

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

KIOXIA AMERICA, INC., and KIOXIA CORPORATION

Petitioners,

v.

BITMICRO LLC,

Patent Owner.

Case No.: IPR2023-00741
U.S. Patent No. 9,135,190

DECLARATION OF R. JACOB BAKER, Ph.D., P.E.

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	1
	A. Industry Experience.....	2
	B. Academic Experience.....	6
	C. Other Relevant Experience.....	8
III.	MATERIALS CONSIDERED	9
IV.	TECHNOLOGY BACKGROUND.....	11
V.	Overview of the '190 Patent.....	13
	A. Specification of the '190 Patent	13
	B. Prosecution History for the '190 Patent	17
VI.	LEGAL PRINCIPLES USED IN ANALYSIS	24
	A. Prior Art.....	24
	B. Anticipation.....	25
	C. Obviousness.....	26
	D. Claim Construction	31
VII.	LEVEL OF ORDINARY SKILL IN THE ART.....	33
VIII.	CLAIM CONSTRUCTION	35
IX.	OVERVIEW OF THE PRIOR ART	42
	A. Cornwell Discloses a Memory Management Process and System Using a Memory Controller that Performs Memory Transactions on Memory Devices that Can Have Different Attributes	42
	B. Ware Discloses a Memory Controller for Performing Memory Transactions on Memory Devices Having Different Attributes	45
X.	THE PRIOR ART DISCLOSES ALL OF THE CLAIMED ELEMENTS OF THE '190 PATENT.....	48
	A. Each of the Challenged Claims Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge.....	48
	1. Claim 1 Is Rendered Obvious by Cornwell in View of a	

	POSITA's Knowledge	48
2.	Claim 5 Is Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	82
3.	Claim 6 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	84
4.	Claim 29 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	85
5.	Claim 36 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	108
6.	Claim 37 Is Rendered Obvious by Cornwell in view of a POSITA's Knowledge	110
7.	Claim 45 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	112
8.	Claim 51 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	139
9.	Claim 52 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	141
10.	Claim 59 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	142
11.	Claim 71 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	169
12.	Claim 76 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	171
13.	Claim 80 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	173
14.	Claim 84 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	194
15.	Claim 85 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	196
16.	Claim 98 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge	197
B.	Each of the Challenged Claims Is Rendered Obvious by Ware in View of a POSITA's Knowledge.....	216
1.	Claim 1 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	217
2.	Claim 5 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	252
3.	Claim 6 Is Rendered Obvious by Ware in View of a	

	POSITA's Knowledge	254
4.	Claim 29 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	255
5.	Claim 36 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	277
6.	Claim 37 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	279
7.	Claim 45 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	280
8.	Claim 51 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	305
9.	Claim 52 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	307
10.	Claim 59 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	308
11.	Claim 71 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	336
12.	Claim 76 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	338
13.	Claim 80 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	339
14.	Claim 84 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	362
15.	Claim 85 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	364
16.	Claim 98 Is Rendered Obvious by Ware in View of a POSITA's Knowledge	365
XI.	CONCLUSION.....	387

I, R. Jacob Baker, Ph.D., P.E., declare as follows:

I. INTRODUCTION

1. I have been retained by KIOXIA Corporation and KIOXIA America, Inc. (“Petitioners”) as an independent expert in this *inter partes* review before the United States Patent and Trademark Office (“PTO”) regarding U.S. Patent No. 9,135,190 (“the ’190 patent”), Ex-1001. Specifically, I have been asked to consider whether certain references disclose or suggest the features recited in claims 1, 5, 6, 29, 36, 37, 45, 51, 52, 59, 71, 76, 80, 84, 85, and 98 of the ’190 patent (“challenged claims”). As explained below, it is my opinion that the challenged claims are invalid as anticipated and/or obvious in view of the prior art references discussed in this Declaration and the accompanying Petition.

2. I am being compensated at my ordinary consulting rate of \$655.00 per hour. No part of my compensation is dependent on the outcome of this proceeding or any other proceeding involving the ’190 patent or any of its related patents. I have no other interests in this proceeding.

II. QUALIFICATIONS

3. My qualifications are generally set forth in my Curriculum Vitae, which is attached as Ex-1004. My Curriculum Vitae also includes a list of the publications I have authored and a list of the other cases in which I have testified during the last

four years.

4. I have been working as an Engineer since 1985 and have been teaching Electrical and Computer Engineering courses since 1991. I am currently a Professor of Electrical and Computer Engineering at the University of Nevada, Las Vegas (“UNLV”).

5. I received B.S. and M.S. degrees in Electrical Engineering from UNLV in 1986 and 1988, respectively. I received my Ph.D. in Electrical Engineering from the University of Nevada, Reno, in 1993.

6. My doctoral research, culminating in the award of a Ph.D., investigated the use of power MOSFETs in the design of very high peak power, and high-speed, instrumentation. I developed techniques to reliably stack power MOSFETs to switch higher voltages, that is, greater than 1,000 V and 100 Amps of current with nanosecond switching times. This work was reported in the paper entitled “Transformerless Capacitive Coupling of Gate Signals for Series Operation of Power MOSFET Devices,” published in the IEEE Transactions on Power Electronics. The paper received the Best Paper Award in 2000.

A. Industry Experience

7. I have done technical and expert witness consulting for over 200 companies and their subsidiaries since I started working as an Engineer in 1985.

From 1985 to 1993, I worked for EG&G Energy Measurements and the Lawrence Livermore National Laboratory designing nuclear diagnostic instrumentation for underground nuclear weapon tests at the Nevada test site. During this time, I designed, and oversaw the fabrication of, over 30 electronic and electro-optic instruments, including high-speed cable and fiber-optic receiver/transmitters, PLLs, frame and bit-syncs, data converters, streak-camera sweep circuits, Pockel's cell drivers, micro-channel plate gating circuits, charging circuits for battery backup of equipment for recording test data, and analog oscilloscope electronics.

8. My work during this time, as one example, had a direct impact on my doctoral research work using power MOSFETs, subsequent publishing efforts, and industry designs. In addition to the 2000 Best Paper Award from the IEEE Power Electronics Society, I published several other papers in related areas while working in industry. I hold a patent, Patent No. 5,874,830, in the area of power supply design, titled, "Adaptively biased voltage regulator and operating method," which was issued on February 23, 1999. I have designed dozens of linear and switching power supplies for commercial products and scientific instrumentation.

9. I am a licensed Professional Engineer and have extensive industry experience in circuit design, fabrication, and manufacture of Dynamic Random Access Memory (DRAM) semiconductor integrated circuit chips, Phase-Change

Random Access Memory (PCRAM) chips, and CMOS Image Sensors (CISs) at Micron Technology, Inc. (“MTI”) in Boise, Idaho. I spent considerable time working on the development of flash memory chips while at MTI. My efforts resulted in more than a dozen patents relating to flash memory. One of my projects at MTI included the development, design, and testing of circuit design techniques for a multi-level cell (MLC) flash memory using signal processing. This effort resulted in higher-density memories for use in solid-state drives and flash memory cards having an ATA interface that are ubiquitous in consumer electronics, including cameras and data storage systems. Further, the use of higher-density memory can result in fewer changes in the flash translation layer for logical-to-physical addressing, less need for garbage collection, and larger data segments that can improve a computing system’s performance.

10. Another project I worked on at MTI focused on the design of buffers for high-speed double-data rate DRAM, which resulted in around 10 U.S. patents in buffer design. Among many other experiences, I led the development of the delay locked loop (DLL) in the late 1990s so that MTI DRAM products could transition to the DDR memory protocol, used in mobile and non-mobile (server, desktop, cell phones, tablets, etc.) computing systems as main computer memory, for addressing and controlling accesses to memory via interprocess communications (IPC) with the

memory controller (MC). I provided technical assistance with MTI's acquisition of Photobit during 2001 and 2002, including transitioning the manufacture of CIS products into MTI's process technology. Further, I did consulting work at Sun Microsystems and then Oracle on the design of memory modules during 2009 and 2010. This work entailed the design of low-power, high-speed, and wide interconnection methods with the goal of transmitting data to/from the memory module and the MC at higher speeds.

11. I have extensive experience in the development of instrumentation and commercial products in a multitude of areas including: integrated electrical/biological circuits and systems, array (memory, imagers, and displays) circuit design, CMOS analog and digital circuit design, diagnostic electrical and electro-optic instrumentation for scientific research, CAD tool development and online tutorials, low-power interconnect and packaging techniques, design of communication/interface circuits (to meet commercial standards such as USB, firewire, DDR, PCIe, SPI, etc.), circuit design for the use and storage of renewable energy, and power electronics. For example, a part of my research at Boise State, for many years, focused on the use of Thru-Silicon-Vias (TSVs), aka Thru-Wafer Vias (TWVs), for high-density packaging. These packaging techniques were utilized in the memory module development work I did with Sun Microsystems and Oracle. As

another example, I designed circuitry for use in implementing Universal Serial Bus (USB) interface circuits while consulting at Tower Semiconductor. I designed PCI communication circuits for IPC between a Graphics Processor Unit (GPU) and memory while consulting for Rendition, Inc.

12. My current research work is focused in part on the design of integrated circuits for wireless sensing using LIDAR (Light Detection And Ranging). I have worked with several companies in the development of these circuits and systems including Freedom Photonics, Aerius Photonics, and FLIR. In the early 1990s, I worked on wireless systems for wideband impulse radar while at Lawrence Livermore Laboratory. Further, part of my research for several years focused on the digitization of IQ channels using delta-sigma modulation. The knowledge and experience gained from this effort are reflected in my textbook *CMOS Mixed-Signal Circuit Design* and a presentation, which I have presented at several universities and companies, http://cmosedu.com/jbaker/papers/talks/BP_DSM_talk.pdf.

B. Academic Experience

13. I was an adjunct faculty member in the Electrical Engineering departments of UNLV and UNR from 1991-1993. From 1993 to 2000, I served on the faculty at the University of Idaho as an Assistant Professor and then as a tenured Associate Professor of Electrical Engineering. In 2000, I joined a new Electrical and

Computer Engineering program at Boise State University (“BSU”), where I served as department chair from 2004 to 2007. At BSU, I helped establish graduate programs in Electrical and Computer Engineering including, in 2006, the university’s second Ph.D. degree. In 2012, I re-joined the faculty at UNLV. Over the course of my career as a professor, I have advised more than 100 masters and doctoral students.

14. I have been recognized for my contributions as an educator in the field. While at Boise State University, I received the President’s Research and Scholarship Award (2005), Honored Faculty Member recognition (2003), and Outstanding Department of Electrical Engineering Faculty recognition (2001). In 2007, I received the Frederick Emmons Terman Award (the “Father of Silicon Valley”), which is bestowed annually upon an outstanding young electrical/computer engineering educator in recognition of the educator’s contributions to the profession. In 2011, I received the IEEE Circuits and Systems Education Award. I received the Tau Beta Pi Outstanding Electrical and Computer Engineering Professor Award every year it was awarded while I have been back at UNLV.

15. I have authored several books and papers in the electrical and computer engineering area. My published books include *CMOS Circuit Design, Layout, and Simulation* (Baker, R.J., Wiley-IEEE, ISBN: 9781119481515 (4th ed., 2019)) and

CMOS Mixed-Signal Circuit Design (Baker, R.J., Wiley-IEEE, ISBN: 9780470290262 (2nd ed., 2009) and ISBN: 9780471227540 (1st ed., 2002)). I co-authored *DRAM Circuit Design: Fundamental and High-Speed Topics* (Keeth, B., Baker, R.J., Johnson, B., and Lin, F., Wiley-IEEE, ISBN: 9780470184752 (2008)), *DRAM Circuit Design: A Tutorial* (Keeth, B. and Baker, R.J., Wiley-IEEE, ISBN: 0-7803-6014-1 (2001)), and *CMOS Circuit Design, Layout and Simulation* (Baker, R.J., Li, H.W., and Boyce, D.E., Wiley - IEEE, ISBN: 9780780334168 (1998)). I contributed as an editor and co-author on several other electrical and computer engineering books.

C. Other Relevant Experience

16. I have performed technical and expert witness consulting for more than 200 companies and their subsidiaries and given more than 50 invited talks at conferences, companies, and universities. Further, I am the author or co-author of more than 100 papers and presentations in the areas of electrical and computer engineering design, fabrication, and packaging.

17. I currently serve, or have served, as a volunteer on the IEEE Press Editorial Board (1999-2004); as editor for the Wiley-IEEE Press Book Series on Microelectronic Systems (2010-2018); as the Technical Program Chair of the 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS

2015); on the IEEE Solid-State Circuits Society (SSCS) Administrative Committee (2011-2016); as a Distinguished Lecturer for the SSCS (2012-2015); the Technology Editor (2012-2014) and Editor-in-Chief (2015-2020) for IEEE Solid-State Circuits Magazine; IEEE Kirchhoff Award Committee (2020-2023); and advisor for the student branch of the IEEE at UNLV (2013-present). These meetings, groups, and publications are intended to allow researchers to share and coordinate research. My active participation in these meetings, groups, and publications allows me to see what other researchers in the field have been doing.

18. In addition to the above, I am an IEEE Fellow for contributions to semiconductor memory design and a member of the honor societies Eta Kappa Nu and Tau Beta Pi.

III. MATERIALS CONSIDERED

19. My opinions, as set forth in this Declaration, are based on the documents I have reviewed, my experience and background, and my knowledge and professional judgment. In forming these opinions, I have reviewed the '190 patent (Ex-1001), the prosecution history for the '190 patent (Ex-1002), the prior art discussed below, and any other material and information identified in this Declaration. I have also relied on my experience and knowledge in the field of computer memory and data storage devices.

20. In particular, my opinions regarding the invalidity of the challenged claims are based on the following prior art references, as informed by my knowledge in the field of computer memory and data storage devices.

Exhibit	Description
1005	U.S. Patent App. Pub. No. 2007/0180186 (“Cornwell”). It is my understanding that Cornwell was filed on January 27, 2006, and published on August 2, 2007. Petitioners’ counsel has informed me that Cornwell is prior art under pre-AIA 35 U.S.C. § 102(e). See <i>infra</i> § VI.B for a more detailed description regarding my understanding of this category of prior art.
1006	U.S. Patent No. 7,269,708 (“Ware”). It is my understanding that Ware was filed on April 20, 2004, and issued on September 11, 2007. Petitioners’ counsel has informed me that Ware is prior art under pre-AIA 35 U.S.C. § 102(b). See <i>infra</i> § VI.B for a more detailed description regarding my understanding of this category of prior art.
1007	<i>BiTMICRO LLC v. KIOXIA Am. Inc., et al.</i> , Case No. 6:22-cv-00331-ADA, Dkt. No. 54 (W.D. Tex. Feb. 16, 2023) (Claim Construction Order in co-pending litigation)
1010	U.S. Patent App. Pub. No. 2005/0108500 (“Kishida”).

Exhibit	Description
1011	BRENT KEETH ET AL., DRAM CIRCUIT DESIGN (Mohamed E. El-Hawary et al. eds., IEEE Press 2008)

IV. TECHNOLOGY BACKGROUND

21. The '190 patent relates to a memory controller that can control different types of semiconductor memories such as dynamic random access memories (DRAMs) and flash memories. The '190 patent alleges that, prior to its filing, multi-profile memory controllers had been traditionally designed to operate with the same types of memories having the same characteristics, such as block size, density, speed, and type. Ex-1001 at 1:21-24. According to the '190 patent, flash memory devices that have a relatively large block size generally provide better sequential memory access performance than flash memory devices that have a relatively small block size. *Id.* at 1:27-31. These devices are also more cost effective than flash memory devices with lower density. *Id.* at 34-37. However, the '190 patent contents that these devices also suffer from relatively higher read-modify-write cycle latency, higher random memory access latency and lower random input/output operations per second compared to a lower density memory system when operating with the same memory device profiles. *Id.* at 1:38-43. Accordingly, the '190 patent purports

to address “a need for a multi-profile memory controller that can operate with memory locations, memory devices, or both that are associated with different memory attributes, different attribute qualifiers, or the like, while minimizing or avoiding some or all of the above described disadvantages.” *Id.* at 1:44-48.

22. However, multi-profile memory controllers that operate with different types of memory, each having device profiles with different attributes, predate the September 4, 2009, priority date for the '190 patent. For example, as explained below, Cornwell discloses a memory controller “operatively coupled to [] one or more non-volatile memory devices and configurable to access the one or more non-volatile memory devices in accordance with a memory management policy.” Ex-1005, ¶ [0007]. This memory controller can operate with different types of memory, such NOR and NAND flash, each of which can have different attributes including density, block size, page size, etc. Ex-1005, ¶ [0042].

23. Likewise, Ware is directed to “a memory controller for a non-homogenous memory.” Ex-1006 at 1:7-8. This controller can be operatively connected to two different types of memory, such DRAM and flash. *Id.* at 3:56-61; Figure 1. As a result, the memory controller “can be configured to exploit the different attributes of the memory device types in the respective portions of main memory so as to improve one or more aspects of memory system performance.” *Id.*

at 4:47-52.

24. In conclusion, well before the September 2009 priority date of the '190 patent, there was a long history of memory controllers that operated with different types of memory, each having different attributes.

V. OVERVIEW OF THE '190 PATENT

A. Specification of the '190 Patent

25. I understand that the application for the '190 patent was filed on September 4, 2010, and claims priority to a provisional application filed on September 4, 2009. Challenged claims 1, 29, 45, 59, 80, and 98 of the '190 patent are independent claims, from which the remainder of the challenged claims depend.

26. The '190 patent purports to relate to solid-state storage devices utilizing multi-profile memory controllers. Ex-1001 at 1:17-19. Notably, however, the '190 patent does not purport to invent a new type of memory, multi-profile memory controller, or communication protocol for memory devices and controllers. To the contrary, the '190 patent confirms that the memory devices discussed in the specification, such as Flash memory, SRAM, or DRAM, were well-known at the time of filing. *Id.* at 1:21-37; 4:47-5:20. The '190 patent also acknowledges that multi-profile memory controllers for use with these memory devices were well-known at the time of filing, having “been traditionally designed to operate with

memory stores or modules that comprise of memory devices that use the same memory device characteristics.” *Id.* at 1:21-24. Similarly, the ’190 patent makes clear that the recited methods and systems can simply utilize known standards and protocols, such as the Open NAND Flash Interface Specification (“ONFI”), which is a “known device interface standard created by a consortium of technology companies.” *Id.* at 4:47-61; *see also id.* at 4:61-5:3 (explaining that the recited devices may also utilize the Non-Volatile Memory Host Controller Interface protocol (“NVMHCI”), which, as the ’190 specification explains, was developed by a working group including both Intel and Microsoft.). Nonetheless, the ’190 patent purports to identify a need for a multi-profile memory controller that can operate with memory devices having different memory attributes. *Id.* at 1:43-48.

