14-17 are unpatentable under 35 U.S.C. § 103(a) over *CompactFlash* and *Henkel*.

1. Dependent Claim 14

(i) The memory device of claim 10, wherein the controller is configured to conduct two or more simultaneous memory access operations.

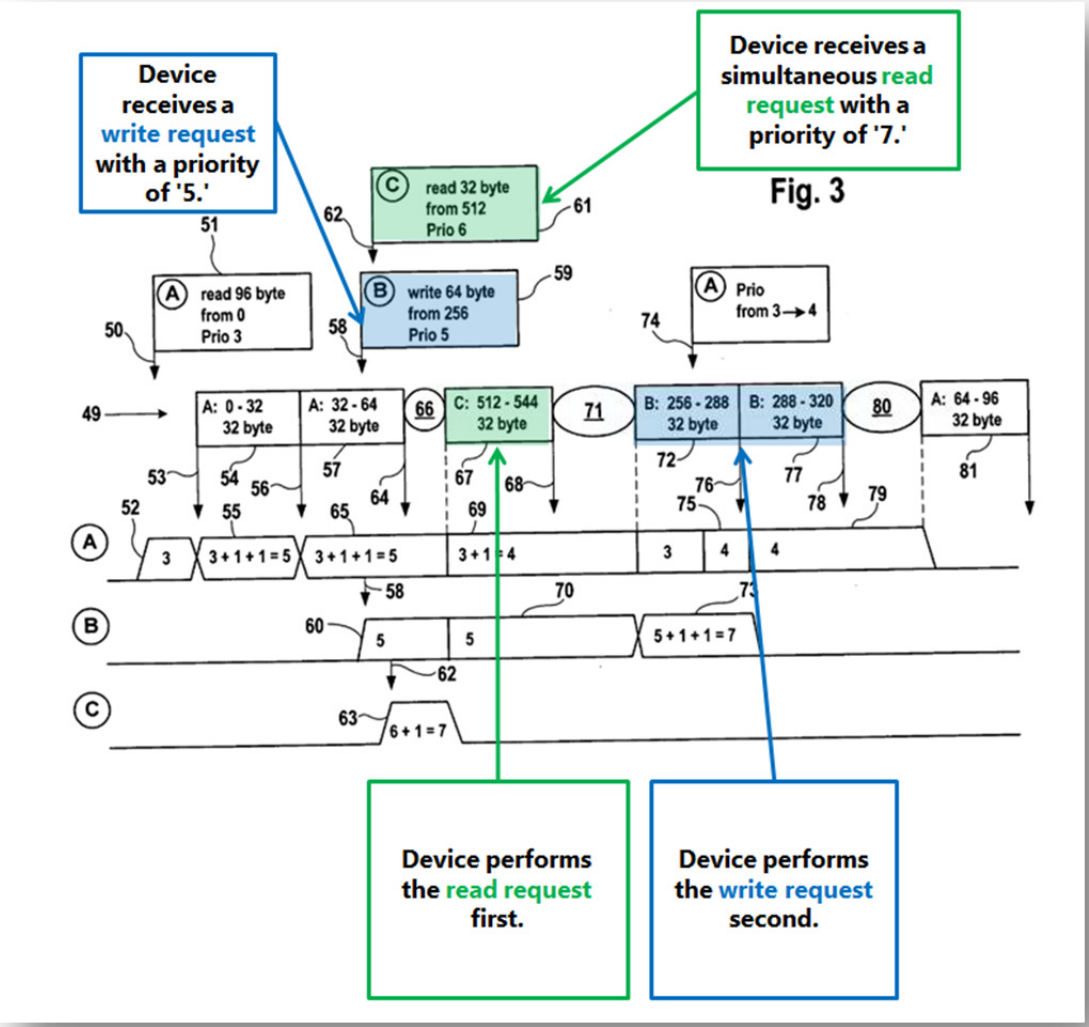
As discussed above, *CompactFlash* discloses the memory device of Claim 10.

Henkel additionally discloses a controller that is configured to conduct two or more simultaneous memory access operations. To conduct the simultaneous memory access operations, *Henkel* discloses an arbitration unit (*i.e.*, a controller) that prioritizes each access operation. *See* Ex. 1006 at ¶5 (disclosing arbitration unit as comprising "control logic"), ¶53; Ex. 1002 at ¶237.

By prioritizing simultaneous access operations, the controller in *Henkel* is able to efficiently conduct simultaneous access operations. *See* Ex. 1006 at ¶¶61-

62; Ex. 1002 at ¶238. The "read and write accesses to the shared memory are scheduled in accordance with priorities that are assigned ... to at least some of the incoming and outgoing data streams." Ex. 1006 at ¶12. For example, "if a high priority read request is received, the corresponding read accesses will be carried out favorably, while the low-priority write and read accesses will be postponed." *Id*. at ¶12.

Henkel even depicts a scenario where the controller conducts simultaneous read and write requests (*i.e.*, memory access operations). In Fig. 3 (shown below), the memory device simultaneously receives and schedules (*i.e*., conducts) a write request and a read request. *Id*. at ¶60-61. Because the read request has the highest priority, the controller performs the read request first. *Id*. at ¶63. The memory device performs the write request immediately afterward. *Id*. at ¶¶64-66.



Id. at Fig. 3 (annotated).

It would have been obvious to include the access operation prioritization scheme as taught by *Henkel* in the *CompactFlash* device for the following reasons.

Explicit Teaching to Combine

CompactFlash and *Henkel* are in a similar field, technology, and time frame. Ex. 1002 at ¶¶229-30. Both *CompactFlash* and *Henkel* expressly teach an exemplary embodiment using a storage device. Ex. 1003 at 15 ("[CompactFlash] ***Storage Cards*** provide[] the capability to easily transfer all types of digital information and software between a large variety of digital systems."); Ex. 1006 at Abstract ("The invention provides an arbitration unit adapted for controlling accesses to a shared ***storage***."); Ex. 1002 at ¶229. Moreover, both references were made available in a similar time frame: *CompactFlash* was publicly available for download by the beginning of 2005 and *Henkel* published at the end of 2004. Ex. 1015; Ex. 1003.

One of ordinary skill in the art would have especially been motivated to combine *CompactFlash* and *Henkel,* because *Henkel* improves the efficiency of data transfers. Ex. 1006 at ¶13 ("As a result [of the prioritization scheme], a low-latency storage access can be provided."). Improving the efficiency of data transfers was a known motivation in the art, and was a primary reason *CompactFlash* introduced Ultra DMA modes. Ex. 1002 at ¶231; *see, e.g.*, Ex. 1018 ("Ultra DMA … modes will increase the CompactFlash interface data transfer rate[.]").

An additional motivating reason to combine *CompactFlash* with *Henkel* is that *Henkel* provides greater control for the host to order its operations. A host may impose requirements on read/write operations communicated to a memory device. Ex. 1006 at ¶¶7, 10 ("Incoming data streams that are received from a certain functional unit ... can be processed in accordance with the requirements imposed by said functional unit."). One of ordinary skill in the art would have been motivated by the teachings of *Henkel* to use host prioritization of data to gain the advantage of meeting "required properties of the ... data streams," while retaining the advantages of *CompactFlash* of performing read/write operations under a well-established standard. Ex. 1002 at ¶232; Ex. 1006 at ¶7.

Known Technique to a Known Device to Yield Predictable Results

Base system: *CompactFlash* discloses a memory device that performs DMA access operations. Ex. 1003 at 145, 164-65; Ex. 1002 at ¶234.

Known technique: As shown by *Henkel*, a controller (*e.g.*, an arbitration unit) may prioritize multiple memory access operations in order to minimize delays. Ex. 1006 at ¶14; Ex. 1002 at ¶234.

Predictable results and improved system: A POSITA would have recognized that implementing *Henkel's* memory access prioritization scheme in *CompactFlash*'s memory device would yield the predictable result of a memory

device that employed a memory access prioritization scheme to lessen access operation delays. Ex. 1002 at ¶234.

Moreover, it would have been obvious to use the memory access prioritization scheme taught by *Henkel* as the scheme is a desirable and efficient way of prioritizing and processing multiple memory access operatoins. Ex. 1002 at ¶235.

2. Dependent Claim 15

(i) The memory device of claim 14, wherein the controller is configured to assign access priority levels to resolve conflicting simultaneous memory access operations.

As discussed above, the *CompactFlash-Henkel* combination discloses the memory device of Claim 14. *Henkel* further discloses the controller assigning access priority levels to resolve conflicting simultaneous memory access operations.