27. To address this alleged need, the ’190 patent purports to describe a “multi-profile memory controller disposed to perform memory transactions using different device profiles.” *Id.* at 2:44-47. According to the ’190 patent, a memory store may include multiple memory locations, each of which are associated with at least one memory “attribute” or “characteristic of a memory location that is related to accessing or addressing the memory location.” *Id.* at 2:63-3:2. These attributes, which are collectively organized into a “device profile,” may include such things as memory device type, block size, page size, and partial page size. *Id.* at 3:2-15.

According to the '190 patent, a multi-profile memory controller can perform memory transactions using the different profiles for these memory devices. *Id.* at 3:61-63; *see also id.* at 2:44-47. The specification alleges that information can then be mapped to a memory location that is associated with a device profile that is optimal for the type of data subject to the memory transaction request. *Id.* at 9:7-16.

28. Annotated Figure 1 of the '190 patent, shown below, illustrates the system of the '190 patent, including a memory store comprising a **first memory device** with a **first memory location**, a **second memory device** with a **second memory location**, and a **memory controller**:

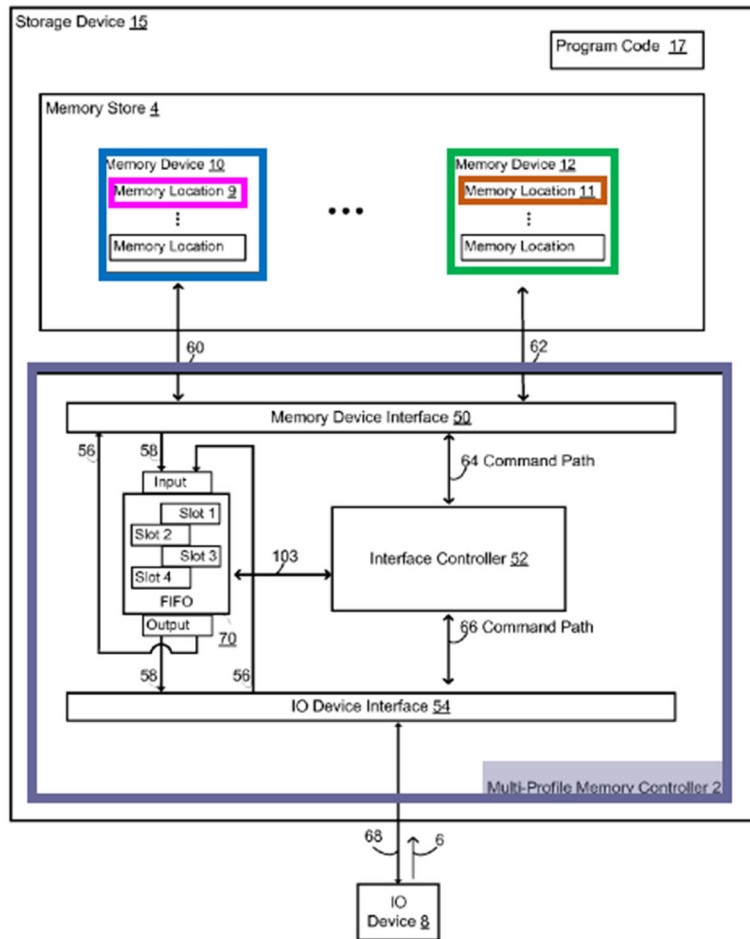


FIG. 1

29. As explained below, however, multi-profile memory controllers that performed memory transactions on memory locations having different device profiles or attributes were well known. Using device profiles to determine a specific memory location for storing information was similarly well-known. Both Cornwell and Ware, discussed below, disclose such systems and methods. As a result, nothing recited in the claims of the '190 patent would have been novel or non-obvious to a person of ordinary skill in the art ("POSITA") at the time the application for the '190

patent was filed.

B. Prosecution History for the '190 Patent

30. As explained above, it is my understanding that the application for the '190 patent was filed on September 4, 2010. I understand that, as originally filed, claim 1 recited the following method:¹

A method of performing memory transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:

addressing a first memory location, said first memory location and a second memory location respectively associated with a first device profile and a second device profile;

said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles; and

using attributes from said first and second device profiles.

Ex-1002 at 1.

31. I understand that, on December 21, 2012, the Examiner issued a non-final Office Action, rejecting claims 1-14, 20, 21, 25, 45-60, and 69-92 as anticipated by U.S. Patent No. 8,200,879 ("Falik"). *Id.* at 22-45. The Examiner also found the

¹ For purposes of this discussion, I have focused on claim 1, as it recites elements common to many of the other independent claims.

remaining claims obvious over Falik in view of U.S. 7,330,954 (“Nangle”), U.S. Patent App. Pub. No. 2006/0184723 (“Sinclair”), U.S. Patent No. 6,961,805 (“Lakhani”), U.S. Patent No. 5,913,215 (“Rubinstein”), U.S. Patent App. Pub. No. 2001/0010066 (“Chin”), and/or U.S. Patent App. Pub. No. 2007/0174506 (“Tsuruta”). *Id.* at 46-71.

32. It is my understanding that, on March 26, 2013, the Applicant attempted to overcome the rejection. *Id.* at 73-81. In doing so, the Applicant did not propose any further amendments to claim 1. *Id.* at 82. I understand that the Examiner was not persuaded by these arguments and issued a Final Office Action May 14, 2013, maintaining the positions from the previous rejection. *Id.* at 104-160.

33. In response to this rejection, I understand that, on October 21, 2013, the Applicant amended claim 1 as follows (amendments shown in red):

A method of performing memory transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:

addressing a first memory location, said first memory location and a second memory location respectively associated with a first device profile and a second device profile, wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location, wherein data is transferred to the first memory location or second memory location for a memory write transaction, wherein data is transferred from the first memory location or second memory location for a memory read transaction;

said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles; and

using attributes from said first and second device profiles.

Id. at 161-163, 170-171.

34. I understand that, on March 13, 2014, the Examiner again rejected the claims, explaining that the Applicant's arguments had been fully considered but were not found persuasive. *Id.* at 246-249. In particular, I understand that the Examiner maintained the position that claims 1-14, 20, 21, 25, 45-60, 69-92, and 94-97 were anticipated by Falik. *Id.* at 195-221. The Examiner also maintained the position that the remaining claims were rendered obvious by Falik in view of Nangle, Sinclair, Lakhani, Rubinstein, Chin, and/or Tsuruta. *Id.* at 221-246.

35. In response to this rejection, I understand that, on June 19, 2014, the Applicant further amended claim 1 as shown below (additional amendments shown in pink):

A method of performing memory transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:

addressing a first memory location in a first memory store, said first memory location and a second memory location respectively associated with a first device profile and a second device profile, wherein said second memory location is in a second memory

device in the memory store, wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location, wherein data is transferred to the first memory location or second memory location for a memory write transaction, wherein data is transferred from the first memory location or second memory location for a memory read transaction;

said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles; and

wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory location;

wherein the memory store is coupled by at least one memory bus to a controller, and wherein the controller performs the memory transactions on the memory store;

wherein the first device profile and second device profile are each stored in the memory store; and

using attributes from said first and second device profiles.

Id. at 251-252, 258-259.

36. It is my understanding that, on July 11, 2014, the Examiner issued another Office Action rejecting the claims. *Id.* at 283-345. Once more, the Examiner found that—notwithstanding Applicant’s amendments—claims 1-14, 20, 21, 25, 45-60, 69-92, and 94-97 were still anticipated by Falik. *Id.* at 286-315. Likewise, the Examiner maintained that the remaining claims were rendered obvious

by Falik in view of Nangle, Sinclair, Lakhani, Rubinstein, Chin, and/or Tsuruta. *Id.* at 316-342.

37. It is my understanding that, in response to this rejection, the Applicant amended claim 1 further on October 7, 2014, as shown below (additional amendments shown in [blue](#)):

A method of performing memory transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:

addressing a first memory location [in a first memory store](#), said first memory location and a second memory location respectively associated with a first device profile and a second device profile, [wherein said second memory location is in a second memory device in the memory store, wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location, wherein data is transferred to the first memory location or second memory location from a host for a memory write transaction, wherein data is transferred from the first memory location or second memory location to the host for a memory read transaction;](#)

said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles; and

[wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory location;](#)

[wherein the memory store is directly coupled \[\\[\\[by\\]\\]\]\(#\) to at least one memory bus and wherein the at least one memory bus is](#)

directly coupled to a controller, and wherein the controller performs the memory transactions on the memory store;

wherein the host is directly coupled to the controller by a communication path as the controller performs the memory read transaction or memory write transaction;

wherein the first device profile and second device profile are each stored in the memory store; and

using attributes from said first and second device profiles.

Id. at 346-348, 354-355.

38. On December 5, 2014, the Examiner issued another Office Action rejecting the claims. *Id.* at 379-446. In this rejection, the Examiner found that claims 1-14, 20, 21, 25, 45-60, 69-92, and 94-101 were rendered obvious by U.S. Patent App. Pub. No. 2011/0167204 (“Estakhri”) in view of Falik. *Id.* at 383-418. The remaining claims were rejected as obvious by Estakhri in view of Falik in further view of Nagle, Sinclair, Lakhani, Rubinstein, Chin, and/or Tsuruta. *Id.* at 419-445.

39. It is my understanding that, in response to this rejection, the Applicant amended the claims still further on May 8, 2015, as shown below (additional amendments shown in green):

A method of performing memory transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:

performing a memory transaction comprising addressing a first memory location in a first memory store, said first memory

location and a second memory location respectively associated with a first device profile and a second device profile, wherein said second memory location is in a second memory device in the memory store, wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location, wherein data is transferred to the first memory location or second memory location from a host for a memory write transaction, wherein data is transferred from the first memory location or second memory location to the host for a memory read transaction;

wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;

said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;

said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;

after identifying the command details, obtaining the first set of attributes;

wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory location;

wherein the memory store is directly coupled [[by]] to at least one memory bus and wherein the at least one memory bus is directly coupled to a controller, and wherein the controller performs the memory transactions on the memory store;

wherein the host is directly coupled to the controller by a communication path as the controller performs the memory read transaction or memory write transaction;

wherein the first device profile and second device profile are each stored in the memory store;

using attributes from said first and second device profiles; and
selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.

Id. at 447, 450-451.

40. On June 22, 2015, the Examiner subsequently issued a Notice of Allowance. *Id.* at 469-478. As explained below, however, both Cornwell and Ware—which were not before the Examiner during prosecution—disclose the elements shown above in green, which formed the basis for the Notice of Allowance.

VI. LEGAL PRINCIPLES USED IN ANALYSIS

41. I am not an attorney and I offer no opinions on the law. I have been advised by Petitioners' counsel of certain legal principles relevant to an invalidity analysis, as summarized below. I have conducted my analysis in accordance with these principles.

A. Prior Art

42. I understand that a patent or other publication must qualify as prior art before it can be used to invalidate a patent claim.

43. I understand that a U.S. or foreign patent qualifies as prior art to a challenged claim if the patent was issued prior to the invention date of the challenged claim. I also understand that a U.S. or foreign patent or published patent application qualifies as prior art to a challenged claim if it was published before the invention date of the challenged claim.

B. Anticipation

44. I understand that a prior art reference “anticipates” an asserted claim, thereby rendering the claim invalid, if all elements of the claim are disclosed in the reference, either expressly or inherently. I understand that an element is disclosed inherently in a prior art reference if it is necessarily present in the prior art reference or is the natural result flowing from the disclosure of the prior art reference. I have been informed that a prior art reference does not need to use the same words as the challenged claim in order to anticipate the claim.

45. Petitioners’ counsel informs me that the application for the ’190 patent was filed before the America Invents Act (“AIA”) was enacted. Accordingly, I have been informed that pre-AIA 35 U.S.C. § 102 defines the types of references that can be used as prior art against the challenged claims.

46. I understand that under pre-AIA 35 U.S.C. § 102(b), a challenged claim is anticipated if was patented or described in a printed publication in this or a foreign

country, or in public use or on sale in this country, more than one year prior to the date of the patent application in the United States.

47. I also understand that under pre-AIA 35 U.S.C. § 102(e), a challenged claim is anticipated if it was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent, or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent.²

C. Obviousness

48. I also understand that a patent is invalid if the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious to a person of ordinary skill in the pertinent art at the time the invention was made. I understand that a person of ordinary skill in the art

² For subsection (2), I understand that an international application filed under the treaty defined in section 351(a) shall have the effects for the purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language. In this case, Cornwell is not an international application; it is United States Patent Application Publication.

provides a reference point from which the prior art and claimed invention should be viewed. This reference point prevents one from using his or her own personal insight or hindsight in deciding whether a claim is obvious.

49. I understand that an obviousness determination includes the consideration of various factors, such as (1) the scope and content of the prior art, (2) the differences between the prior art and the asserted claims, (3) the level of ordinary skill in the pertinent art, and (4) the existence of secondary considerations such as commercial success, long-felt but unresolved needs, failure of others, etc.

50. I understand that an obviousness evaluation can be based on a combination of multiple prior art references. I understand that the prior art references themselves may provide a suggestion, motivation, or reason to combine, which does not need to be explicit. I further understand that any need or problem known in the field and addressed by the patent can provide a reason for combining the elements in the manner claimed. To determine whether there was an apparent reason to combine the known elements in the way a patent claims, I have been informed that I may look to interrelated teachings of multiple patents, to the effects of demands known to the relevant community or present in the marketplace, and to the background knowledge possessed by a person of ordinary skill in the art. I also understand that practical and common-sense considerations should guide a proper

obviousness analysis because familiar items may have obvious uses beyond their primary purposes. I further understand that a person of ordinary skill in the art looking to overcome a problem will often be able to fit together the teachings of multiple publications. I understand that an obviousness analysis therefore accounts for the inferences and creative steps that a person of ordinary skill in the art would employ under the circumstances.

51. I understand that the nexus linking two or more prior art references can also be simple common sense. I further understand that the obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may be supplied by the direction of the marketplace.

52. I understand that if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.

53. I understand that a particular combination may be proven obvious merely by showing that it was obvious to try the combination. For example, when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good

reason to pursue the known options within his or her technical grasp because the result is likely not the product of innovation, but of ordinary skill and common sense.

54. I understand that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. When a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different field. If a person of ordinary skill can implement a predictable variation, the patent claim is likely obvious.

55. As noted above, I understand that a proper obviousness analysis focuses on what was known or obvious to a person of ordinary skill in the art, not just the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

56. I understand that a claim can be obvious in light of a single reference, without the need to combine references, if the elements of the claim that are not found explicitly or inherently in the reference can be supplied by the common sense or knowledge of one of skill in the art.

57. I also understand that secondary indicia of non-obviousness may include: (1) a long-felt but unmet need in the prior art that was satisfied by the

invention of the patent; (2) commercial success of processes covered by the patent; (3) unexpected results achieved by the invention; (4) praise of the invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the invention; (7) failure of others to find a solution to the long-felt need; and (8) skepticism by experts.

58. I also understand that there must be a relationship between any such secondary considerations and the invention. I further understand that contemporaneous and independent invention by others is a secondary consideration supporting an obviousness determination.³

³ In this case, I am not aware of any such secondary considerations that would support the conclusion that the claims of the '190 patent are not obvious. To the contrary, it does not appear that there was any long-felt but unmet need for the alleged invention; to the contrary, as explained below, memory devices with different profiles and attributes—as well as multi-profile memory controllers used to perform memory transactions on such memory devices—were well-known before the priority date of the '190 patent. I am also not aware of any evidence that BiTMICRO Networks, Inc. (the assignee shown on the face of the '190 patent) had commercial success marketing any product covered by the '190 patent. In addition,

59. In sum, it is my understanding that prior art teachings are properly combined where a person of ordinary skill in the art—having the understanding and knowledge reflected in the prior art and being motivated by the general problem facing the inventor—would have been led to make the combination of elements recited in the claims. Under this analysis, the prior art references themselves, or any need or problem known in the field of endeavor at the time of the invention, can provide a reason for combining the elements of multiple prior art references in the claimed manner.

D. Claim Construction

60. I understand that, before an invalidity analysis can be performed, the scope and meaning of the challenged claims must be determined. It is my

I am unaware of any unexpected results achieved by the alleged invention; instead, as described below, prior art systems and methods used memory devices with different profiles and multi-profile memory controllers to perform memory transactions. I am also unaware any of any praise of the '190 patent by those skilled in the art, any deliberate copying of the '190 patent, any failure of others to find a solution, or any skepticism by experts regarding the systems recited in the '190 patent.

understanding that this is accomplished through the process of claim construction.

61. I have been informed and understand that claims are generally construed from the perspective of one of ordinary skill in the art at the time of the invention, which I have been asked to assume is the '190 patent's priority date of September 4, 2009. More specifically, I understand that the claims should be afforded a meaning in accordance with the ordinary and customary meaning of the claim, as understood by one of ordinary skill in the art at the time of the invention.

62. I have been advised that, in construing a claim term, one must first look to the "intrinsic" patent evidence, which includes the claims themselves, the patent specification, and the prosecution history of the patent. I have also been informed that "extrinsic" evidence, which is outside the patent and prosecution history, may be useful in construing patent claims. It is my understanding that such extrinsic evidence may include dictionaries, textbooks, treatises, and similar sources of information.

63. I have been advised that a claim may recite a function to be performed rather than a structure for performing the function. I have been informed that such claim terms are known as "means-plus-function" claim terms. I understand that a term introduced by the phrase "means for" is presumed to be a means-plus-function claim term. I have also been informed that a claim term may be construed as a means-

plus-function term where it fails to recite sufficiently definite structure, or else recites function without reciting sufficient structure for performing that function.

64. I have been advised that the construction of a means-plus-function claim term is a two-step process. I understand that the first step requires an identification of the function recited in the claim. I understand that the second step requires determining what structure is disclosed in the specification for performing the recited function. I have been advised that a structure disclosed in the patent specification qualifies as corresponding structure if the intrinsic evidence clearly links or associates that structure to the function recited in the claim. I have been advised that the proper construction for a means-plus-function term is restricted to the corresponding structure identified in the specification and equivalents thereof.

65. I understand that claims must be construed in an *inter partes* review using the same claim construction standard that would be used to construe the claim in a civil action before a district court.

VII. LEVEL OF ORDINARY SKILL IN THE ART

66. My opinions have been guided by my appreciation of how a person of ordinary skill in the art (described below) would have understood the claims of the '190 patent as of the invention date, which I have been asked to assume is the '190 patent's priority date, September 4, 2009.

67. It is my understanding that a person of ordinary skill in the art is one who is presumed to be aware of all pertinent art, thinks along conventional wisdom in the art, and is a person of ordinary creativity. I further understand that the following factors may be considered in establishing the level of ordinary skill in the art in the relevant field: the type of problems encountered in the art; the prior art solutions to those problems; the rapidity with which innovations are made; sophistication of the technology; and the educational level of active workers in the field.

68. Based on the above factors, the materials and information I have reviewed, and my extensive experience with the subject matter of the '190 patent around the date of the alleged invention, it is my opinion that a person of ordinary skill in the art related to the technology of the '190 patent would have had at least a Bachelor's degree in electrical or computer engineering or a related field, and at least two years of work experience in the computer memory or data storage industry. Someone with less technical education but more experience, or more technical education but less experience, would also meet this standard.

69. Based on my education and experience in the fields of electrical engineering, computer memory, and data storage, I believe I am well qualified to provide opinions about how one of ordinary skill in the art would have interpreted

and understood the claims of the '190 patent and the prior art discussed below

VIII. CLAIM CONSTRUCTION

I have set forth my understanding of relevant claim construction principles above, which I have followed in my analysis. Below I discuss Petitioners' constructions of several claim terms. In my view, the remaining claim terms should be given their plain and ordinary meaning.

A. "optimal"

70. Claims 1, 29, 45, 59, 80, 84, 85, and 98 recite that a "first device profile is optimal for a data type subject to the memory transaction." Counsel informs me that the district court has construed the term "optimal" as having a plain and ordinary meaning. *See* Ex-1007 at 42-45. For the purposes of this declaration, I have been asked to adopt this construction in my analysis.

B. "means for performing a memory transaction"

71. Independent claim 45 recites a "means for performing a memory transaction." This term is introduced by the phrase "means for," which I am informed creates a presumption that it is a means-plus-function term. It is also my opinion that this phrase does not connote any particular structure. Accordingly, I have been asked to assume that this is a means-plus-function element and conduct my analysis in accordance with the two-step process for a means-plus-function term set forth above. *See supra* ¶¶ 63-64.

1. Step 1 – Recited Function

72. I understand that Petitioners have proposed that the recited function is “performing a memory transaction.” This function is recited verbatim in the language of claim 45. Accordingly, I agree with the Petitioners that “performing a memory transaction” is the recited function.

2. Step 2 – Corresponding Structure

73. I understand that Petitioners have proposed that the structure corresponding to the recited function is a multi-profile memory controller or similar structure configured to perform a memory transaction. Based on my review of the intrinsic record, together with the knowledge of a POSITA, I agree with this construction.

74. The specification of the '190 patent states that the invention is directed to a “***multi-profile memory controller*** disposed to ***perform a memory transaction.***” Ex-1001 at 2:44-47 (emphasis added); *see also id.* at 3:61-63 (“Multi-profile memory controller 2 can perform memory transactions using different device profiles.”); *id.* at 7:50-53 (noting that, “for each memory transaction requested, multi-profile memory controller 2 can perform memory operations on more than one memory location”).

75. Accordingly, I agree that this claim term recites the function of

“performing a memory transaction.” Based on the description in the specification and the knowledge of a POSITA, I further agree that the corresponding structure linked to this function is a multi-profile memory controller or similar structure configured to perform a memory transaction.

C. “interface controller means”

76. Independent claim 45 also recites an “interface controller means for processing memory transaction requests.” This term uses the phrase “means for,” which I am informed creates a presumption that it is a means-plus-function term. It is also my opinion that this phrase does not connote any particular structure. Accordingly, I have been asked to assume that this is a means-plus-function element and conduct my analysis in accordance with the two-step process for a means-plus-function term set forth above. *See supra* ¶¶ 63-64.

1. Step 1 – Recited Function

77. I understand that Petitioners have proposed that the recited function is “processing memory transaction requests.” This function is recited verbatim in the language of claim 45. Accordingly, I agree with the Petitioners that “performing a memory transaction” is the recited function.

2. Step 2 – Corresponding Structure

78. I understand that Petitioners have proposed that the structure corresponding to the recited function is an interface controller or similar structure

configured to process memory transaction requests. Based on my review of the intrinsic record, together with the knowledge of a POSITA, I agree with this construction.

79. The specification of the '190 patent states that “[i]n **interface controller 52 processes memory transaction request 6** by interpreting the memory transaction request and performing a memory transaction on a memory location.” Ex-1001 at 7:21-24; *see also id.* at 5:21-42 (describing the interface controller); Fig. 1 (illustrating interface controller 52).

80. Accordingly, I agree that this claim term recites the function of “processing memory transaction requests.” Based on the description in the specification and the knowledge of a POSITA, I further agree that the corresponding structure linked to this function is an interface controller or similar structure configured to process memory transaction requests.

D. “means for addressing a first memory location”

81. Independent claim 45 recites a “means for addressing a first memory location.” This term is introduced by the phrase “means for,” which I am informed creates a presumption that it is a means-plus-function term. It is also my opinion that this phrase does not connote any particular structure. Accordingly, I have been asked to assume that this is a means-plus-function element and conduct my analysis

in accordance with the two-step process for a means-plus-function term set forth above. *See supra* ¶¶ 63-64.

1. Step 1 – Recited Function

82. I understand that Petitioners have proposed that the recited function is “addressing a first memory location.” This function is recited verbatim in the language of claim 45. Accordingly, I agree with the Petitioners that “addressing a first memory location” is the recited function.

2. Step 2 – Corresponding Structure

83. I understand that Petitioners have proposed that the structure corresponding to the recited function is a combination of the following: a memory device interface or equivalent structure; an IO device interface or equivalent structure; an interface controller or similar structure configured to process memory transaction requests; and associated communication pathways. Based on my review of the intrinsic record, together with the knowledge of a POSITA, I agree with this construction.

84. Independent claim 45 outlines a number of requirements for the means for addressing. *First*, claim 45 recites that the means for addressing is part of the means for performing a memory transaction, which—as noted above—is a multi-profile memory controller or similar structure configured to perform a memory

transaction. *See* Ex-1001, claim 45 (reciting “means for addressing a first memory location comprising a means for addressing a first memory location”); ***Second***, claim 45 recites that the means for addressing “is directly coupled to the first memory location and second memory location.” *Id.* ***Third***, claim 45 recites that the means for addressing is also “directly coupled to a host as said means for addressing addresses the first memory location or second memory location.” ***Fourth***, claim 45 recites that the means for addressing “includes an interface controller means for processing memory transaction requests,” which, as explained above, is an interface controller or similar structure configured to process memory transaction requests.

85. The specification demonstrates that this structure would encompass a memory device interface (which is directly coupled to the memory locations as recited in the claim), an IO device interface (which is directly coupled to the host as recited in the claim), and the interface controller (which is part of the means for addressing as recited in the claims), and associated communication pathways between these components. This is confirmed by the following excerpt from the specification:

In FIG. 1, multi-profile memory controller 2 may be configured to include a memory device interface 50, an interface controller 52 and an IO device interface 54. Communication path 64 couples memory device interface 50 with interface controller 52, while communication path 66 couples interface controller 52 with IO device interface 54, which enables these coupled devices to assert or transmit signals across their respective communication paths. Interface controller 52 and memory device interface 50 communicate through communication bus 64. Communication through communication path 64 includes interface controller 52 sending information, such as command details, to memory device interface 50 and receiving status from memory device interface 50. In addition, interface controller 52 communicates with IO device interface 54 through communication path 66. Communication through communication path 66 includes interface controller 52 receiving status and request information from IO device interface 54 and sending commands to IO device interface 54. Communication paths 64 and 66 may be implemented in the form of a bus that includes control or command lines. A set of control or command lines are sometimes generically referred to as a control bus or a command line bus, respectively.

Id. at 5:21-42; see also Fig. 1.

86. Accordingly, I agree that this claim term recites the function of “addressing a first memory location.” Based on the description in the specification and the knowledge of a POSITA, I further agree that the corresponding structure linked to this function is a combination of the following: a memory device interface or equivalent structure; an IO device interface or equivalent structure; an interface controller or similar structure configured to process memory transaction requests;

and associated communication pathways.

IX. OVERVIEW OF THE PRIOR ART

A. Cornwell Discloses a Memory Management Process and System Using a Memory Controller that Performs Memory Transactions on Memory Devices that Can Have Different Attributes

87. Cornwell is directed to memory management systems and methods that utilize a memory controller, which is configured to retrieve and exploit attributes of different non-volatile memory devices coupled to the memory controller. Ex-1005, Abstract. As Cornwell explains, flash memory chips are divided into blocks, which consist of an array of memory cells that are organized into pages or sectors. *Id.*, ¶ [0003]. Cornwell further explains that, while flash memory can be read or programmed one byte at a time, it can only be erased one block at a time. *Id.*, ¶ [0004]. As a result, once a byte has been programmed, it cannot be changed until the entire block is erased. *Id.* Because flash memory has a finite number of erase-write cycles, however, it is necessary to minimize the number of those cycles to prolong the life of the memory. *Id.*, ¶ [0005]. Cornwell thus cites the need for systems, methods, and devices that can efficiently manage flash memory and other non-volatile memories, while still maintaining compatibility with existing standards and protocols. *Id.*

88. To address this need, Cornwell discloses a memory management

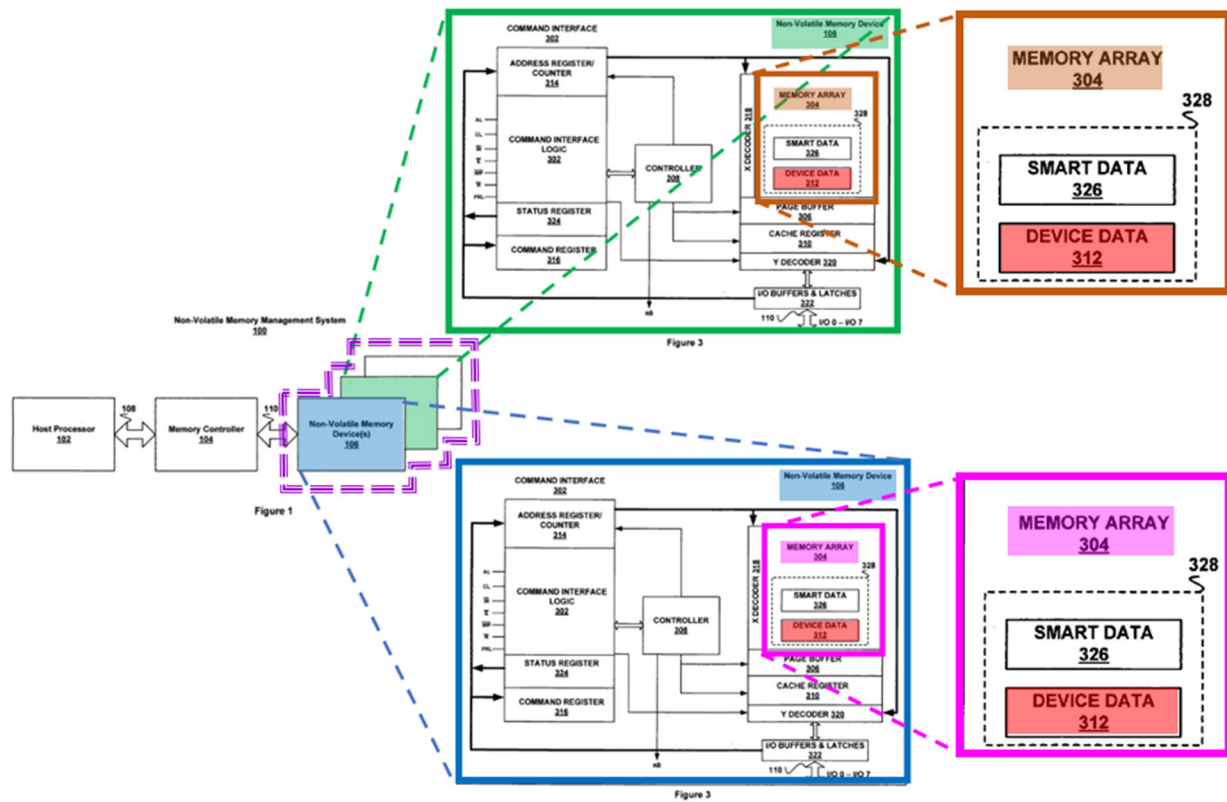
system that includes one or more non-volatile memory devices. *Id.*, ¶ [0007]. Cornwell further discloses a memory controller that is coupled to, and can determine information for, the non-volatile memory devices, including information associated with various attributes of the memory devices. *Id.*, ¶¶ [0007]-[0008]. These attributes, also referred to as “device data,” are used to identify the memory device. *Id.*, ¶ [0042]. For example, these attributes may include the device serial number, device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage, page size (1k, 2k, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64k, 128k, etc.), bus width, bad block identification, and “any other information that is associated with attributes, properties or characteristics of the memory device.” *Id.*

89. Cornwell also explains that different “memory devices 106” may have different densities depending on whether they utilize single-layer cell technology or multi-layer cell technology, teaching that “syncing with memory devices 106 that include multi-level (MLC) technology can be performed at a different frequency than with memory devices 106 that include single-level cell technology (SLC).” *Id.*, ¶ [0078]. Accordingly, Cornwell’s memory controller “supports single-level cell (SLC) and/or multi-level cell (MLC) flash media.” *Id.*, ¶ [0019].

90. As Cornwell explains, the memory controller includes a micro-

controller unit (“MCU”) that translates commands and signals from the host processor into data control signals required for memory operations. *Id.*, ¶¶ [0028]-[0029]. Moreover, in response to a read command, the memory controller obtains the device data or attributes from the memory devices, which it uses to perform memory management tasks. *Id.*, ¶¶ [0043]-[0044]. A memory management policy may also be determined based on the attributes of the non-volatile memory. *Id.*, Abstract; ¶ [0008].

91. Annotated Figures 1 and 3, below, illustrate Cornwell’s **device profile** (i.e., the collection of attributes comprising device data 312), which is associated with a **first memory location** in a **first memory device**, as well as a **second memory location** in a **second memory device**:



B. Ware Discloses a Memory Controller for Performing Memory Transactions on Memory Devices Having Different Attributes

92. Ware is directed to a memory controller for a non-homogenous memory system. Ex-1006 at 1:6-8. As Ware explains, computer program code and data typically reside on the main memory of a computer system, such as DRAM. *Id.* at 1:12-14. DRAM “has a near-optimal combination of operational attributes for implementing main memory,” including low cost, low read time, low write time, and unlimited endurance. *Id.* at 2:5-13. However, Ware explains that DRAM also has certain disadvantages such as volatility, which results in the loss of stored information when power is removed from DRAM cells. *Id.* at 3:5-12. In contrast,

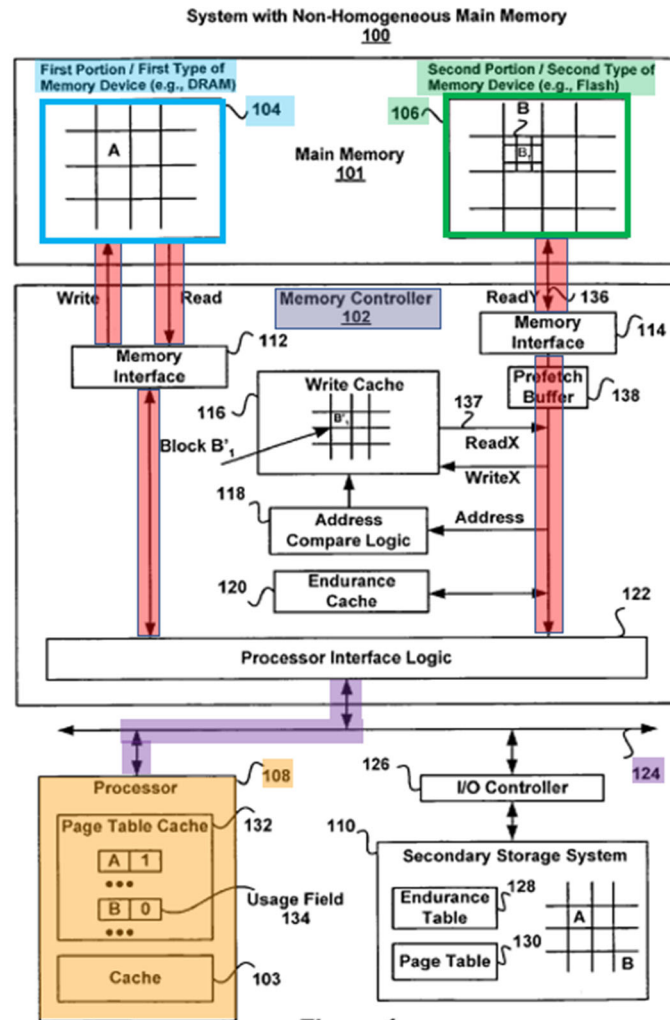
flash memory has different attributes compared to DRAM, including low read time and non-volatility, which “make it particularly attractive for use in portable computing applications.” *Id.* at 3:19-33.

93. Accordingly, Ware discloses a memory controller for a “non-homogenous memory system” in which the main memory is comprised of two or more memory “portions” or device types, such as DRAM and Flash. *Id.* at 2:18-25; 3:57-4:11. Each of these memory types may have different attributes, including read time, write time, endurance (the number of times the cell can be written to), volatility (volatile or non-volatile), and standby power. *Id.* at 3:19-30; 4:11-37. As a result, data may be distributed among the different memory portions or devices “in a manner designed to provide fast access to the data and programs that are most frequently used.” *Id.* at 3:64-67.