To determine the access operation to perform in light of conflicting simultaneous memory access operations, *Henkel* teaches assigning a priority to each access operation to determine which access operation to perform first. Ex. 1006 at ¶41 ("The arbitration unit 30 schedules a sequence of write and read data slices, whereby each of said data slices is of fixed size.... Said data slices ***are***

scheduled in accordance with their respective priority.") (emphasis supplied). The higher-priority access operations are processed more quickly than lower-priority access operations. *Id*. at ¶12.

3. Dependent Claim 16

(i) The memory device of claim 10, wherein the controller is configured to interleave two or more simultaneous memory access operations.

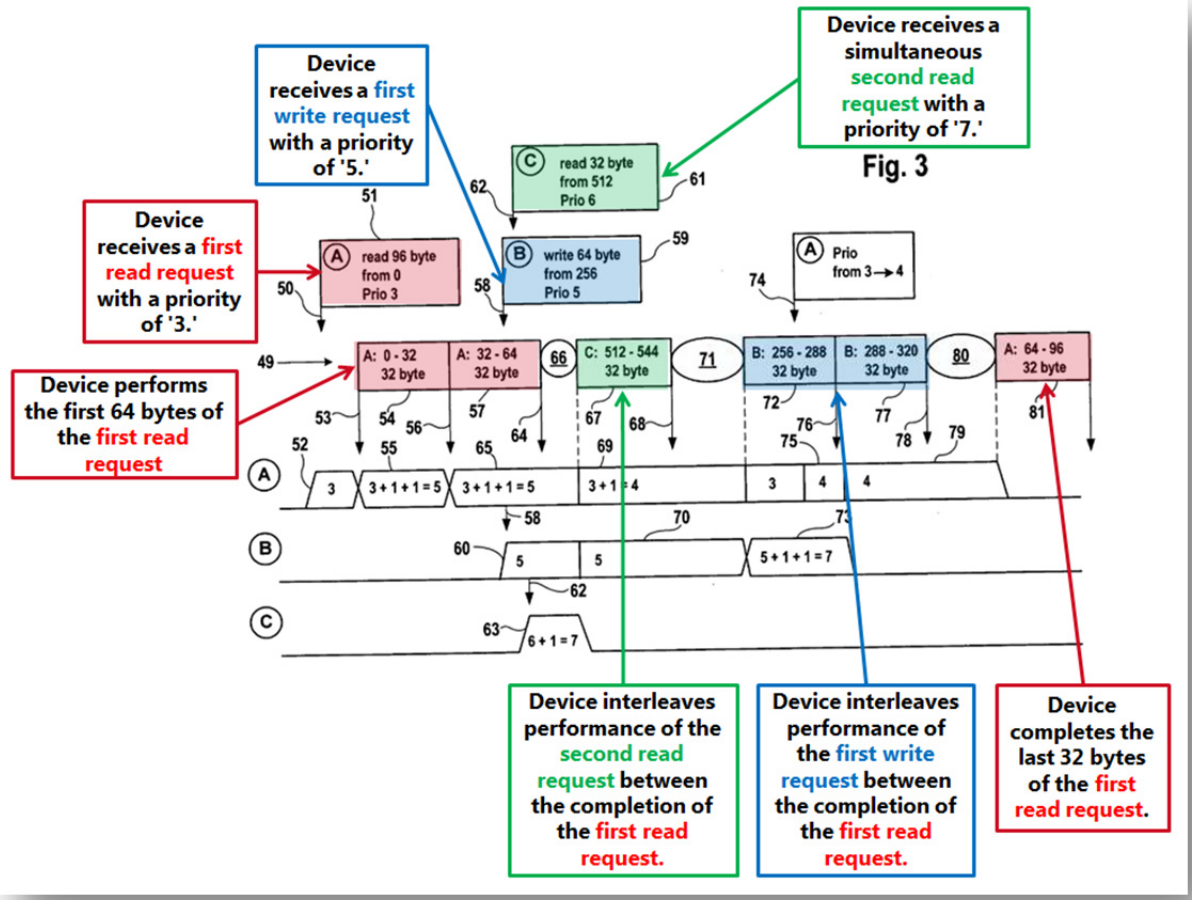
As discussed above, *CompactFlash* discloses the memory device of Claim 10. In addition, as discussed above, it would have been obvious to a POSITA to include the access operation prioritization scheme as taught by *Henkel* in the *CompactFlash* device. *See* Ex. 1002 at ¶¶228-35.

Henkel additionally discloses that the controller is configured to interleave two or more simultaneous memory access operations. To conduct two or more simultaneous memory access operations, *Henkel* discloses an arbitration unit (*i.e.*, a controller) that prioritizes each access operation. *See* Ex. 1006 at ¶5 (disclosing arbitration unit as comprising "control logic"), ¶53; Ex. 1002 at ¶¶243-44.

By prioritizing simultaneous access operations, the controller in *Henkel* is able to efficiently conduct simultaneous access operations by interleaving the access operations. *See* Ex. 1006 at ¶61-63 (detailing an example scenario where the memory device interleaves a write request between the performance of a read

request because the write request has a higher priority). The controller in *Henkel* performs higher-priority access operations before completion of lower-priority access operations. *Id*. at ¶12 ("[I]f a high priority read request is received, the corresponding read accesses will be carried out favorably, while the low-priority write and read accesses will be postponed.").

Henkel even depicts a scenario where the controller interleaves simultaneous read and write requests (*i.e.*, memory access operations). In Fig. 3 (shown below), the memory device simultaneously receives a write request and a second read request after partially processing a first read request. *Id*. at ¶¶60-61. The memory device interleaves the write request and second read request between the performance of the first read request, because the write request and second read request had a higher priority. *Id*. at ¶¶62-66. After completion of the write request and second read request, the memory device completes the first read request. *Id*. at ¶67.



Id. at Fig. 3 (annotated).

4. Dependent Claim 17

(i) The memory device of claim 16, wherein the controller is configured to assign access priority levels to resolve conflicting interleaved memory access operations.

As discussed above, the *CompactFlash-Henkel* combination discloses the memory device of Claim 16. *Henkel* further discloses the controller assigning

access priority levels to resolve conflicting interleaved memory access operations.

To determine the access operation to perform in light of conflicting interleaved memory access operations, *Henkel* teaches assigning a priority to each memory access operation to determine which access operation to perform first. Ex. 1006 at ¶41 ("The arbitration unit 30 schedules a sequence of write and read data slices, whereby each of said data slices is of fixed size.... Said data slices ***are scheduled in accordance with their respective priority***.") (emphasis supplied). The higher priority access operations are processed more quickly than lower priority access operations. *Id*. at ¶12.

C. GROUND 3: Claims 10, 12, and 18 are unpatentable under 35 U.S.C. § 103(a) over *Ziv* and *Vogt*.

1. Independent Claim 10

(i) A memory device, comprising:

Ziv discloses a memory device, such as a portable storage device that includes a storage memory. Ex. 1004 at 3:48-55; Fig. 2.

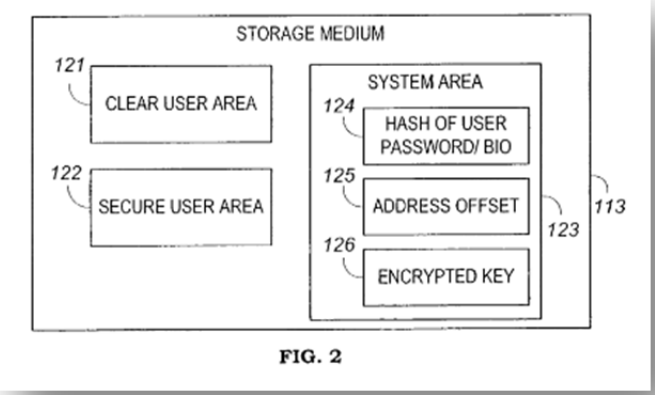


FIG. 2

Id. at Fig. 2.