94. Ware also discloses a memory controller “configured to exploit the different attributes of the memory device types . . . so as to improve one or more aspects of memory system performance.” *Id.* at 4:47-52. The memory controller includes memory interfaces for coupling the controller to the two more different memory types. *Id.* at 7:22-25. The memory controller also comprises processor interface logic that “may include signal conditioning circuitry and other devices for transmitting data, addresses and control signals” to and from a host processor over a

bus. *Id.* at 7:46-49. The processor interface logic may also include decoding logic “to determine whether the first or second portion of main memory will receive a memory access transaction.” *Id.* at 7:49-54.

95. Annotated Figure 1, shown below, illustrates the non-homogenous memory system, comprising a **first memory portion** (DRAM), **second memory portion** (Flash), **memory controller** (memory controller 102) coupled to the memory portions via **communication pathways**, and a **bus** (bus 124) that couples the memory controller to a **host processor** (processor 108):



X. THE PRIOR ART DISCLOSES ALL OF THE CLAIMED ELEMENTS OF THE '190 PATENT

A. Each of the Challenged Claims Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge

96. It is my opinion that each of the challenged claims—claims 1, 5, 6, 29, 36, 37, 45, 51, 52, 59, 71, 76, 80, 84, 85, and 98—of the '190 patent is rendered obvious by Cornwell in view of a POSITA's knowledge.

1. Claim 1 Is Rendered Obvious by Cornwell in View of a

POSITA's Knowledge

97. As explained below, it is my opinion that claim 1 of the '190 patent is rendered obvious by Cornwell in view of a POSITA's knowledge.

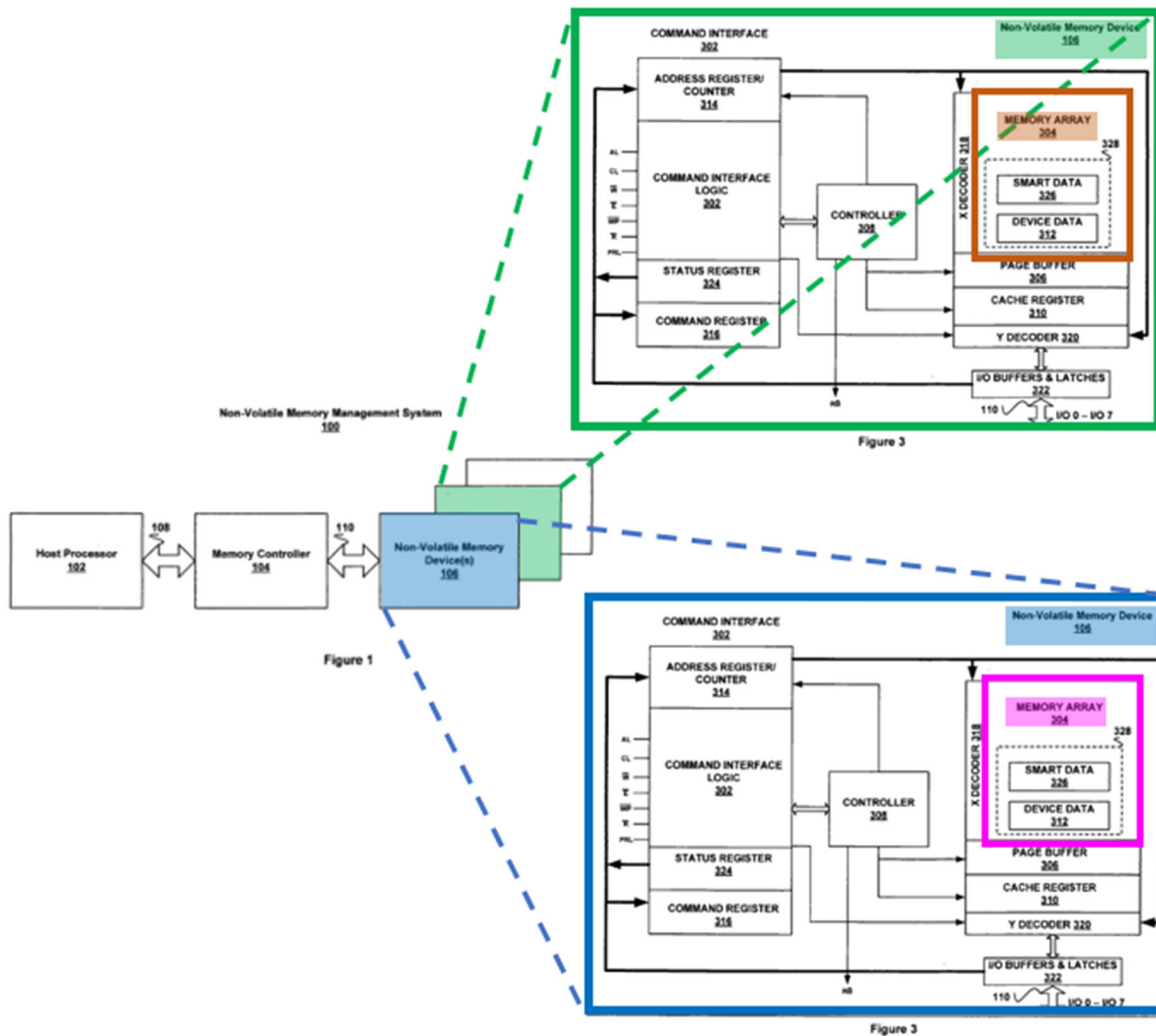
- a. **[Claim 1 Pre]: "A method of performing transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:"**

98. To the extent the preamble is limiting, it is my opinion that Cornwell discloses this claim element. Cornwell discloses "systems, methods and devices for managing non-volatile memory." Ex-1005, ¶ [0006]; *see also id.*, ¶ [0008] (discussing "a method of managing memory"); claim 12 (reciting "[a] method of managing memory"); claims 13-15 (same). In particular, Cornwell discloses memory transactions in the form of "read or write requests . . . [that] are mapped to physical memory addresses that can span two or more memory devices." *Id.*, ¶ [0024]; *see also id.*, ¶ [0038] (explaining that the memory controller carries out "read and write operations on memory array 304"); *id.*, ¶ [0040] (describing a page read operation in the memory); *id.*, ¶ [0046] (describing a memory management policy that "addresses how read and write operations should be performed"). In this manner, Cornwell discloses a method of performing transactions, i.e., read and write operations, on a set of memory locations.

99. Cornwell also discloses performing memory transactions on a set of

memory locations that includes a first memory location and a second memory location. For example, Cornwell discloses “one or more non-volatile memory devices 106,” which “can be discrete chips, chipsets and or memory modules.” *Id.*, ¶¶ [0018], [0020]. “Examples of non-volatile memory devices 106 include but are not limited to: NAND and/or NOR flash media,” as well as other well-known types of data storage devices. *Id.*, ¶ [0020].

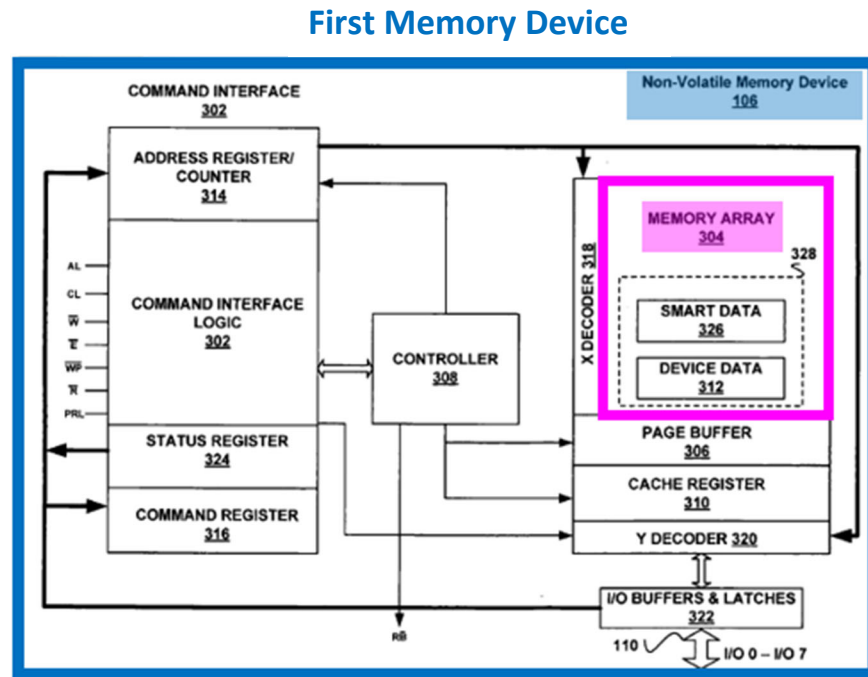
100. Annotated Figures 1 (showing the non-volatile memory management system) and 3 (detailing the non-volatile memory devices 106 in this system), combined below, illustrate the **first memory location** (memory array 304 in a **first memory device 106**) and a **second memory location** (memory array 304 in a **second memory device 106**).



101. Expanded views of annotated Figure 3,⁴ further illustrating a **first**

⁴ Because Figure 3 is representative of each of the “memory device(s) 106,” it has been duplicated below to illustrate the first memory device and second memory device. *See, e.g.*, Ex-1005 at ¶ [0018] (noting that the system may contain “**one or more** non-volatile memory devices 106”); *id.* at ¶ [0020] (describing aspects of

memory location (memory array 304 in a first memory device 106) and a second memory location (memory array 304 in a second memory device 106), are shown below:



“non-volatile memory *devices* 106”); *id.* at ¶ [0023] (same); Figure 1 (showing set of “Non-Volatile Memory *Device(s)* 106”) (all emphasis added).

Second Memory Device

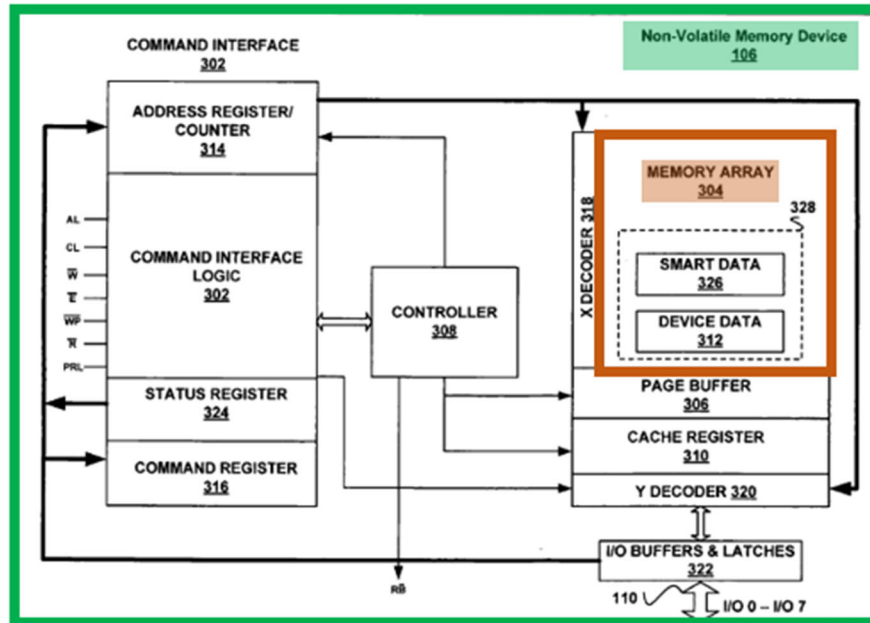


Figure 3

102. For at least these reasons, it is my opinion that Cornwell discloses “[a] method of performing transactions on a set of memory locations that includes a first memory location and a second memory location.”

b. [Claim 1a]: “performing a memory transaction comprising addressing a first memory location in a first memory device in a memory store,”

103. It is my opinion that Cornwell discloses this claim element. Cornwell discloses a memory store comprising “one or more non-volatile memory devices 106,” which “can be discrete chips, chipsets and/or memory modules.” Ex-1005, ¶¶ [0018], [0020]. The disclosed memory devices may “include but are not limited to: NAND and/or NOR flash media, read-only memory (ROM), erasable,

programmable ROM (EPROM), electrically-erasable, programmable ROM (EEPROM), Ferroelectric RAM (FeRAM), magnetoresistive RAM (MRAM), non-volatile static RAM (nvSRAM), and any other memory device that does not need its memory contents periodically refreshed and/or can retain memory without power.” *Id.*, ¶ [0020].

104. The **memory store**, comprising a plurality of memory devices 106, is illustrated in annotated Figure 1 below:

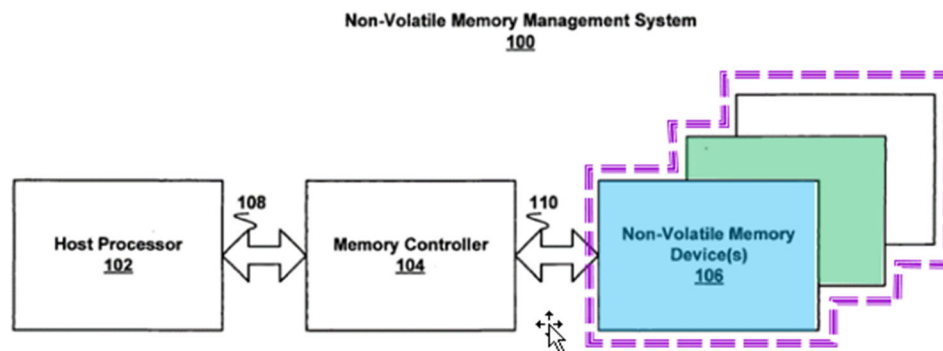
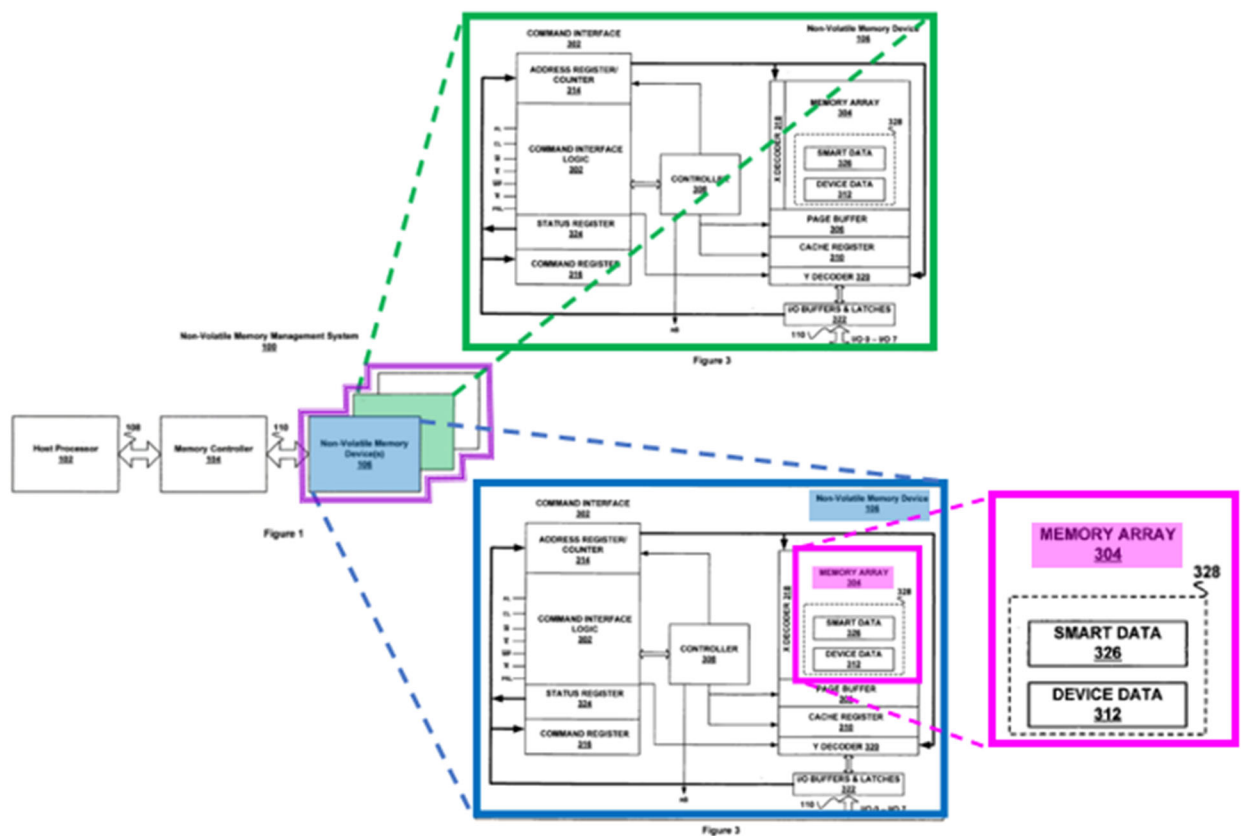


Figure 1

105. Cornwell also teaches performing a memory transaction comprising addressing a first memory location in a first memory device. Cornwell explains that “the memory controller 104 recognizes control, **address**, and data signals transmitted . . . by the host processor 102” and “translates the control, **address** and data **signals into** memory access **requests on memory devices 106.**” *Id.*, ¶ [0021] (emphasis added); *see also id.* (explaining that an IDE/ATA bus may translate “control, **address** and data signals into memory access requests”); *id.*, ¶ [0035]

(noting that the memory device has a decoder that “decodes input *addresses*”); *id.*, ¶ [0037] (“The address register/counter 314 is configured to receive *addresses* from the memory controller 104 via I/O bus 110.”) (all emphasis added). A POSITA would readily understand that an address for accessing data on a memory device would include information addressing the location in the memory device where that data is located.

106. Annotated Figures 1 and 3, below, illustrate the **first memory location** (memory array 304 in a **first memory device 106**) in the **memory store**:



107. For at least these reasons, it is my opinion that Cornwell discloses

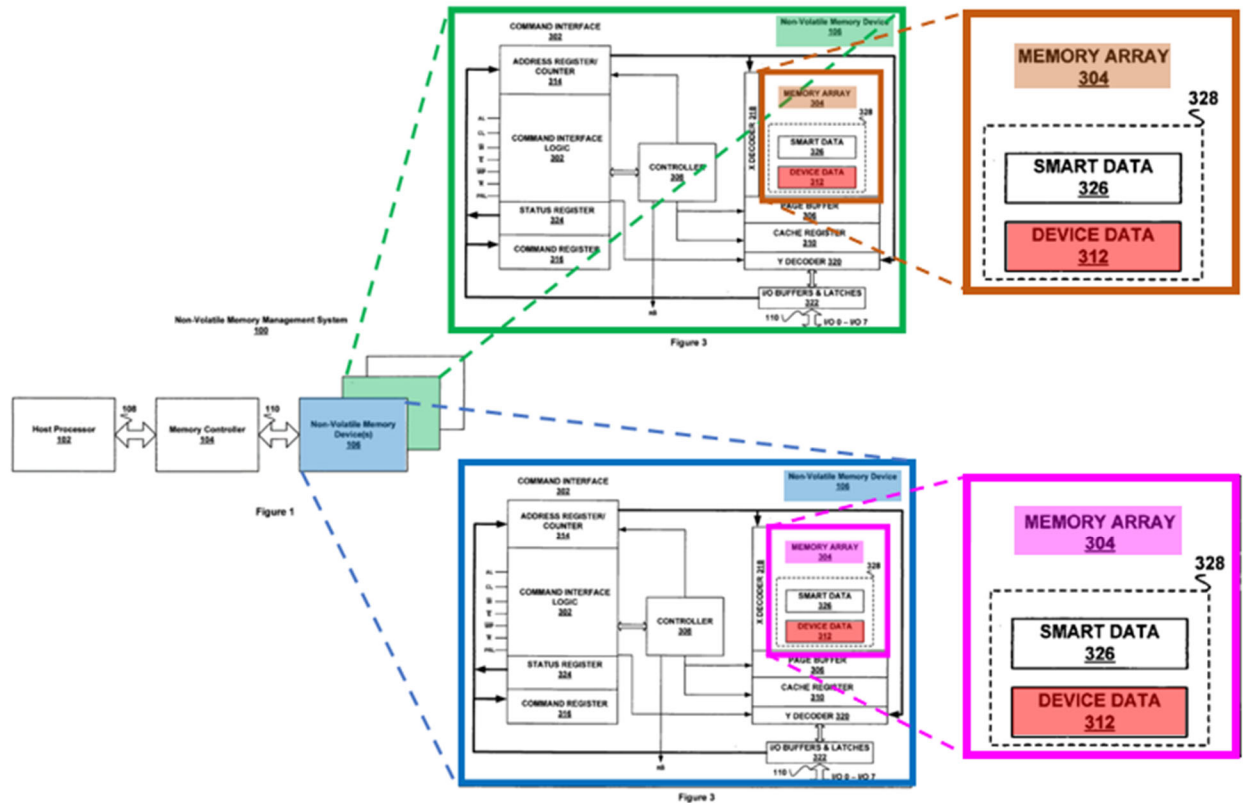
“performing a memory transaction comprising addressing a first memory location in a first memory device in a memory store.”

c. [Claim 1b]: “said first memory location and a second memory location respectively associated with a first device profile and a second device profile,”

108. It is my opinion that Cornwell discloses this claim element. Cornwell discloses that each memory device 106 may include “device data 312 [that] can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. The device data “can include an electronic signature or serial number that includes a manufacturer code and/or device code,” as well as information such “as device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.* Accordingly, Cornwell’s collective set of “device data” comprises a “device profile.”⁵

⁵ As the ’190 patent explains, a device profile is simply “a set of memory attributes.”

109. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of device data 312) associated with both the **first memory location** and the **second memory location**:



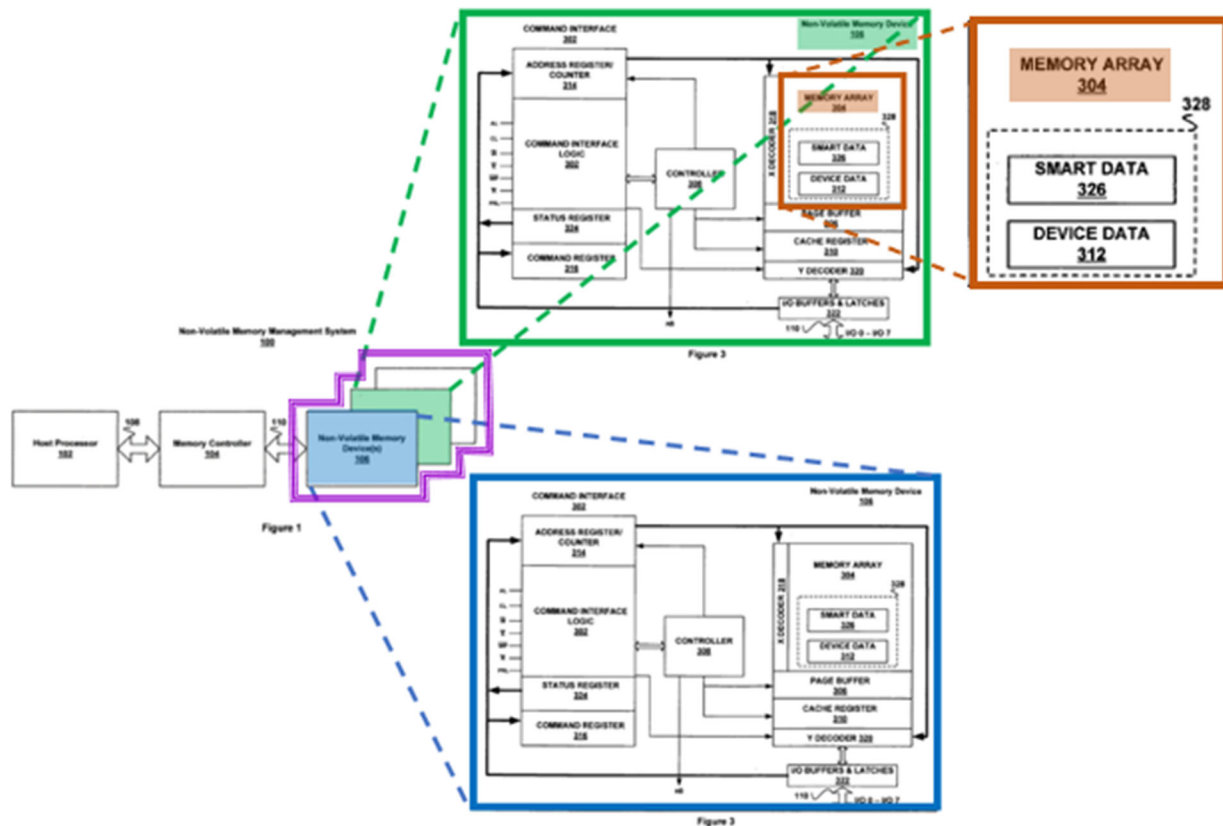
110. For at least these reasons, Cornwell discloses that “said first memory location and a second memory location respectively associated with a first device

Ex-1001 at 3:14-15. According to the '190 patent, a device profile may have “any number of attributes,” including page size, block size, memory protocol, memory device type, partial page size, etc. *Id.* at 3:40-61.

profile and a second device profile.”

d. [Claim 1c]: “wherein said second memory location is in a second memory device in the memory store,”

111. It is my opinion that Cornwell discloses this claim element. As explained above, the second memory location (i.e., memory array) is in the second memory device in the memory store. *See supra* §§ X.A.1.b, X.A.1.c. Annotated Figures 1 and 3, below, further illustrate that the **second memory location** is in the **second memory device** in the **memory store**:



112. For at least these reasons, it is my opinion that Cornwell discloses that

“said second memory location is in a second memory device in the memory store.”

- e. **[Claim 1d]: “wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location,”**

113. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that, “[d]uring operation, one or more memory devices 106 receive signals from the memory controller 104 over Input/Output (I/O) bus 110, which enables the *memory devices 106 to perform* memory access requests (e.g., *read or write operations*).” Ex-1005, ¶ [0024]; *see also id.* (explaining that “[i]n some implementations, the memory devices 106 are interleaved, so that *read or write requests* to logical block addresses (LBAs) are mapped to physical memory addresses that can span two or more memory devices 106”); ¶ [0031] (noting that an I/O interface “provides connectivity to the memory devices 106 through I/O bus 110, and includes circuitry for *enabling read, program, and erase operations to one or more memory devices 106*”); ¶ [0034] (noting that the I/O bus includes lines that are used to “(b) output data during a *read operation*, or (c) input a command or data during a *program operation*”); ¶ [0039] (describing a “page program” or write operation); ¶ [0040] (describing a “page read” operation) (all emphasis added).

114. For at least these reasons, it is my opinion that Cornwell teaches “wherein a memory read transaction or a memory write transaction is performed on

at least one of the first memory location or second memory location.”

- f. [Claim 1e]: “wherein data is transferred to the first memory location or second memory location from a host for a memory write transaction, wherein data is transferred from the first memory location or second memory location to the host for a memory read transaction;”

115. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that “[i]n some flash memory systems, *a host system performs reads and writes* to logical block addresses (LBAs) which are mapped or translated to physical block addresses of flash memory.” Ex-1005, ¶ [0004] (emphasis added); *see also id.*, ¶ [0007] (“A host processor is operatively coupled to the memory controller and configured to retrieve from the memory controller at least one attribute of at least one non-volatile memory device”). Likewise, “the memory controller 104 recognizes control, address, and data signals *transmitted* on bus 108 *by the host processor* 102,” which the controller translates “into memory access requests on memory devices 106.” *Id.*, ¶ [0021]; *see also id.*, ¶ [0029] (explaining that firmware in the controller “can translate signals from the host processor 102 into memory read and write operations”).

116. Annotated Figure 1, shown below, illustrates data being transferred by the **host processor (highlighted in yellow)** over **bus 108** and **bus 110** (I/O bus 110) to and from memory locations in the **first memory device** and the **second memory**

device:

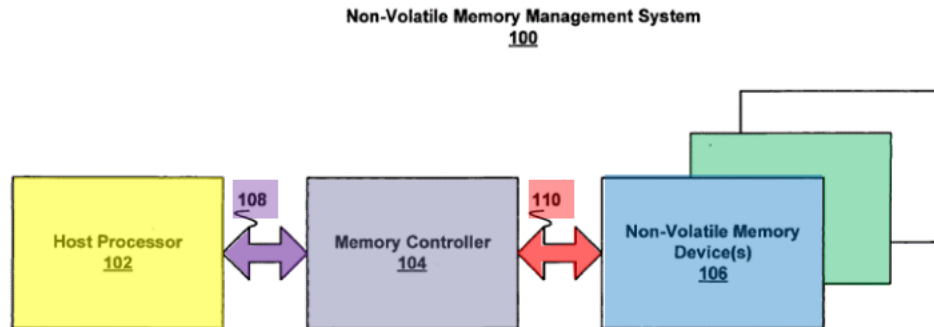
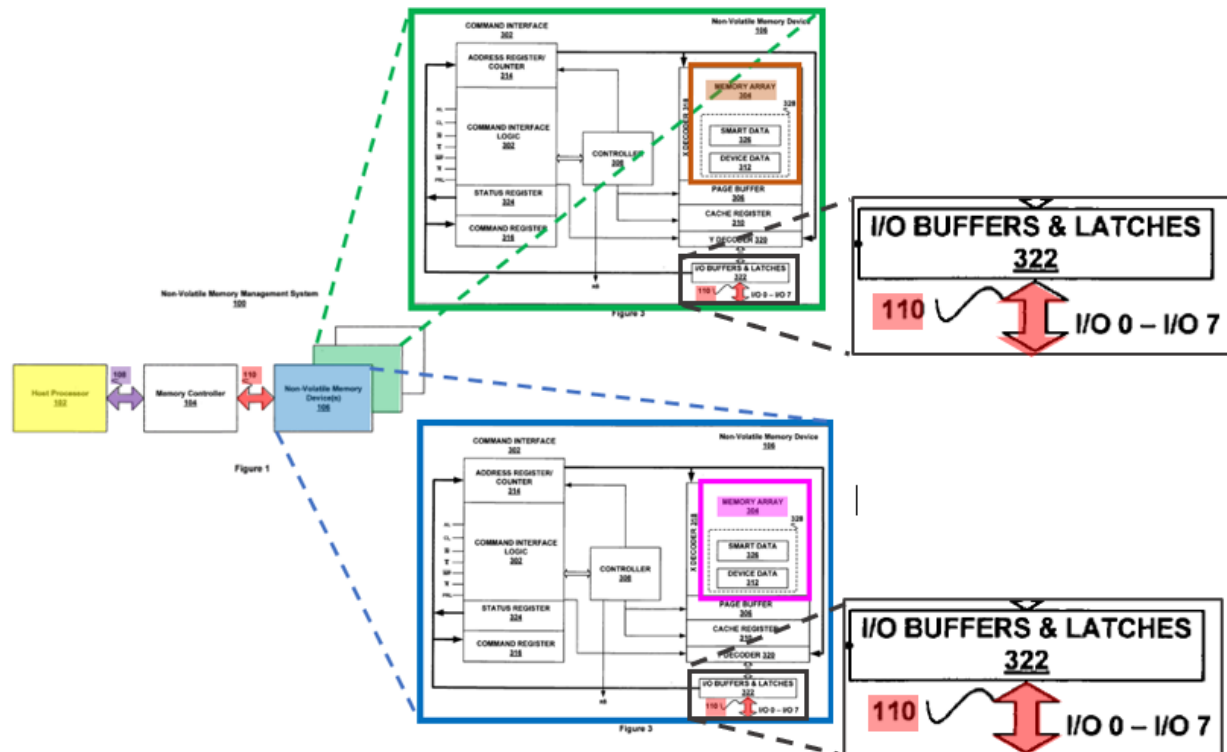


Figure 1

117. Annotated Figures 1 and 3, below, further illustrate data being transferred by the **host processor (highlighted in yellow)** over **bus 108** and **bus 110** (I/O bus 110) to and from memory locations in the **first memory device** and the **second memory device**:



118. For at least these reasons, it is my opinion that Cornwell teaches “wherein data is transferred to the first memory location or second memory location from a host for a memory write transaction, wherein data is transferred from the first memory location or second memory location to the host for a memory read transaction.”

- g. [Claim 1f]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”

119. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. As explained above, Cornwell teaches a “device

profile” in the form of a collection of “device data 312.” *See supra* § X.A.1.c. The device profile “can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. Information comprising the device profile may include “device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.*

120. A POSITA would have also understood that the device profiles taught in Cornwell are optimal for a data type subject to the memory transaction. For example, as explained in the preceding paragraph, Cornwell teaches that the device profile identifies whether the memory is NAND or NOR. *Id.* At the time Cornwell was filed, a POSITA would have readily understood that NOR provides random access (i.e., access to a “random data type”) and NAND provides sequential access (i.e., access to a sequential data type”).⁶ Accordingly, it would have been obvious

⁶ These are some of the most basic functionalities of NAND and NOR. For example, Kishida confirms that NAND is “a serially or sequentially outputting memory.” Ex-

to a POSITA, based on Cornwell alone, that a profile for NOR memory would have been optimal for a random data type subject to a memory transaction, and that a profile for NAND memory would have been optimal for a sequential data type subject to a memory transaction.

121. Similarly, Cornwell teaches “block defining” as another way of ensuring that a device profile is optimal for a data type subject to the memory transactions. *See id.*, ¶¶ [0048]-[0050]. For example, Cornwell teaches that “[m]emory access efficiency can be improved by *matching the average size of files* to be stored in the flash media *to the block size* of the flash media.” *Id.*, ¶ [0048]; *see also id.*, ¶ [0050] (explaining that “it may be advantageous to *tailor the size of files* such that the average file size is *proportional to the block size.*”) (all emphasis added). As noted above, block size is one of the device profile attributes taught by Cornwell.

122. Moreover, Cornwell teaches that the data type comprises one of a random data type or a sequential data type, explaining that “[i]n some

1010, ¶ [0037]; *see also id.*, Abstract (noting that NAND increments to a next, sequential memory location”); *id.*, ¶ [0032]. Conversely, NOR is used “for its random access storage capabilities.” *Id.*, ¶ [0023].

implementations a *random [data type] read command* may be issued first, followed by a page [*sequential data type*] read command.” *Id.*, ¶ [0040] (emphasis added). Thus, by matching block size to file size, and teaching that files contain random or sequential data types, Cornwell further teaches first device profile is optimal for a data type subject to the memory transaction.⁷

123. For at least these reasons, it is my opinion that Cornwell renders obvious “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- h. [Claim 1g]: “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

⁷ In its claim construction order, the Court stated that examples of optimal device profiles are “(1) a device profile that includes a block size attribute value that is the largest available from the other device profiles when the data type is sequential,” and “(2) a device profile that includes a block size attribute value that is the smallest available from the other device profiles[] when the data type is random.” Ex-1007, at 44.

124. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the memory device “generally includes a command interface 302, a memory array 304 and a controller 308.” Ex-1005, ¶ [0034]. Cornwell further identifies, for example, a variety of commands that are “translated into signals which can be used by a controller and decoding logic in the memory device 106 to access a memory array.” *Id.*, ¶ [0027]. “For example, when the host processor 102 makes a read request, the “Read Sectors” opcode (20 h) is transmitted to the memory controller 104, together with address and control signals for accessing the sector(s).” *Id.*; *see also id.*, ¶ [0035] (describing a decoder that “decodes signals from the command interface logic 302 for reading or writing data into the memory line”); ¶ [0038] (explaining that the controller receives “one or more input addresses and command inputs, which are used by the controller 308 in combination with the control signals from the command interface logic 302 to carry out read and write operations on memory array 304”). Likewise, Cornwell teaches an “‘identify’ command that returns a chip ID,” which can then be used “to look-up information about the memory devices 106.” *Id.*, ¶ [0077]. As a result, Cornwell discloses identifying command details for causing the memory transaction to be performed, wherein the command details comprise the first memory device.