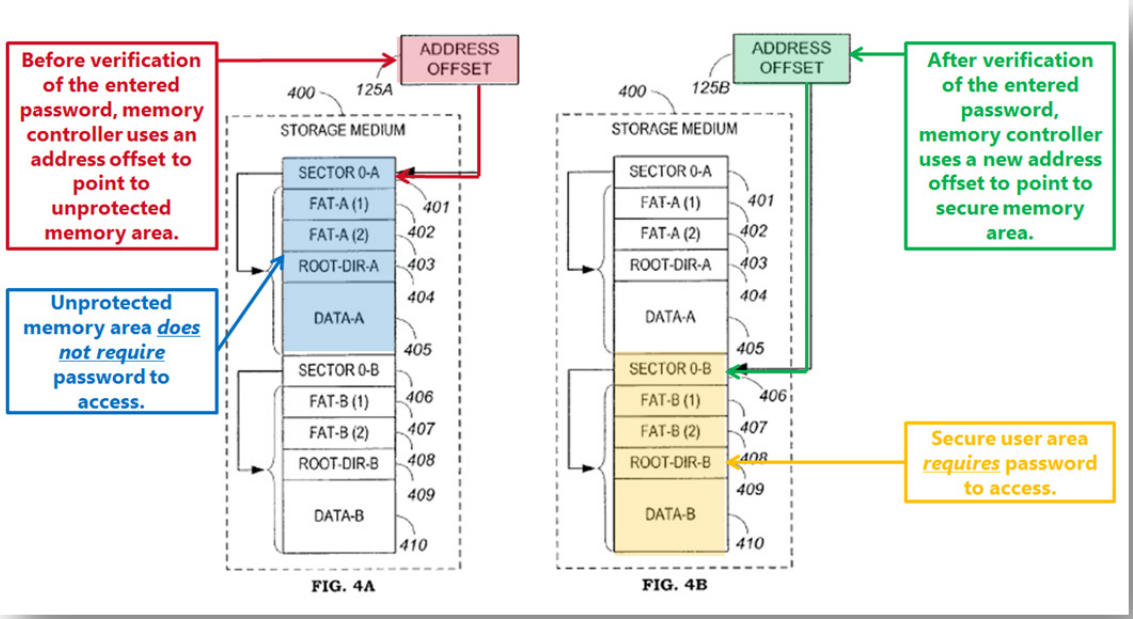
(ii) one or more predefined access profiles to determine how access to the memory device is configured for at least one usage of the memory device;

Ziv discloses a predefined access profile that contains a password hash, an address offset, and a key, which are predefined. Ex. 1004 at 5:1-25 (describing initial setup); Ex. 1002 at ¶¶259-60. The password hash, address offset, and key in *Ziv* constitutes a predefined access profile that determines how access to the memory device is configured for the host's access to the secure memory area (*i.e*., the usage). *See* Ex. 1004 at 1:60-63.

The password hash, address offset, and key in *Ziv* constitutes a predefined access profile that governs the access operations to the memory. *See* Ex. 1001 at 3:56-58 ("***This profile***, which may be any one of the supported predefined profiles, ***governs the current access operations to the memory device***.") (emphasis

supplied). Additionally, each of these components individually forms an access profile. Ex. 1002 at ¶¶280.

Access to the secure user data is only available when a hash of an entered password matches the password hash stored in the register. Ex. 1004 at 4:11-12 ("[A] secure area 122 that contains secure user data [is] accessible only upon the provision of a password[.]"). In addition, the address offset governs access operations to the memory. After entering the proper password, the memory device uses the stored address offset to properly view the secure memory area. *Id*. at 6:46-48 ("[H]ost 101 will seek 'sector 0' of the remounted device, controller 111 will use offset 125B to point at 'sector 0-B' 406[.]"). Without the address offset, the secure area will not be properly mounted. *Id.*; Ex. 1002 at ¶261.



Ex. 1004 at Fig. 4A-4B (illustrating the use of the address offset to access the secure area) (annotated).

Moreover, the key governs the access operations to the memory. The key is the "permanent encryption key for all data stored in the secure memory." *Id*. at 7:7-9. Without the key, a user cannot read or write to the secure area. *Id*. at 7:7-9; 7:36-46 (instructing the controller to decrypt data being read from and encrypt data being written to the secure memory area using the key); Fig. 10 (shown below); Ex. 1002 at ¶262.

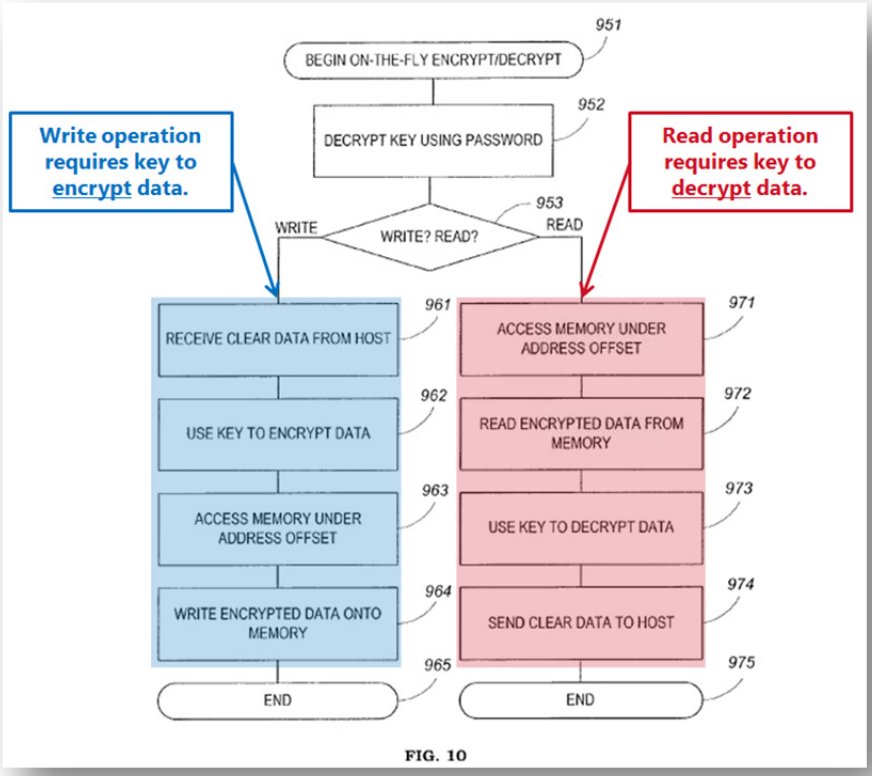


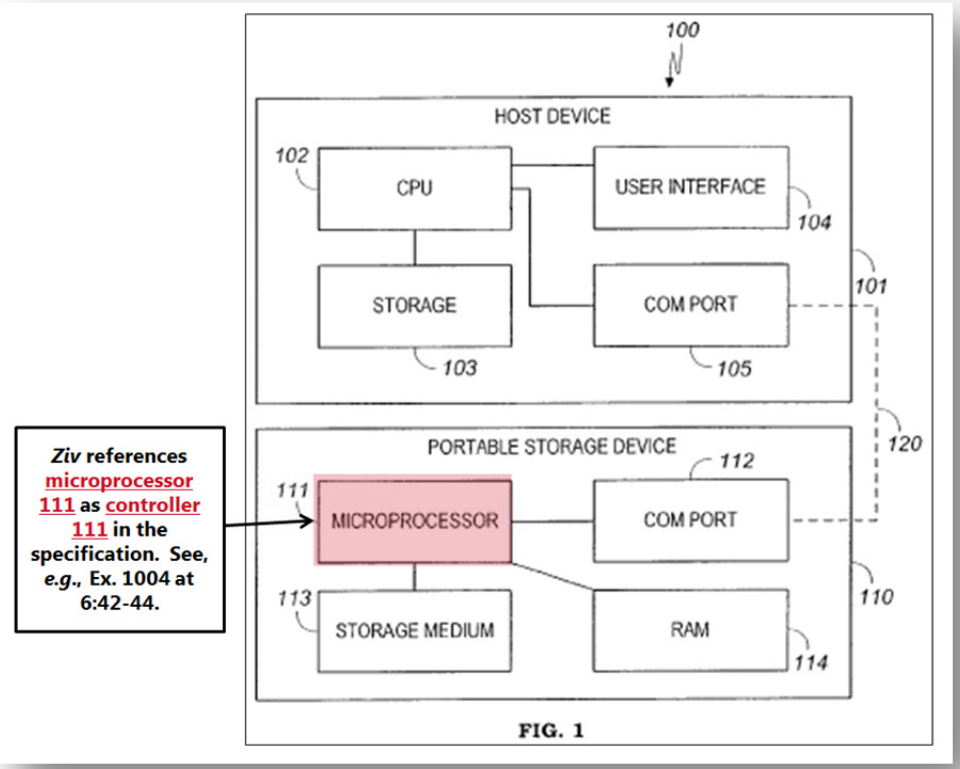
FIG. 10

Ex. 1004 at Fig. 10 (annotated).