125. Examples of such commands are provided in Table 1 of Cornwell,

reproduced below:

TABLE I	
<u>Examples of Standard ATA-6 Commands</u>	
Opcode	Command
10h	Recalibrate
20h	Read Sectors
30h	Write Sectors
40h	Read Verify
B0h	SMART
C8h	Read DMA
CAh	Write DMA
E0h	Standby Immediate
E2h	Standby
E7h	Flush Cache
ECh	Identify
EFh	Set Features

126. The **command interface logic 302**, which identifies command details for causing the memory transaction to be performed, is illustrated in annotated Figure 3 below:

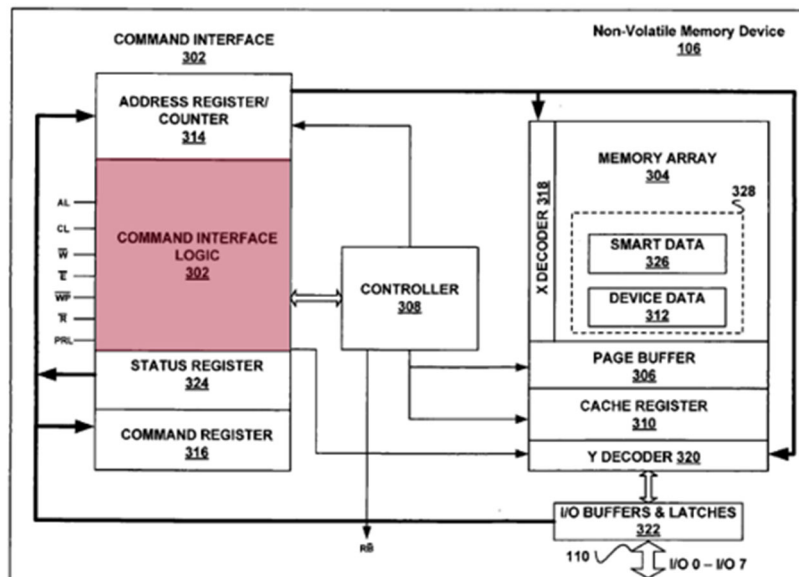


Figure 3

127. For at least these reasons, it is my opinion that Cornwell teaches “said

performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- i. **[Claim 1h]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

128. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. Cornwell teaches that “the device data 312,” which collectively comprises the device profile, “can include an electronic signature or serial number that includes a manufacturer code and/or device code.” Ex-1005, ¶ [0042]. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer.” *Id.* For example, device data 312 can include the following:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Potential
differences
between first and
second device

Id. (highlighting and underlining added). Cornwell expressly states that this information “is associated with *attributes*, properties or characteristics of the memory device 106 (*collectively, referred to herein as “attributes”*).” *Id.* (emphasis added).

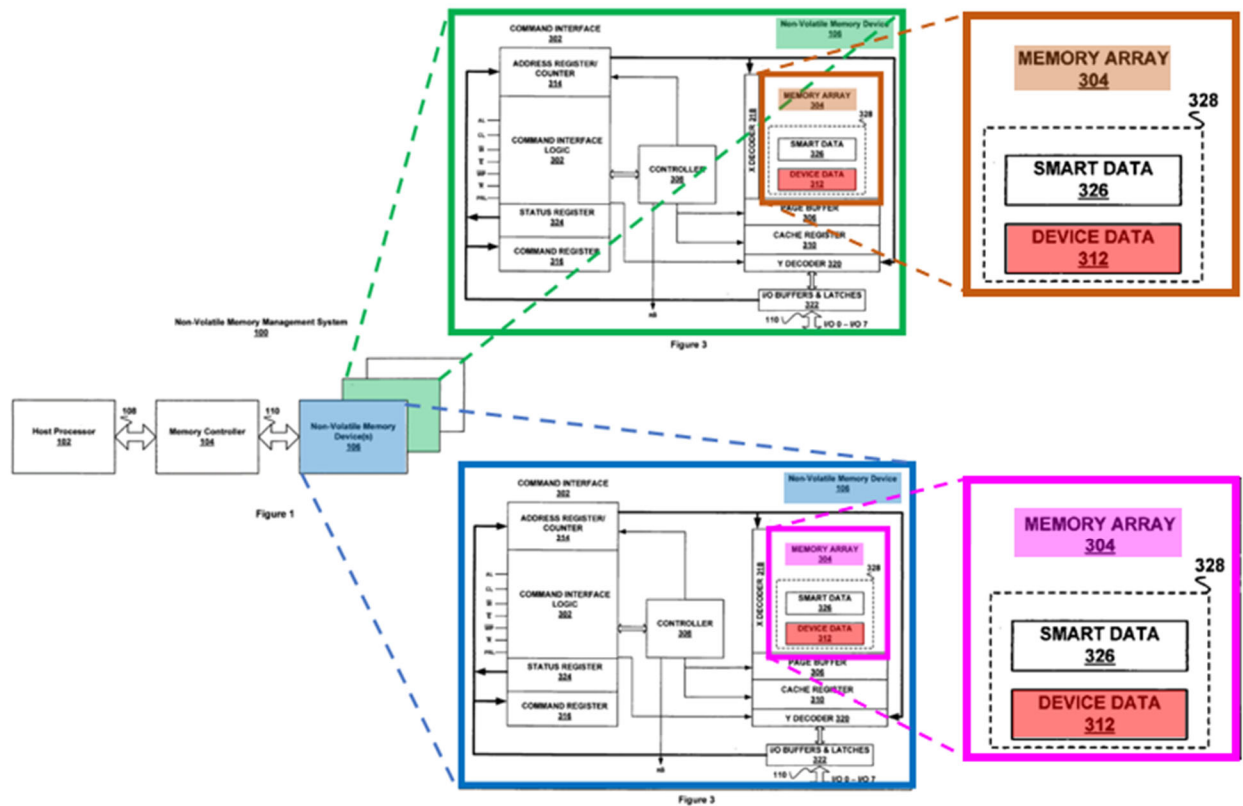
129. Cornwell also describes that “differences exist” between memory profiles. For example, Cornwell explains that different memory devices 106 may have different densities depending on whether they utilize single-layer cell technology or multi-layer cell technology, teaching that “syncing with memory devices 106 that include multi-level (MLC) technology can be performed at a different frequency than with memory devices 106 that include single-level cell technology (SLC).” *Id.*, [0078]. This is one example of a difference between profiles.

130. In my opinion, having a “difference exist” between memory profiles would also be obvious in view of Cornwell and a POSITA’s knowledge. As demonstrated above, Cornwell identifies a large number of potential differences in device attributes (describing differences in device type, device density, spare area, block size, etc.). *Id.*, ¶ [0042]. A difference in any of these attributes, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). As a result, I believe that Cornwell renders obvious having a “difference exist” between memory profiles.

131. Similarly, Cornwell explains that memory devices have certain levels of bad blocks, which may increase over the lifetime of the device and which are identified so that the host system can avoid writing to them. *Id.*, ¶¶ [0054]-[0056]; *see also id.*, ¶ [0048] (explaining that flash memory “is available in a variety of block sizes” and memory access efficiency can be improved by “matching the average size of the files to be stored in the flash media to the block size of the flash media”); ¶ [0060] (describing different interleave levels for chips). As noted above, bad block identification is yet another example of an attribute described by Cornwell. *Id.*, ¶ [0042]. Accordingly, a difference in the number of bad blocks, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). For this additional reason, I believe that Cornwell renders obvious having a “difference

exist” between memory profiles.

132. Annotated Figures 1 and 3, below, illustrate the **device profile** (i.e., the collection of attributes comprising device data 312) associated with both the **first memory location** and the **second memory location**:



133. For at least these reasons, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

j. [Claim 1i]: “after identifying the command details, obtaining the first set of attributes;”

134. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that attributes comprising device data 312 can be obtained or “transmitted to the memory controller 104 via the I/O bus 110 *in response to a read command* issued by the memory controller.” Ex-1005, ¶ [0043] (emphasis added). Cornwell’s teaching that the attributes are transmitted “in response to a command” confirms that the attributes are obtained after the command details are identified.

135. For at least these reasons, it is my opinion that Cornwell discloses “after identifying the command details, obtaining the first set of attributes.”

k. [Claim 1j]: “wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory location;”

136. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that each of the attributes discussed above—for example, device type, device density, etc.—is associated with a respective attribute qualifier. Ex-1005, ¶ [0042]; *see also supra* § X.A.1.i. For example, the “device type” attribute is associated with the qualifiers “NAND” or “NOR,” the “device density” attribute is associated with the qualifiers “512 Mb,” “1Gb,” and “2Gb,” etc. Ex-1005, ¶ [0042]. The qualifiers

associated with each attribute are illustrated in the following excerpt from Cornwell:

manufacturer code and/or device code. Chip data 312 can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

137. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer” and qualify the memory location. *Id.*; *see also id.* (noting that device data 312 “can include an electronic signature or serial number that includes a manufacturer code and/or device code”). For example, this device data can qualify a respective memory location either NAND or NOR, as having a page size of 1K or 2K, as having a spare area size of 8 or 16 bytes, etc. In this manner, the device data 312 qualifies each memory location.

138. For at least these reasons, it is my opinion that Cornwell teaches “wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory

location.”

- I. [Claim 1k]: “wherein the memory store is directly coupled to at least one memory bus and wherein the at least one memory bus is directly coupled to a controller, and wherein the controller performs the memory read transaction and memory write transaction on the memory store;”

139. It is my opinion that Cornwell discloses this claim element. Cornwell discloses that the memory store comprising “one or more memory devices 106 receive[s] signals *from the memory controller 104 over Input/Output (I/O) bus 110*, which enables the memory devices 106 to perform memory access requests (e.g., read or write operations).” Ex-1005, ¶ [0024]; *see also id.*, ¶ [0025] (“The memory controller 104 translates the request into signals for transmission on the I/O bus 110” to the memory device 106, which “receives the signals from the I/O bus 110 and performs the requested operation.”)

140. Annotated Figure 1, below, illustrates that the **memory store**—which may comprise a plurality of memory devices—is directly coupled to **bus 110**, which is also directly coupled to the **memory controller 104**:

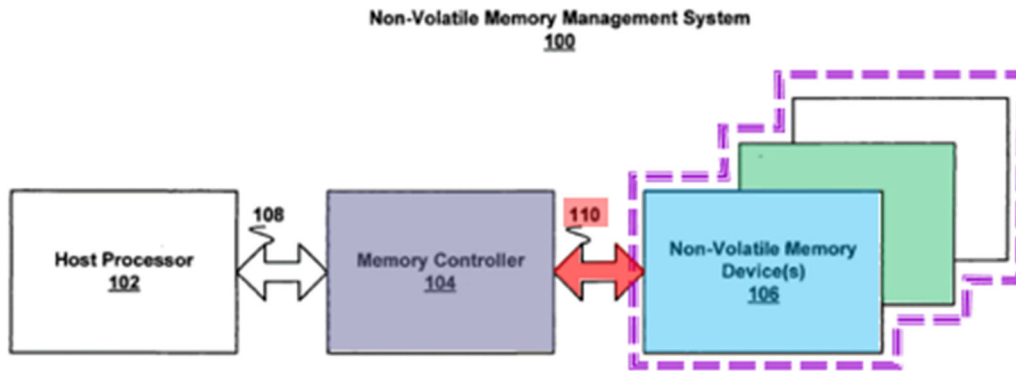


Figure 1

141. Cornwell also teaches that the controller performs read and write transactions, explaining that the memory devices “receive signals from the memory controller 104,” which “enables the memory devices 106 to perform memory access requests (e.g., read or write operations).” *Id.*, ¶ [0024]; *see also id.*, ¶ [0029] (noting that micro-controller unit firmware in the controller “translate[s] signals from the host processor 102 into memory read and write operations”).

142. For at least these reasons, it is my opinion that Cornwell discloses “wherein the memory store is directly coupled to at least one memory bus and wherein the at least one memory bus is directly coupled to a controller, and wherein the controller performs the memory read transaction and memory write transaction on the memory store.”

- m. **[Claim 11]: “wherein the host is directly coupled to the controller by a communication path as the controller performs the memory read transaction or memory write transaction;”**

143. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that data requests “are transmitted to the memory controller 104 on the IDE/ATA **bus 108** by the host processor 102.” Ex-1005, ¶ [0025] (emphasis added); *see also id.*, ¶ [0021] (noting that “the memory controller 104 recognizes control, address, and data signals transmitted on bus 108 by the host processor 102”).

144. Annotated Figure 1, shown below, further illustrates that the **host processor** (highlighted in yellow) is directly coupled to the **memory controller 104** via **bus 108**:

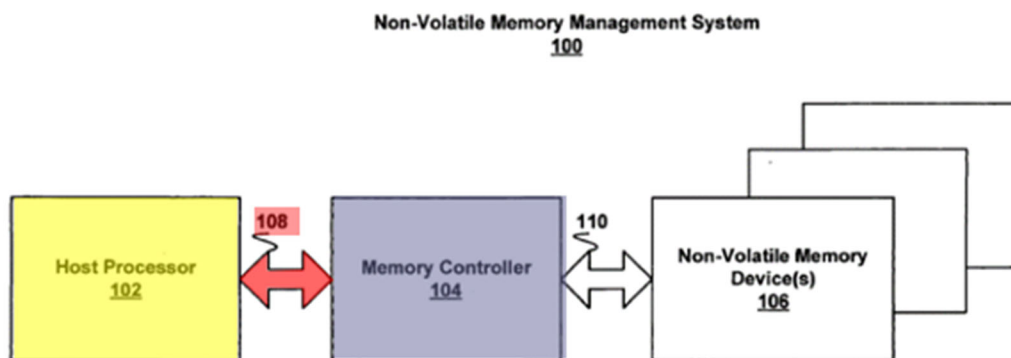


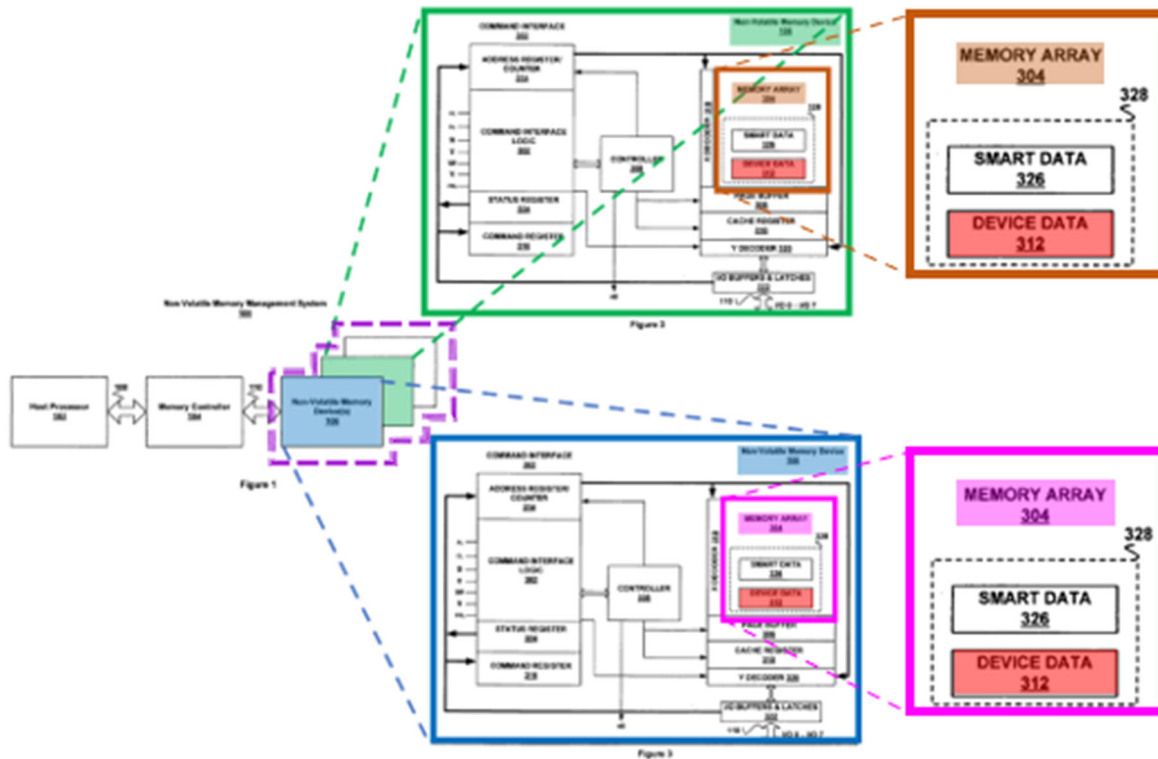
Figure 1

145. For at least these reasons, it is my opinion that Cornwell discloses “wherein the host is directly coupled to the controller by a communication path as the controller performs the memory read transaction or memory write transaction.”

- n. [Claim 1m]: “wherein the first device profile and second device profile are each stored in the memory store;”

146. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the set of device data 312, which collectively comprises the device profile, may be “stored in a spare area 328 of the memory array 304” and “can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. As previously explained, this data “can include an electronic signature or serial number that includes a manufacturer code and/or device code,” as well as “as device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.*; see also *supra* §§ X.A.1.c, X.A.1.i.

147. As illustrated in annotated Figures 1 and 3 below, **device profiles** for both the **first memory location** and the **second memory location** are stored in the **memory store**:



148. For at least these reasons, it is my opinion that Cornwell discloses “wherein the first device profile and second device profile are each stored in the memory store.”

- o. **[Claim 1n]: “and using attributes from said first and second device profiles; and”**

149. It is my opinion that Cornwell discloses this claim element. As explained above, Cornwell teaches that the first and second device profiles (i.e., collectively comprising device data 312) include a variety of attributes. *See supra* § X.A.1.i; Ex-1005, ¶ [0042] (describing attributes). Cornwell also teaches using the

attributes from the first and second device profiles, explaining that “device data 312 can be transmitted to the memory controller 104 . . . *in response to a read command* issued by the by the memory controller” and “*used by the memory controller 104 and/or host system* to perform various memory management tasks.” Ex-1005, ¶ [0043] (emphasis added); *see also id.*, ¶ [0042] (explaining that “device data 312 can be used to identify the memory device 106 and its manufacturer”); *id.*, ¶ [0046] (describing a memory management policy that “addresses how read and write operations should be performed”); *id.*, ¶ [0058] (explaining that the host processor “can request and receive memory device information (e.g., signatures, block size, interleave date, etc.) for one or more memory devices 106 over a standard IDE/ATA bus”).