(iii) a controller configured to receive at least one first command to activate at least one of the one or more predefined access profiles associated with the memory device; and

Ziv discloses a controller (microprocessor) in the memory device that receives a command indicating the password entered to gain access the secure area. Ex. 1004 at 6:42-44 ("***[C]ontroller 111*** dismounts and remounts portable storage device 110[.]") (emphasis supplied). A skilled artisan would understand the microprocessor in *Ziv* to be a controller. Ex. 1002 at ¶267. *Ziv* further discloses

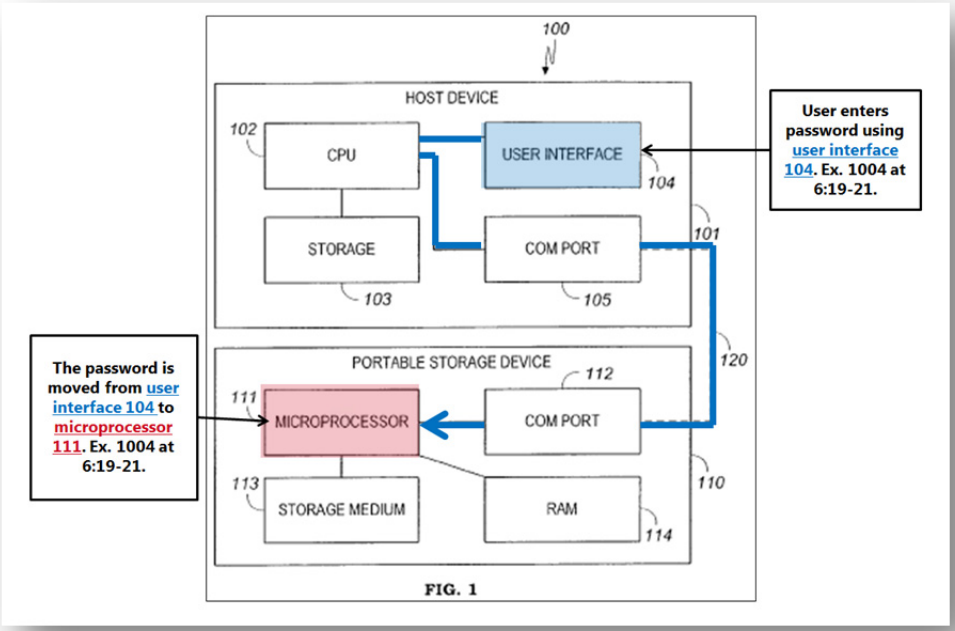
that a password entry from the host triggers several activation procedures associated with the predefined access profile.



Ex. 1004 at Fig. 1 (annotated).

The controller receives a command indicating the password entered. *Id*. at 6:19-30 ("If the password has been entered ... via user interface 104, then in step 702 this password is moved to microprocessor 111.... [T]he hashed entered password is then compared to the hashed stored password in register 124."), Fig. 1 (below). In other words, the host system instructs (*i.e.,* sends a command to) the

controller (microprocessor) to use the entered password to gain access to the secure memory area. Ex. 1002 at ¶268.



Ex. 1004 at Fig. 1 (annotated).

The entered password in *Ziv* activates a comparison of a hash of the entered password with the stored password hash. Ex. 1004 at 6:28-30; Ex. 1002 at ¶271.

Additionally, the password in *Ziv* activates the decryption of the stored key. Ex. 1002 at ¶272. When the entered password matches the stored password, the microprocessor decrypts the key of the predefined access profile. Ex. 1004 at 7:32-35.

The communication of a proper password also activates the remounting of the memory according to the address offset. Ex. 1002 at ¶273. After remounting,

the stored address offset is used to point to the secure memory area. Ex. 1004 at 6:44-46 ("[W]hen remounting device 110, controller 111 will use an address offset[.]").

If the Board disagrees that *Ziv* teaches a "command," it would have been obvious nevertheless to use the command-based password authentication method in *Vogt* in the memory device of *Ziv*. In particular, *Vogt* discloses a "password verify" command that communicates the entered password to the flash device, like a MMC card. Ex. 1005 at 6:36-38 ("The OS sends a 'password verify' command to the flash. ***The command*** ... ***includes the entered password***.") (emphasis supplied). The "password verify" command instructs the internal processor to "compare[] the entered password with the stored password value." *Id*. at 6:30-32.

Moreover, the "password verify" command in the *Ziv*-*Vogt* combination activates the predefined access profile for substantially similar reasons as discussed above. *Id*. at 6:30-32; Ex. 1002 at ¶270.

It would have been obvious to combine the teachings of *Ziv* and *Vogt* to form a memory controller that receives the "password verify" command in *Vogt* for the following reasons.

Simple Substitution of One Known Element for Another

Ziv's memory device differs from the claimed device at most by the command element. Ex. 1002 at ¶248. Commands were well-known in the art to be a fundamental approach to communicating information, such as a password, between devices. *Id*.; Ex. 1005 at 6:36-38. One of ordinary skill could have substituted one known element (a controller receiving a password) for the other (a controller receiving a command containing the password) and the results would have been predictable (a controller receiving a command containing the password). Ex. 1002 at ¶¶250-51.

Moreover, those predictable results would have included the known advantages of *Ziv*'s memory device, which provides access to a secure memory area, and the known advantages of a command, which include reliably and efficiently communicating a password from the host system to memory device. Ex. 1004 at 6:19-23; Ex. 1005 at 3:23-24; Ex. 1002 at ¶¶250-51.

Analogous Art

One of skill in the art would have been motivated to look to the teachings of the prior art due to the similar field, technology, and time frame of *Vogt* and *Ziv*. Ex. 1002 at ¶252. Both patents concern securing data using non-volatile memory in the early 2000s. Ex. 1004 at 3:48-55; Ex. 1005 at 1:13-15; Ex. 1002 at ¶252. In addition, both patents describe a solution to a similar problem–improving security

of portable flash drives using a secure memory and password. Ex. 1004 at 1:60-63 ("[A] portable storage device for securing data stored in the device in a way that will be both convenient and secure."); Ex. 1005 at 2:5-12 ("Techniques for implementing hidden storage in a non-volatile memory storage," where "[t]he hidden storage area cannot be accessed without a valid password.").

Known Technique to a Known Device to Yield Predictable Results

Base system: *Ziv* discloses a memory device, such as a SecureDigital or CompactFlash device, that receives an entered password from the host system. Ex. 1004 at 6:19-30; Ex. 1002 at ¶254.

Known technique: As shown by *Vogt*, receiving a command containing a password was well known. Ex. 1005 at 6:36-38.

Predictable results and improved system: A POSITA would have recognized that implementing *Vogt*'s structured communication protocol of the "password verify command" in *Ziv*'s memory device would yield the predictable result of a memory device that receives an entered password through a structured command, like the password verify command. Ex. 1002 at ¶256.

As such, using *Vogt*'s password-command implementation of communicating a password was an obvious design choice to implement *Ziv*'s teaching of communicating a password. *Id*. at ¶257.

(iv) receive at least one second command to configure access to the memory device in accordance with the at least one of the one or more predefined access profiles such that at least a portion of the memory device is configured according to the at least one of the one or more predefined access profiles for the at least one usage.

The controller in *Ziv* may receive a second command (in the form of a read or write access command) to configure access to the secure memory area in accordance with the password hash, address offset, and key (*i.e.*, predefined access profile). Ex. 1004 at 2:10-12 ("[T]he host device selectably sending data to be written onto the portable storage device and receiving data read from the portable storage device"); 7:35-36 ("In step 953, it is decided whether a read or a write process is required."); Ex. 1002 at ¶276. *See* Ex. 1001 at 5:47-50 (indicating that a second command may indicate an upcoming read or write operation). Using a read or write command, the host ***instructs*** the memory device to read or write to the secure memory area. Ex. 1004 at 2:4-6 ("[A] microprocessor [is] operable to use the clear key to decrypt data read from the secure user area and encrypt data written onto the secure user area."). As shown in Figure 10, the device then enters the appropriate state to process the read or write command (*i.e.*, performing steps

961-965 in the case of a write command, or steps 971-975 in case of a read command). The memory device is then effective for the host to read from or write to the secure memory area (*i.e.*, the usage).

Upon receiving the second command, the memory device is configured to properly read to or write from the secure area. After receiving a read or write access, the microprocessor must point to the proper memory sector in order to access the correct memory to perform the access operation. Ex. 1004 at Fig. 10 (shown below); Ex. 1002 at ¶277.

The microprocessor must also configure the read/write access operations to perform on-the-fly encryption/decryption. When accessing the secure memory area, the microprocessor must apply the encryption key to properly read from and/or write to the secure memory area. Ex. 1004 at 7:35-46, Fig. 10 (shown below); Ex. 1002 at ¶278. Erase operations, however, will not require on-the-fly encryption/decryption. Ex. 1002 at ¶278.