150. For at least these reasons, it is my opinion that Cornwell discloses “using attributes from said first and second device profiles.”

- p. **[Claim 1o]: “selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

151. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches a memory transaction in which “the controller 308 *programs the page* into the memory array 304” Ex-1005, ¶ [0039] (emphasis added). As Cornwell further explains, a page has a specific “page size (1 k, 2K,

etc.).” *Id.*, ¶ [0042]; *see also, e.g., id.*, ¶ [0048] (“Memory access efficiency can be improved by ***matching the average size of files*** to be stored in the flash media ***to the block size of the flash media.***”); *id.*, ¶ [0050] “[I]t may be advantageous to ***tailor the size of files*** such that the average file size is ***proportional to the block size.***”) (all emphasis added). In this manner, Cornwell teaches selecting a transfer size for the memory transaction.

152. Cornwell further discloses that the page size is a function of the device density of the memory transactions. *Id.*, ¶ [0042]. For example, Cornwell teaches varying “device density (e.g., 512 Mb, 1Gb, 2Gb, etc.).” *Id.*; *see also supra* §§ X.A.1.i-X.A.1.k. A POSITA would have understood that page size is the length of a row (number of columns) in the memory. A POSITA would have further understood that the length of a row, again the number of columns in a row, varies with the device density since the device density is the number of rows multiplied by the number of columns in a memory array multiplied by the number of arrays used in the memory chip.⁸ For example, a page size of 2k (there are 2,048 columns in a row), with 8k rows and 1k arrays would result in device density of 16Gb.

⁸ *See, e.g.,* Ex-1011 at 6-7 (discussing memory array size and how addressing is used for setting page size).

153. Cornwell also demonstrates that the transfer size is selected by the controller. As seen in Figure 1 of Cornwell, the host 102 communicates via bus 108 with the memory controller 104 in a format consistent with the IDE/ATA bus standard and its command, including the data size information (e.g., the number of logical sectors to be transferred), while the memory controller 104 communicates with the memory store via the I/O bus 110, using the signals required by the chips used in the memory store and a transfer size, which is selected based on the data size information. As a result, the memory controller selects the transfer size for the memory transaction in response to a request from a host.

154. For instance, Cornwell teaches an example in which a media player sends a request to an operating system on a host. *Id.*, ¶ [0025]. This “request is received by the operating system, which formats the request into IDE/ATA signals, which are transmitted to the memory controller 104 on the IDE/ATA bus 108 by the host processor 102.” *Id.*⁹ The fact that host sends the requests to the memory controller—which must then communicate with the memory store—ensures that the

⁹ As Cornwell also explains, “ATA-6 is the latest version of the IDE/ATA standard, which was approved by the American National Standards Institute (ANSI) in 2001 under document NCITS 347-2001.” *Id.*, ¶ [0026].

memory controller selects the transfer size for the memory transactions, where the transfer size is a function of the data size of the memory transaction and the first set of attributes to retrieve the requested media file.

155. For at least these reasons, it is my opinion that Cornwell discloses “selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

2. Claim 5 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

156. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 5 of the ’190 patent.

a. [Claim 5]: “The method of claim 1, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

157. It is my opinion that Cornwell renders obvious the method of claim 1 for the reasons discussed above. *Supra* § X.A.1. It is also my opinion that Cornwell renders obvious the method recited in claim 5.

158. As explained above, Cornwell teaches that the first and second device profiles (i.e., each set of device data 312) have shared attributes. *Supra* § X.A.1.k; *see also supra* § X.A.1.i; Ex-1005, ¶ [0042]. For example, each device profile may have a “device type” attribute, “device density” attribute, “page size” attribute, etc.

Supra § X.A.1.k; Ex-1005, ¶ [0042]. Further, as explained above, it is my opinion that having different attribute qualifiers for these shared attributes would have been obvious in view of Cornwell and a POSITA's knowledge. *Supra* § X.A.1.i. For example, the first memory location could have a "NAND" qualifier for the "device type" attribute, while the second memory location could have "NOR" qualifier for the "device type" attribute. Ex-1005, ¶ [0042]. Likewise, the first memory location could have a "512 Mb" qualifier for the "device density" attribute, while the second memory location could have a "1Gb" qualifier for the "device density" attribute. *Id.* Cornwell discloses a variety of different attribute qualifiers for each shared attribute in the first and second device profiles, as illustrated below:

manufacturer code and/or device code. Chip data 312 can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as "attributes").

Id. (highlighting and underlining added).

159. For at least these reasons, it is my opinion that Cornwell, in view of the

knowledge of a POSITA, renders obvious the “method of claim 1, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

3. Claim 6 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

160. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 6 of the ’190 patent.

a. [Claim 6]: “The method of claim 5, wherein said different attribute qualifiers include different attribute values.”

161. It is my opinion that Cornwell renders obvious the method of claim 5 for the reasons discussed above. *Supra* § X.A.2. It is also my opinion that Cornwell renders obvious the method recited in claim 6.

162. As explained above, it is my opinion that having different attribute qualifiers would have been rendered obvious by Cornwell in view of a POSITA’s knowledge. *Supra* § X.A.1.i. Cornwell also teaches that each attribute qualifier has an attribute value, which may be different. For example, the “device density” attribute can have different values such as 512 Mb, 1 Gb, 2 Gb, etc. Ex-1005, ¶ [0042]. Likewise, the qualifiers associated with the “page size” attribute can have different values such as 1k, 2K, etc. *Id.*; *see also supra* §§ X.A.1.i, X.A.2.a. Accordingly, it is my opinion that it would have been obvious in view of Cornwell

and a POSITA's knowledge to have different attribute values for attribute qualifiers.

163. A variety of potential differences in attribute values are illustrated in the following excerpt from Cornwell:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

164. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “method of claim 5, wherein said different attribute qualifiers include different attribute values.”

4. Claim 29 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge

165. As explained below, it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious Claim 29 of the '190 patent.

- a. **[Claim 29 pre]: “A non-transitory computer readable medium comprising computer executable instructions adapted to cause a method of performing memory**

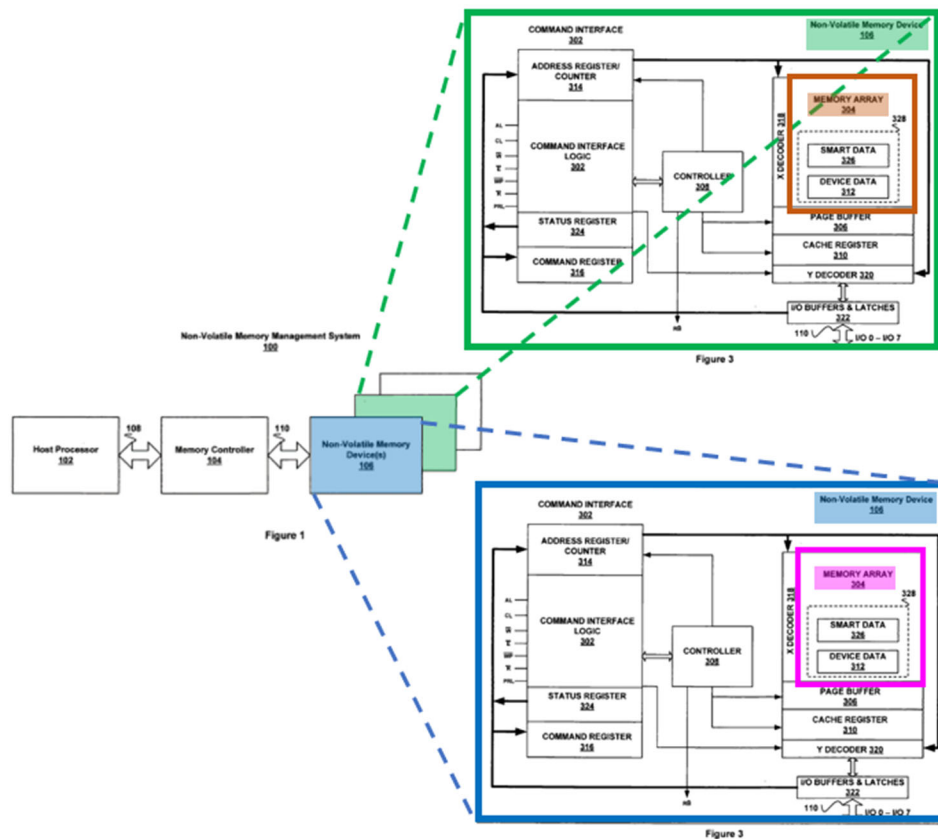
transactions on a first memory location and a second memory location, said method comprising:”

166. To the extent the preamble is limiting, it is my opinion that Cornwell discloses this claim element. Cornwell teaches a method for performing memory transactions on a first memory location and a second memory, describing “systems, methods and devices for managing non-volatile memory.” Ex-1005, ¶ [0006]; *see also id.*, ¶ [0008] (discussing “a method of managing memory”); claim 12 (reciting “[a] method of managing memory”); claims 13-15 (same). In particular, Cornwell discloses memory transactions in the form of “read or write requests . . . [that] are mapped to physical memory addresses that can span two or more memory devices.” *Id.*, ¶ [0024]; *see also id.*, ¶ [0038] (explaining that the memory controller carries out “read and write operations on memory array 304”); *id.*, ¶ [0040] (describing a page read operation in the memory); *id.*, ¶ [0046] (describing a memory management policy that “addresses how read and write operations should be performed”). In this manner, Cornwell discloses a method of performing transactions, i.e., read and write operations, on a set of memory locations.

167. Cornwell also teaches performing memory transactions on a set of memory locations that includes a first memory location and a second memory location. For example, Cornwell discloses “one or more non-volatile memory devices 106,” which “can be discrete chips, chipsets and or memory modules.” *Id.*,

¶¶ [0018], [0020]. “Examples of non-volatile memory devices 106 include but are not limited to: NAND and/or NOR flash media,” as well as other well-known types of data storage devices. *Id.*, ¶ [0020].

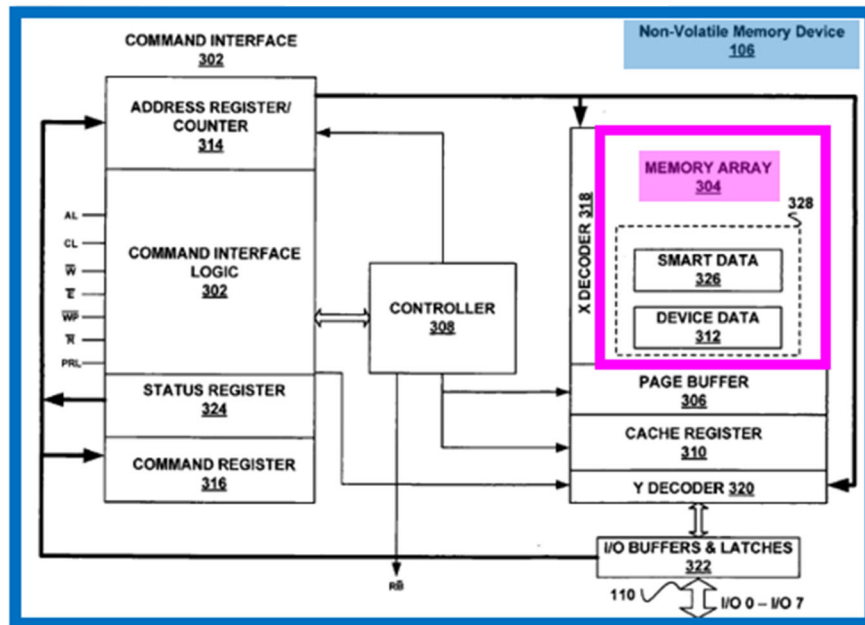
168. Annotated Figures 1 (showing the non-volatile memory management system) and 3 (detailing the non-volatile memory devices 106 in this system), combined below, illustrate the **first memory location** (memory array 304 in a **first memory device 106**) and a **second memory location** (memory array 304 in a **second memory device 106**).



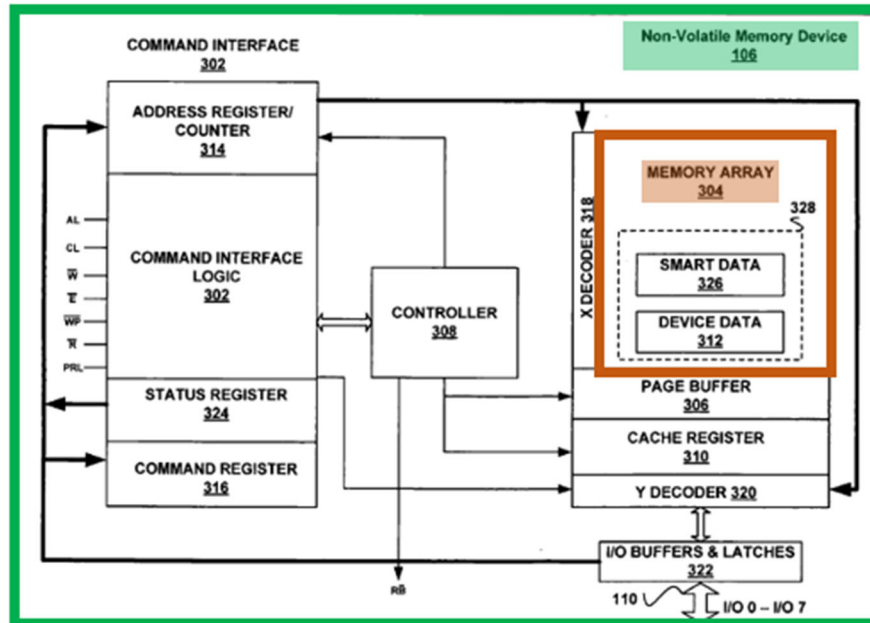
169. Expanded views or annotated Figure 3,¹⁰ further illustrating a **first memory location** (memory array 304 in a **first memory device 106**) and a **second memory location** (memory array 304 in a **second memory device 106**) are shown below:

¹⁰ Because Figure 3 is representative of each of the “memory device(s) 106,” it has been duplicated below to illustrate the first memory device and second memory device. *See, e.g.*, Ex-1005, ¶ [0018] (noting that the system may contain “***one or more*** non-volatile memory devices 106”); *id.*, ¶ [0020] (describing aspects of “non-volatile memory ***devices*** 106”); *id.*, ¶ [0023] (same); Figure 1 (showing set of “Non-Volatile Memory ***Device(s)*** 106”) (all emphasis added).

First Memory Device



Second Memory Device



170. In addition, Cornwell discloses a non-transitory computer readable medium comprising computer executable instructions adapted to cause this method

to be performed. Cornwell teaches that, in some “implementations, the memory device information is stored on a **computer-readable medium** in the host system (e.g., memory, hard disk, CDROM, etc.),” which are non-transitory computer readable media. *See id.*, ¶ [0047] (emphasis added). Likewise, Cornwell discloses that “the host system can include pre-stored information for multiple memory devices that are known to be compatible with the host system and the memory controller.” *Id.* As such, the non-transitory computer readable medium comprises computer executable instructions adapted to cause a method of performing memory transactions on a first memory location and a second memory location. *See also id.*, ¶ [0083] (noting that “[t]he term ‘computer-readable medium’ refers to any medium that participates in providing instructions to a processor for execution, including without limitation non-volatile media (e.g., optical or magnetic disks)”; ¶ [0084].

171. For at least these reasons, it is my opinion that Cornwell teaches a “non-transitory computer readable medium comprising computer executable instructions adapted to cause a method of performing memory transactions on a first memory location and a second memory location.”

- b. **[Claim 29a]: “performing a memory transaction comprising addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile;”**

172. It is my opinion that Cornwell teaches performing a memory transaction comprising addressing a first memory location. For example, Cornwell discloses a memory store comprising “one or more non-volatile memory devices 106,” which “can be discrete chips, chipsets and or memory modules.” Ex-1005, ¶¶ [0018], [0020]. The disclosed memory devices may “include but are not limited to: NAND and/or NOR flash media, read-only memory (ROM), erasable, programmable ROM (EPROM), electrically-erasable, programmable ROM (EEPROM), Ferroelectric RAM (FeRAM), magnetoresistive RAM (MRAM), non-volatile static RAM (nvSRAM), and any other memory device that does not need its memory contents periodically refreshed and/or can retain memory without power.” *Id.*, ¶ [0020].