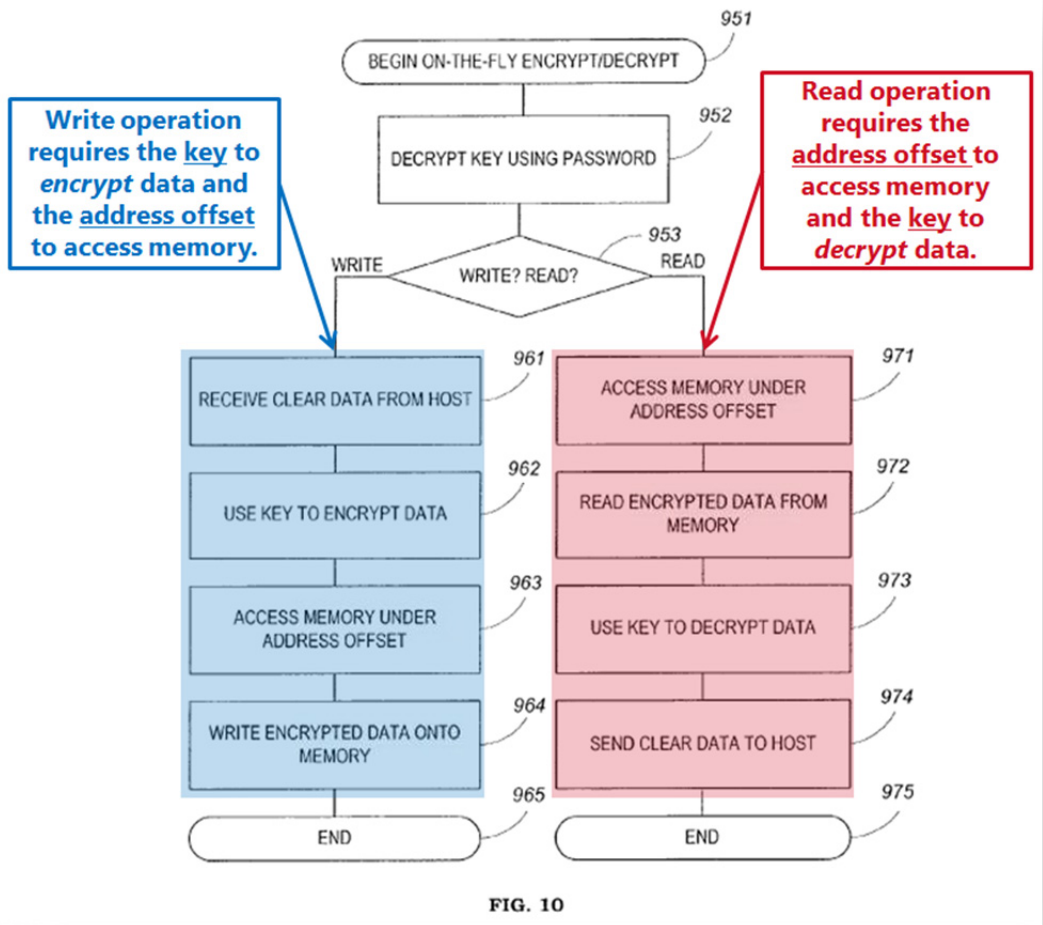


FIG. 10

Ex. 1004 at Fig. 10 (annotated).

After encrypting/decrypting the data and pointing to the proper memory address, a portion of the memory device (*i.e.*, secure area) is now configured according to the usage (*i.e.*, the host reading from or writing to the secure memory area).

Alternatively, *Vogt* discloses a configuration that occurs in accordance with a predefined access profile (*i.e.*, password) after receiving a read/write command (*i.e.*, a second command). Upon receiving a memory read or write signal that

attempts to access a hidden storage area address, the controller configures the memory device to create a hidden storage read or write signal. Ex. 1005 at 3:45-4:6; 4:7-10; Ex. 1002 at ¶¶279-81. To create the hidden storage read/write signal, a Valid_HS_Access signal that accounts for a previously-entered, valid user password for the password-protected hidden storage area is logically "ANDed" with the received memory read/write signal. Ex. 1005 at 3:45-4:10. Moreover, in a read operation accessing a hidden storage, the controller configures the memory device to uniquely transfer data to a "hidden storage bus out" before transferring the data to an external data bus. *Id*. at 5:24-28.

Accordingly, the controller in *Vogt* configures a portion of memory accessed by the hidden read/write signal to properly perform the access operation to the hidden storage (*i.e.*, the usage) in accordance with the host-entered password (*i.e.*, predefined access profile).

In addition to the reasons to combine Ziv and Vogt described *supra* in Section VIII.C.1(iii), one of ordinary skill in the art would have readily combined the read/write process for a hidden storage in *Vogt* with the read/write process in *Ziv* and seen benefits in doing so. Ex. 1002 at ¶252. *Vogt* discloses the "benefit from inclusion of security primitives in flash memory", confirming the desirability of combining *Vogt* with systems, such as *Ziv*, that involve secure access to data in a memory device. Ex. 1005 at 2:32-35; Ex. 1002 at ¶253. Moreover, a POSITA

would look to implement *Vogt*'s known technique of allowing authorized access to password-protected areas to similarly improve the known memory device in *Ziv* that performs access operations to a password-protected secure area. Ex. 1004 at 2:14-16; Ex. 1005 at 2:25-28; Ex. 1002 at ¶¶256-57.

2. Dependent Claim 12

(i) The memory device of claim 10, wherein at least one of the one or more predefined access profiles comprises a default access profile.

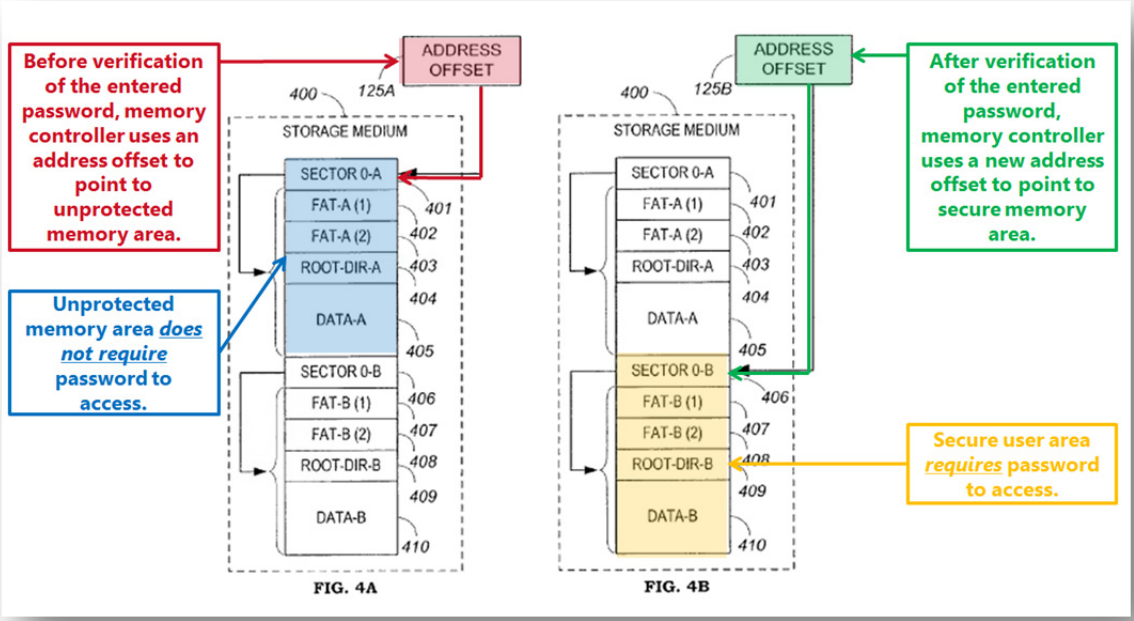
Ziv comprises a default access profile (*i.e.,* access to the clear user area) that is used to set up the memory device upon power up. Ex. 1004 at 6:2-4 ("By default, microprocessor 111 uses an address offset of zero, thus the host sees clear user area 121[.]"). The access to the clear user area does not require entry of a password. *Id*. at 6:4-7.

3. Dependent Claim 18

(i) The memory device of claim 10, wherein the one or more predefined access profiles comprise a plurality of predefined access profiles.

Ziv discloses a plurality of predefined access profiles: (1) a default access profile (*i.e*., access to the clear user area) that is used to set up the memory device upon power up; and (2) a secure access profile (*i.e.*, password, address offset, and

key) that accesses the secure area (as discussed above in Section VIII.C.1.ii). *See* Ex. 1004 at 6:2-4 ("By default, microprocessor 111 uses an address offset of zero, thus the host sees clear user area 121 via 'sector 0-A.'"); Figs. 4A-4B (shown below); *supra* Section VIII.C.1.ii.



Ex. 1004 at Figs. 4A-B (annotated). *See ResQNet. com, Inc. v. Lansa, Inc.*, 346 F.3d 1374, 1382 (Fed. Cir. 2003)("'plurality' suggests the use of 'at least two';" "the term means, simply, 'the state of being plural.'").

D. GROUND 4: Claim 13 is unpatentable under 35 U.S.C. § 103(a) over *Ziv*, *Vogt,* and eMMC.

1. Dependent Claim 13

(i) The memory device of claim 10, wherein the memory device comprises an embedded MultiMedia Card (eMMC) device.

Ground 3 applies *Ziv* and *Vogt* to Claim 10. Embedded MultiMedia Card devices were disclosed in eMMC. Ex. 1022; Ex. 1002 at ¶297.

Additionally, *Vogt* discloses embedding the flash memory within a device (*e.g.*, cell phone). *See* Ex. 1005 at 2:32-42. The "flash memory is ***embedded*** in the device, and cannot be easily reset or replaced." *Id*. at 11:16-19 (emphasis added).

One of ordinary skill in the art would combine the flash memory device in *Vogt* and/or eMMC with the storage medium described in *Ziv* and would see benefits in doing so. Ex. 1002 at ¶294. The flash memory device in eMMC and *Vogt* is a desirable type of storage medium that a skilled artisan would implement in the memory device in *Ziv* as an obvious design choice. Ex. 1002 at ¶295. For instance, eMMC specifically suggests that eMMC was a "versatile" technology and that embedding such a device in a host would give the host "access to all major classes of mass storage memory subsystems," making the "system architecture more flexible than that based upon other memory card-only standards." Ex. 1022. eMMC also notes that employing an eMMC device would make "it easy to embed"

storage on a host system and would keep "technology complexity ... invisible to the host." Ex. 1022.

Similarly, *Vogt* discloses the "benefit from inclusion of security primitives in flash memory", further confirming the desirability of combining *Vogt* with systems, such as *Ziv*, that involve secure access to data in a memory device. Ex. 1005 at 2:32-35.

A Person of Skill in the Art Would Immediately Understand that an MMC Card is a Type of Flash Memory Suggested by Vogt

Because the flash memory in *Vogt* suggests using a MMC card as the flash device, one of ordinary skill in the art would further understand that *Vogt*'s disclosure of "flash memory [that] is embedded in the device" is an embedded MMC (eMMC). Ex. 1002 at ¶299; Ex. 1005 at 11:16-19; Ex. 1022.

E. GROUND 5: Claim 14-17 are unpatentable under 35 U.S.C. § 103(a) over *Ziv*, *Vogt*, and *Henkel*.

1. Dependent Claim 14

(i) The memory device of claim 10, wherein the controller is configured to conduct two or more simultaneous memory access operations.

As discussed above, the *Ziv*-*Vogt* combination discloses the memory device of Claim 10.

As discussed above in Section VIII.B.1, *Henkel* additionally discloses the

controller is configured to conduct two or more simultaneous memory access operations. Ex. 1002 at ¶¶237-39; *see supra* Section VIII.B.1.

It would have been obvious to include the access operation prioritization scheme as taught by *Henkel* in the *Ziv-Vogt* memory device based on the following reasons.

Explicit Teaching to Combine

Ziv, Vogt, and *Henkel* are in a similar field, technology, and time frame. Ex. 1002 at ¶¶301-02. *Ziv*, *Vogt*, and *Henkel* expressly teach an exemplary embodiment using a storage device. Ex. 1004 at 1:7 ("The present invention relates to ***portable storage devices***.") (emphasis added); Ex. 1005 at 1:7-10 ("The present invention relates to ... to a method and device for ***providing hidden storage*** in non-volatile memory.") (emphasis added); Ex. 1006 at Abstract ("The invention provides an arbitration unit adapted for controlling accesses to a shared ***storage***."); Ex. 1002 at ¶301. Moreover, the *Ziv*, *Vogt*, and *Henkel* patents were all filed within a four-year time frame. Ex. 1004; Ex. 1005; Ex. 1006.

One of ordinary skill in the art would have especially been motivated to combine *Ziv* and *Vogt* with *Henkel,* because *Henkel* improves the efficiency of data transfers. Ex. 1006 at ¶13 ("As a result [of the prioritization scheme], a low-latency storage access can be provided."). Ex. 1002 at ¶303. The prioritization scheme discussed in *Henkel* would improve read and write data transfers that occur in the

Ziv-*Vogt* memory device . Ex. 1004 at 2:4-6 ("[A] microprocessor operable to use the clear key to decrypt data read from the secure user area and encrypt data written onto the secure user area."); Ex. 1005 at 8:39-44 ("If the valid password for user(n) bit is asserted, and the decoded address for a read/write instruction is decoded to be part of user(n)'s hidden storage, then the decoded address lines are passed through to the hidden storage memory array and read/write access is enabled.").

An additional motivating reason to combine *Ziv* and *Vogt* with *Henkel* is that *Henkel* provides greater control for the host to order its operations. A host may impose requirements on read/write operations communicated to a memory device. Ex. 1006 at ¶¶7, 10 ("Incoming data streams that are received from a certain functional unit ... can be processed in accordance with the requirements imposed by said functional unit."). One of ordinary skill in the art would have been motivated by the teachings of *Henkel* to use host prioritization of data to gain the advantage of meeting "required properties of the ... data streams," while retaining the advantages of *Ziv* and *Vogt* of performing read/write operations to a secure partition of memory. Ex. 1002 at ¶304; Ex. 1006 at ¶7.

Known Technique to a Known Device to Yield Predictable Results

Base system: The *Ziv-Vogt* combination discloses a memory device that performs read and write access operations. Ex. 1004 at 1:67-2:2; Ex. 1005 at 8:39-44; Ex. 1002 at ¶306.

Known technique: As shown by *Henkel*, a controller (*e.g.*, an arbitration unit) may prioritize multiple memory access operations in order to minimize delays. Ex. 1006 at ¶14; Ex. 1002 at ¶306.

Predictable results and improved system: A POSITA would have recognized that implementing *Henkel's* memory access prioritization scheme in *Ziv-Vogt*'s memory device would yield the predictable result of a memory device that employed a memory access prioritization scheme to lessen access operation delays. Ex. 1002 at ¶306.

Moreover, it would have been obvious to use the memory access prioritization scheme taught by *Henkel* as the scheme is a desirable and efficient way of prioritizing and processing multiple memory access operations. Ex. 1002 at ¶307.

2. Dependent Claim 15

(i) The memory device of claim 14, wherein the controller is configured to assign access priority levels to resolve conflicting simultaneous memory access operations.

As discussed above in Section VIII.B.2, *Henkel* further discloses the controller assigning access priority levels to resolve conflicting simultaneous memory access operations. Ex. 1002 at ¶241; *see supra* Section VIII.B.2.

3. Dependent Claim 16

(i) The memory device of claim 10, wherein the controller is configured to interleave two or more simultaneous memory access operations.

As discussed above, the *Ziv-Vogt* combination discloses the memory device of Claim 10.

In addition, as discussed above in Section VIII.E.1, a POSITA would have readily combined the access operation prioritization scheme as taught by *Henkel* in the *Ziv-Vogt* memory device. Ex. 1002 at ¶¶229-35; *see supra* Section VIII.E.1. Furthermore, as discussed above in Section VIII.B.3, *Henkel* additionally discloses that the controller is configured to interleave two or more simultaneous memory access operations. Ex. 1002 at ¶243-44; *see supra* Section VIII.B.3.

4. Dependent Claim 17

(i) The memory device of claim 16, wherein the controller is configured to assign access priority levels to resolve conflicting interleaved memory access operations.

As discussed above, the *Ziv-Vogt* combination discloses the memory device of claim 16. Moreover, as discussed above in Section VIII.B.4, *Henkel* further discloses the controller assigning access priority levels to resolve conflicting interleaved memory access operations.

F. GROUND 6: Claims 10, 12-13, and 16-18 are unpatentable under 35 U.S.C. § 102 over *Sinclair*.

1. Independent Claim 10

(i) A memory device, comprising:

Sinclair discloses "the operation of re-programmable non-volatile memory systems such as semiconductor flash memory[.]" Ex. 1007 at 1:17-19; Fig. 1.

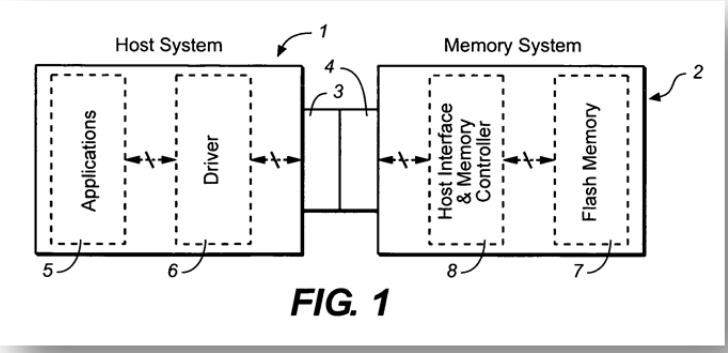


FIG. 1

Id. at Fig. 1.

(ii) one or more predefined access profiles to determine how access to the memory device is configured for at least one usage of the memory device;

Sinclair discloses a Reclaim Normal mode that is a predefined access profile that optimally interrupts host write operations (*i.e*., determine how access to the memory device is configured) based on an expected host activity (*i.e*., usage). Ex. 1007 at 23:28-30. In this mode, the memory device calculates an optimal interleave ratio of reclaim operations such that a memory card will not run out of writeable memory until there is no reclaimable memory left. *Id*. at Abstract. ("A memory controller ... schedules the reclaim operations to be evenly distributed between write operations until the memory is full."), 2:50-57, Fig. 19.

The selection of Reclaim Normal mode, for example, causes the memory controller to configure the device to optimally interleave memory access operations (*e.g*., write/read) with reclaim operations. *Id*. at Abstract. ("A memory controller ... schedules the reclaim operations to be evenly distributed between write operations until the memory is full.").

Reclaim On mode and Reclaim Off mode similarly determine how access to the memory device is configured based on an expected host activity (*i.e*., usage). In Reclaim On mode, the host sits "idle" by not sending additional commands as the

memory device performs continuous reclaim operations. *Id*. at 23:38-44. In Reclaim Off mode, the memory device inhibits reclaim operations and only host operations are performed. *Id.* at 23:55-56.

(iii) a controller configured to receive at least one first command to activate at least one of the one or more predefined access profiles associated with the memory device; and

The controller receives a command to select the appropriate reclaim mode. Ex. 1007 at 23:28-30 ("[T]he host may have commands to select the appropriate reclaim mode based on present host activity or expected host activity"); Ex. 1002 at ¶323. After receiving a reclaim mode command, such as the "Reclaim_normal" command, the controller begins reclaiming memory as specified by the selected profile. Ex. 1007 at 23:27-30. For instance, after receiving a "Reclaim_normal" command, the reclaiming occurs "according to an adaptive schedule" in Reclaim Normal mode. *Id*. at 24:1-3.

(iv) receive at least one second command to configure access to the memory device in accordance with the at least one of the one or more predefined access profiles such that at least a portion of the memory device is configured according to the at least one of the one or more predefined access profiles for the at least one usage.

The memory controller in *Sinclair* receives a second command that optimally configures the performance of memory access operations (*e.g.,* write operations) with reclaim operations (*i.e.*, usage) in accordance with the Reclaim Normal mode (*i.e.*, predefined access profile).

When "triggered by a host command," the memory controller recalculates the interleave ratio in the selected Reclaim Normal mode and configures the memory device to the recalculated interleave ratio. Ex. 1007 at 18:23-26, 3:3-6 ("An interleave ratio may be calculated at intervals or when there is a triggering event such as the deletion of some stored data by a host. Thus, ***the interleave ratio is updated as appropriate so that the ratio is adaptive to changing circumstances***.") (emphasis supplied); Ex. 1002 at ¶327. Accordingly, a portion of the memory device (*i.e.*, the memory in the memory integrated circuit chip) is now configured to operate according to the recalculated interleave ratio. Ex. 1007 at

18:17-23 ("The rate of reclaiming is modified as a result [of recalculating the interleave ratio] and the rate of programming valid data is also changed because the rate of reclaiming affects the rate of programming valid data"); Ex. 1002 at ¶328. Specifically, the portion of memory in the memory integrated circuit chip that is either being written to or reclaimed is configured in accordance with the recalculated interleave ratio. Ex. 1007 at 18:17-23; Ex. 1002 at ¶328. *Sinclair* even depicts a scenario where the controller reconfigures the reclaim operations for the memory device after receiving a host delete command. Ex. 1007 at 18:12-23; Ex. 1002 at ¶327. In Fig. 20 (shown below), when the host issues a delete command to the memory device at time t10, the system alters the interleave ratio such that all reclaimable space is reclaimed when the memory becomes full. Ex. 1007 at 18:12-26; Fig. 20 (shown below). Thus, the delete command configures access to the memory device in accordance with the particular reclaim profile selected for device access (*i.e.*, usage). Ex. 1002 at ¶¶327-28.

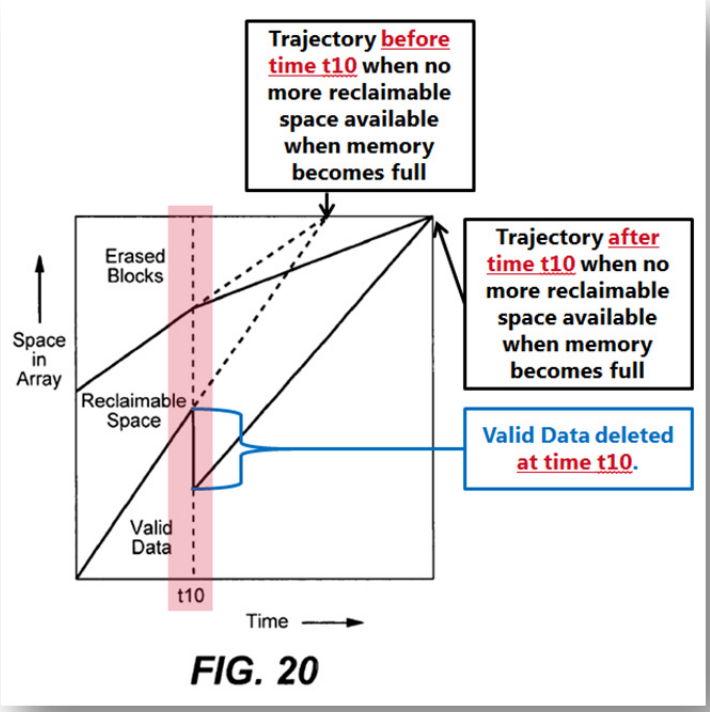


FIG. 20

Ex. 1007 at Fig. 20 (annotated).

2. Dependent Claim 12

(i) The memory device of claim 10, wherein at least one of the one or more predefined access profiles comprises a default access profile.

Sinclair discloses a default access profile, like the Reclaim Normal mode, that configures the memory device upon power up.. Ex. 1007 at 23:47-48; Ex. 1002 at ¶330.

3. Dependent Claim 13

(i) The memory device of claim 10, wherein the memory device comprises an embedded MultiMedia Card (eMMC) device.

The memory device in *Sinclair* may comprise an embedded MultiMediaCard. An exemplary memory device in *Sinclair* is a MultiMediaCard. Ex. 1007 at 4:28-32 ("There are currently many different flash memory cards that are commercially available, examples being the CompactFlash (CF), the MultiMediaCard (MMC), Secure Digital (SD), miniSD, Memory Stick, SmartMedia and TransFlash cards."). *Sinclair* further discloses that the MultiMediaCard may be embedded within the host. *Id*. at 4:25 ("[T]he flash memory can be embedded within the host[.]"). Accordingly, *Sinclair* discloses that an embedded Multimedia Card is an exemplary memory device to implement the various reclaim operation modes. Ex. 1002 at ¶¶332-34.

4. Dependent Claim 16

(i) The memory device of claim 10, wherein the controller is configured to interleave two or more simultaneous memory access operations.

The controller in *Sinclair* is configured to interleave two or more simultaneous access operations. By operating in Reclaim Normal mode, the controller calculates an optimal interleave ratio that interleaves write operations

with reclaim operations. Ex. 1007 at 2:50-57. Accordingly, these simultaneous access operations (*i.e.*, write operations that write to the memory and reclaim operations that reclaim writeable memory) are optimally interleaved. *Id*. at 18:23-26; Ex. 1002 at ¶336.

5. Dependent Claim 17

(i) The memory device of claim 17, wherein the controller is configured to assign access priority levels to resolve conflicting interleaved memory access operations.

Sinclair further resolves a simultaneous access conflict between a write operation that is attempting to write to memory with only unreclaimed memory remaining and a reclaim operation required to reclaim the unreclaimed memory. Ex. 1007 at 15:38-41 ("There is also a need for a system of carrying out reclaim operations in a way that has little or no adverse effect on other memory operations such as the programming of host data."); 15:20-22 ("[P]rogramming may stop when some minimum amount of erased space remains in the memory array."). In this simultaneous access conflict, the memory device gives preference for the reclaim operation to reclaim the unreclaimed memory before writing to the memory. *Id*. at 15:25-31 ("[T]he garbage collection operation may end ... when just enough space has been reclaimed to allow programming of host data to continue.

At time t2, programming of host data begins again and at time t3, writing of host data ceases and another garbage collection 30 operation begins."). Accordingly, an access priority level is provided to the reclaim operation by the memory device that trumps a simultaneous write operation when a minimum amount of erased space remains in memory. *Id*. at 15:36-45; Ex. 1002 at ¶¶339-41.

6. Dependent Claim 18

(i) The memory device of claim 10, wherein the one or more predefined access profiles comprise a plurality of predefined access profiles.

As discussed above, *Sinclair* discloses the memory device of Claim 10. *Sinclair* further discloses a plurality of predefined access profiles.

The Reclaim On, Reclaim Normal, and Reclaim Off mode each constitutes a predefined access profile. Ex. 1002 at ¶¶343-44. Each of these modes determines how access to the memory device is configured for a usage of the memory device. Ex. 1007 at 23:25-29. For example, the Reclaim Normal mode causes the memory controller to operate in an adaptive reclaim mode. *Id*. at Abstract. ("A memory controller ... schedules the reclaim operations to be evenly distributed between write operations until the memory is full."). As another example, the controller performs continuous reclaim operations in Reclaim On mode and, conversely, prohibits reclaim operations in Reclaim Off mode. *Id*. at 23:62-63, 24:7-8.

G. GROUND 7: Claim 13 is obvious over *Sinclair* in view of *eMMC*.

1. Dependent Claim 13

(i) The memory device of claim 10, wherein the memory device comprises an embedded MultiMedia Card (eMMC) device.

Ground 6 applies *Sinclair* to claim 10. If the Board concludes that Sinclair's disclosure of embedding an MMC device in a host is not an eMMC device, claim 13 is obvious over *Sinclair* in view of eMMC. Use of eMMC to implement the memory device of *Sinclair* would be an obvious design choice in view of the explicit teachings in eMMC of the benefits of employing flash memory in the eMMC standard format, as well as the teaching of *Sinclair* to embed the device in the host. *See* Ex. 1022 ("eMMC™ makes it easy to embed mass-storage flash memory on host systems."); Ex. 1007 at 4:25-32; Ex. 1002 at ¶¶297, 345-46; *supra* Section VIII.D.1.

H. GROUND 8: Claims 14-15 are unpatentable under 35 U.S.C. § 103(a) over *Sinclair* and *Henkel*.

1. Dependent Claim 14

(i) The memory device of claim 10, wherein the controller is configured to conduct two or more simultaneous memory access operations.

As discussed above, *Sinclair* discloses the memory device of Claim 10. In

addition, as discussed above in Section VIII.B.1, *Henkel* additionally discloses that the controller is configured to conduct two or more simultaneous memory access operations. Ex. 1002 at ¶¶27-39; *see supra* Section VIII.B.1.

It would have been obvious to include the access operation prioritization scheme as taught by *Henkel* in the *Sinclair* memory device based on the following reasons.

Explicit Teaching to Combine

Sinclair and *Henkel* are in a similar field, technology, and time frame. Ex. 1002 at ¶¶349-50. *Sinclair* and *Henkel* expressly teach a solution to control access to a memory device. Ex. 1007 at 1:7 ("A memory controller ... ***schedules the reclaim operations to be evenly distributed between write operations*** until the memory is full.") (emphasis added); Ex. 1005 at Abstract ("The invention provides an arbitration unit adapted ***for controlling accesses*** to a shared storage."); Ex. 1002 at ¶349. The *Sinclair* and *Henkel* patents also were filed within a one-year time frame. Ex. 1007; Ex. 1006.

Moreover, one of ordinary skill in the art would have especially been motivated to combine *Sinclair* and *Henkel,* because *Henkel* improves the scheduling of reclaim operations. Ex. 1006 at ¶52 (assigning a priority to "maintenance requests"). Improving the scheduling of maintenance requests (like reclaim operations) was the primary advantage for the Reclaim Normal mode in

Sinclair. Ex. 1002 at ¶351; *see, e.g.*, Ex. 1007 at 17:50-54 ("A system to manage space 50 in a memory array in this way carries out reclaim operations according to a schedule so that individual reclaim operations are distributed between individual host write operations to provide a constant speed of writing host data.").

An additional motivating reason to combine *Sinclair* with *Henkel* is that *Henkel* provides greater control for the host to order its operations. A host may impose requirements on read/write operations communicated to a memory device. Ex. 1006 at ¶¶7, 10 ("Incoming data streams that are received from a certain functional unit ... can be processed in accordance with the requirements imposed by said functional unit."). One of ordinary skill in the art would have been motivated by the teachings of *Henkel* to use host prioritization of data to gain the advantage of meeting "required properties of the ... data streams," while retaining the advantages of *Sinclair* of performing write and reclaim operations in accordance with a host-selected mode. Ex. 1002 at ¶351; Ex. 1006 at ¶7.

Known Technique to a Known Device to Yield Predictable Results

Base system: *Sinclair* discloses a memory device that optimally interleaves memory access operations (*e.g.*, write and reclaim operations). Ex. 1007 at 17:5-63; Ex. 1005 at 8:39-44; Ex. 1002 at ¶354.

Known technique: As shown by *Henkel*, a controller (*e.g.*, an arbitration unit) may prioritize multiple memory access operations (*e.g.*, read, write, and

management operations) in order to minimize delays. Ex. 1006 at ¶¶14, 52; Ex. 1002 at ¶354.

Predictable results and improved system: A POSITA would have recognized that implementing *Henkel's* memory access prioritization scheme in *Sinclair*'s memory device would yield the predictable result of a memory device that employed a memory access prioritization scheme to lessen access operation delays. Ex. 1002 at ¶354.

Moreover, it would have been obvious to use the memory access prioritization scheme taught by *Henkel* as the scheme is a desirable and efficient way of prioritizing and processing multiple memory access operations. Ex. 1002 at ¶355.

2. Dependent Claim 15

(i) The memory device of claim 14, wherein the controller is configured to assign access priority levels to resolve conflicting simultaneous memory access operations.

As discussed above, the *Sinclair-Henkel* combination discloses the memory device of Claim 14. In addition, as discussed above in Section VIII.B.2, *Henkel* further discloses the controller assigning access priority levels to resolve conflicting simultaneous memory access operations. Ex. 1002 at ¶241; *see supra*

Section VIII.B.2.

IX. CONCLUSION

SanDisk respectfully requests that *inter partes* review of the '850 Patent be instituted and that Claims 10 and 12-18 be cancelled as unpatentable under 35 U.S.C. § 318(b).

Date: May 10, 2017

Respectfully submitted,
BAKER BOTTS L.L.P.

/s/Eliot D. Williams/

Eliot D. Williams
Lead Counsel
Reg. No. 50,822
1001 Page Mill Road
Building One, Suite 200
Palo Alto, CA 94304
Phone: (650) 739-7511
Facsimile: (650) 739-7611
eliot.williams@bakerbotts.com

Brian W. Oaks (Reg. No. 44,981)
98 San Jacinto Blvd., Suite 1500
Austin, Texas 78701
Phone: (512) 322-2500
Facsimile: (512) 322-2501

Ebby Abraham (Reg. No. 73,399)
2001 Ross Avenue, Suite 600
Dallas, Texas 75201
Phone: (214) 953-6801
Facsimile: (214) 661-480

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 13,810 words and therefore complies with the type-volume limitations of 37 C.F.R. § 42.24(a). The word count was calculated by starting with Microsoft Word's total document word count and subtracting the words for the Table of Contents, the Exhibit List, the Mandatory Notices, the Certificate of Compliance, and the Certificate of Service

May 10, 2017

/s/Eliot D. Williams/
Eliot D. Williams

CERTIFICATE OF SERVICE ON PATENT OWNER UNDER

37 C.F.R. § 42.105

In accordance with 37 C.F.R. §§ 42.6(e) and 42.105, the undersigned certifies that on the 10th day of May, 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 '850 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 Matters identified herein:

MTL_Service@tensegritylawgroup.com

TENSEGRITY LAW GROUP, LLP
555 Twin Dolphin Drive, Suite 650
Redwood Shores, CA 94065
Telephone: (650) 802-6000

May 10, 2017

Respectfully submitted,
BAKER BOTTS L.L.P.

/s/Eliot D. Williams/

Eliot D. Williams
Lead Counsel
Reg. No. 50,822
1001 Page Mill Road
Building One, Suite 200
Palo Alto, CA 94304
Phone: (650) 739-7511
Facsimile: (650) 739-7611
eliot.williams@bakerbotts.com

Brian W. Oaks (Reg. No. 44,981)
98 San Jacinto Blvd., Suite 1500
Austin, Texas 78701
Phone: (512) 322-2500
Facsimile: (512) 322-2501

Ebby Abraham (Reg. No. 73,399)
2001 Ross Avenue, Suite 600
Dallas, Texas 75201
Phone: (214) 953-6801
Facsimile: (214) 661-480

ATTORNEYS FOR PETITIONER
SANDISK LLC