173. The **memory store**, comprising a plurality of memory devices 106, is illustrated in annotated Figure 1 below:

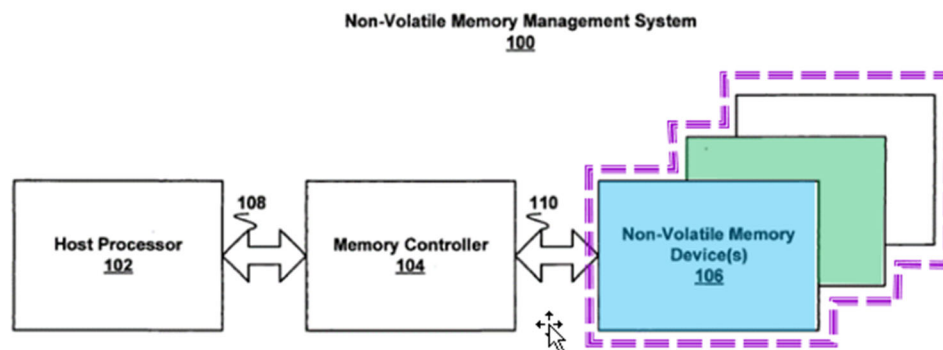
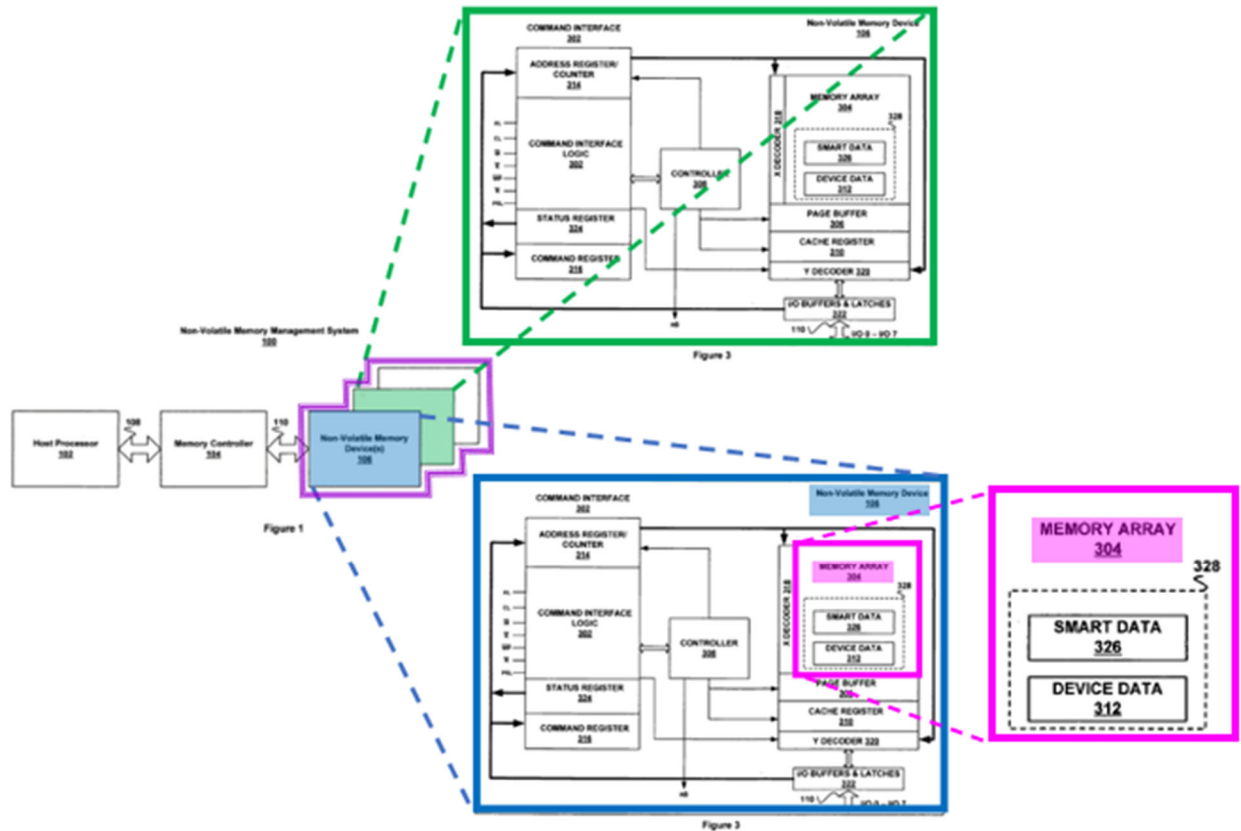


Figure 1

174. Cornwell also teaches performing a memory transaction comprising

addressing a first memory location in a first memory device. Cornwell explains that “the memory controller 104 recognizes control, **address**, and data signals transmitted . . . by the host processor 102” and “translates the control, **address** and data **signals into memory access requests** on memory devices 106.” *Id.*, ¶ [0021] (emphasis added); *see also id.* (explaining that an IDE/ATA bus may translate “control, **address** and data signals into memory access requests”); *id.*, ¶ [0035] (noting that the memory device has a decoder that “decodes input **addresses**”); *id.*, ¶ [0037] (“The address register/counter 314 is configured to receive **addresses** from the memory controller 104 via I/O bus 110.”) (all emphasis added). A POSITA would readily understand that an address for accessing data on a memory device would include information addressing the location in the memory device where that data is located.

175. Annotated Figures 1 and 3, below, illustrate the **first memory location** (memory array 304 in a **first memory device 106**) in the **memory store**:

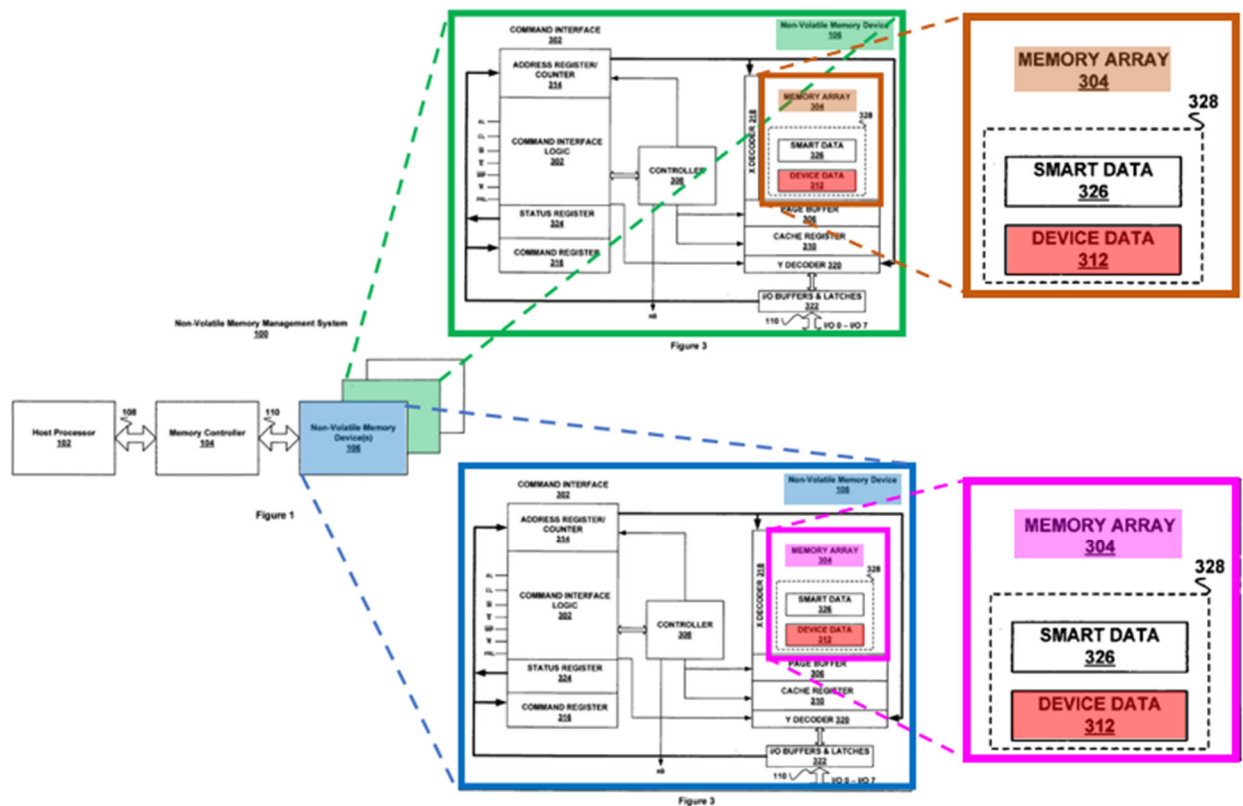


176. Cornwell also teaches that the first memory location and a second memory location are respectively associated with a first device profile and a second device profile. For example, Cornwell discloses that each memory device 106 may include “device data 312 [that] can be used to identify the memory device 106 and its manufacturer.” *Id.*, ¶ [0042]. The device data “can include an electronic signature or serial number that includes a manufacturer code and/or device code,” as well as information such “as device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g.,

30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., $\times 8$, $\times 16$, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.* Accordingly, Cornwell’s collective set of “device data” comprises a “device profile.”¹¹

177. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of device data 312) associated with both the **first memory location** and the **second memory location**:

¹¹ See *supra* note 5 (explaining characteristics of device profiles).



178. For at least these reasons, it is my opinion that Cornwell discloses “performing a memory transaction comprising addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile.”

- c. [Claim 29b]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”

179. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. As explained above, Cornwell teaches a “device

profile” in the form of a collection of “device data 312.” *See supra* § X.A.4.b. The device profile “can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. Information comprising the device profile may include “device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.*

180. A POSITA would have also understood that the device profiles taught in Cornwell are optimal for a data type subject to the memory transaction. For example, as explained in the preceding paragraph, Cornwell teaches that the device profile identifies whether the memory is NAND or NOR. *Id.* At the time Cornwell was filed, a POSITA would have readily understood that NOR provides random access (i.e., access to a “random data type”) and NAND provides sequential access (i.e., access to a sequential data type”).¹² Accordingly, it would have been obvious to a POSITA, based on Cornwell alone, that a profile for NOR memory would have

¹² *See supra* note 6.

been optimal for a random data type subject to a memory transaction, and that a profile for NAND memory would have been optimal for a sequential data type subject to a memory transaction.

181. Similarly, Cornwell teaches “block defining” as another way of ensuring that a device profile is optimal for a data type subject to the memory transactions. *See id.*, ¶¶ [0048]-[0050]. For example, Cornwell teaches that “[m]emory access efficiency can be improved by matching the average size of files to be stored in the flash media *to the block size of the flash media.*” *Id.*, ¶ [0048]; *see also id.*, ¶ [0050] (explaining that “it may be advantageous to *tailor the size of files* such that the *average file size is proportional to the block size.*”) (all emphasis added). As noted above, block size is one of the device profile attributes taught by Cornwell. The district court’s claim construction order states that such block size comparison information is another example of a device profile that is optimal for a data type subject to the memory transaction.¹³

¹³ As noted above, the district court’s claim construction order states that examples of optimal device profiles are “(1) a device profile that includes a block size attribute value that is the largest available from the other device profiles when the data type is sequential,” and “(2) a device profile that includes a block size attribute value that

182. Moreover, Cornwell teaches that the data type comprises one of a random data type or a sequential data type, explaining that “[i]n some implementations a ***random [data type] read command*** may be issued first, followed by a page [***sequential data type***] read command.” *Id.*, ¶ [0040] (emphasis added). Thus, by matching block size to file size, and teaching that files contain random or sequential data types, Cornwell further teaches first device profile is optimal for a data type subject to the memory transaction.

183. For at least these reasons, it is my opinion that Cornwell in view of a POSITA’s knowledge renders obvious “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.

184. For at least these reasons, it is my opinion that Cornwell discloses “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.

- d. **[Claim 29c]: “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed,**

is the smallest available from the other device profiles” when the data type is random.” *Supra* note 7 (citing Ex-1007 at 44).

wherein said command details comprising the first memory device;”

185. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the memory device “generally includes a command interface 302, a memory array 304 and a controller 308.” Ex-1005, ¶ [0034]. Cornwell further identifies a variety of commands that are “translated into signals which can be used by a controller and decoding logic in the memory device 106 to access a memory array.” *Id.*, ¶ [0027]. “For example, when the host processor 102 makes a read request, the “Read Sectors” opcode (20 h) is transmitted to the memory controller 104, together with address and control signals for accessing the sector(s).” *Id.*; *see also id.*, ¶ [0035] (describing a decoder that “decodes signals from the command interface logic 302 for reading or writing data into the memory line”); ¶ [0038] (explaining that the controller receives “one or more input addresses and command inputs, which are used by the controller 308 in combination with the control signals from the command interface logic 302 to carry out read and write operations on memory array 304”). Likewise, Cornwell teaches an “‘identify’ command that returns a chip ID,” which can then be used “to look-up information about the memory devices 106.” *Id.*, ¶ [0077]. As a result, Cornwell discloses identifying command details for causing the memory transaction to be performed, wherein the command details comprise the first memory device.

186. Examples of such commands are provided in Table 1 of Cornwell, reproduced below:

TABLE I

<u>Examples of Standard ATA-6 Commands</u>	
Opcode	Command
10h	Recalibrate
20h	Read Sectors
30h	Write Sectors
40h	Read Verify
B0h	SMART
C8h	Read DMA
CAh	Write DMA
E0h	Standby Immediate
E2h	Standby
E7h	Flush Cache
EC	Identify
EFh	Set Features

187. The **command interface logic 302**, which identifies command details for causing the memory transaction to be performed, is also illustrated in annotated Figure 3 below:

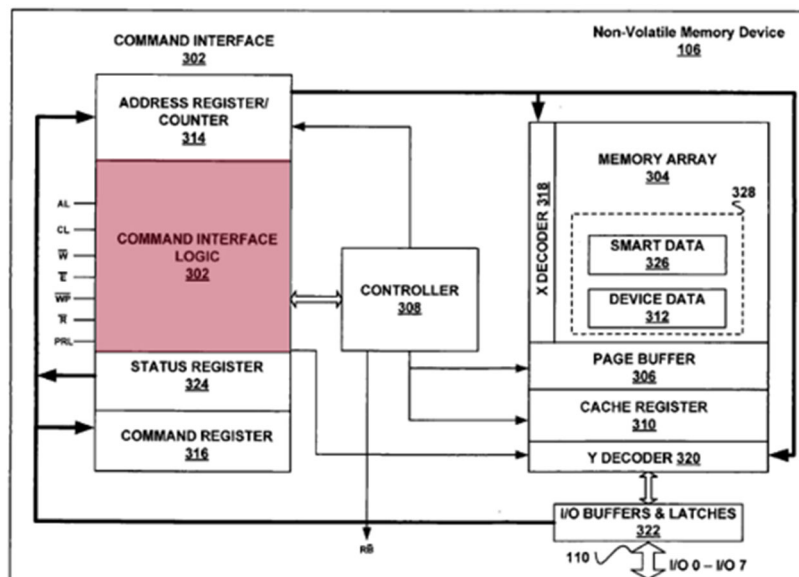


Figure 3

188. For at least these reasons, it is my opinion that Cornwell teaches “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- e. **[Claim 29d]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

189. It is my opinion that Cornwell renders this claim obvious in view of a POSITA’s knowledge. Cornwell teaches that “the device data 312,” which collectively comprises the device profile, “can include an electronic signature or serial number that includes a manufacturer code and/or device code.” Ex-1005, ¶ [0042]. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer.” *Id.* For example, device data 312 can include the following:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Potential
differences
between first and
second device

Id. (highlighting and underlining added). Cornwell expressly states that this information “is associated with *attributes*, properties or characteristics of the memory device 106 (*collectively, referred to herein as “attributes”*).” *Id.*

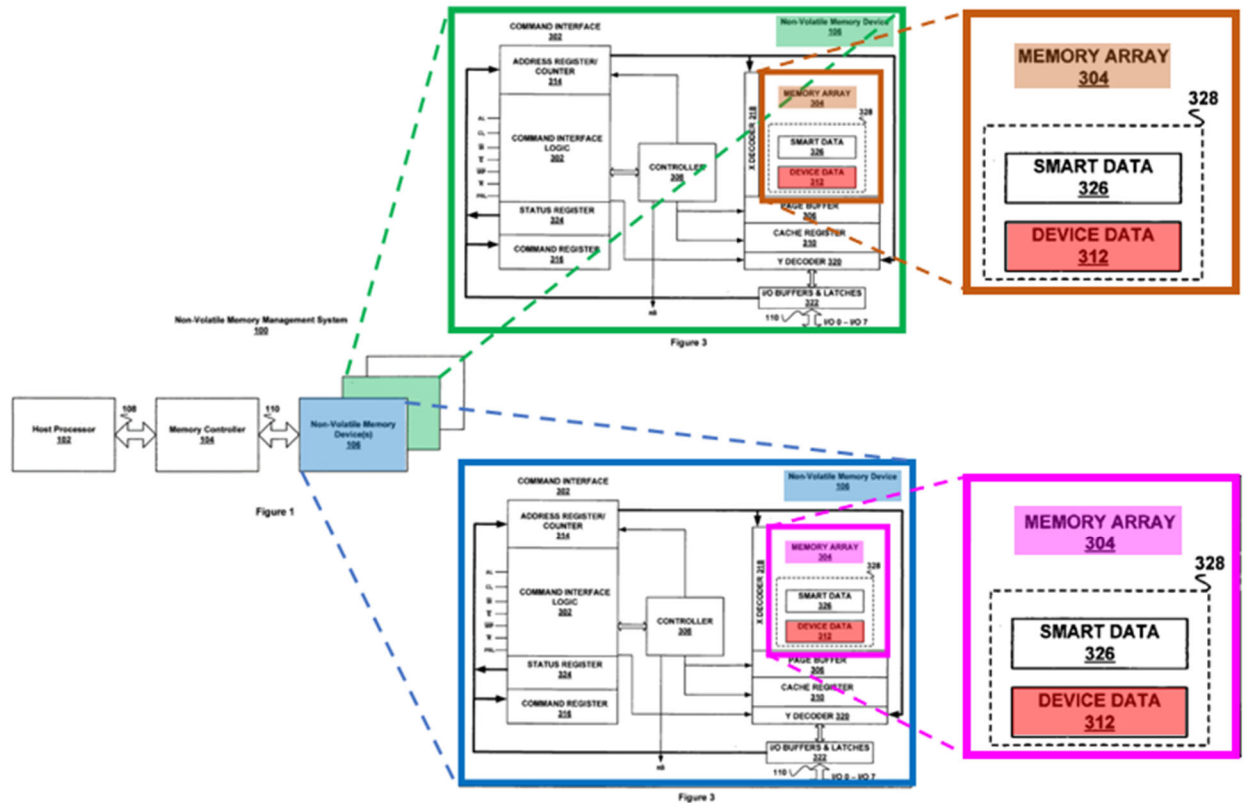
190. Cornwell also describes that “differences exist” between memory profiles. For example, Cornwell explains that different memory devices 106 may have different densities depending on whether they utilize single-layer cell technology or multi-layer cell technology, teaching that “syncing with memory devices 106 that include multi-level (MLC) technology can be performed at a different frequency than with memory devices 106 that include single-level cell technology (SLC).” *Id.*, ¶ [0078]. This is one example of a difference between profiles.

191. In my opinion, having a “difference exist” in these profiles would also

be obvious in view of Cornwell and a POSITA's knowledge. As demonstrated above, Cornwell identifies a large number of potential differences in device attributes (describing differences in device type, device density, spare area, block size, etc.). *Id.*, ¶ [0042]. A difference in any of these attributes, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). As a result, I believe that Cornwell renders obvious having a "difference exist" between memory profiles.

192. Similarly, Cornwell explains that memory devices have certain levels of bad blocks, which may increase over the lifetime of the device and which are identified so that the host system can avoid writing to them. *Id.*, ¶¶ [0054]-[0056]; *see also id.*, ¶ [0048] (explaining that flash memory "is available in a variety of block sizes" and memory access efficiency can be improved by "matching the average size of the files to be stored in the flash media to the block size of the flash media"); ¶ [0060] (describing different interleave levels for chips). As noted above, bad block identification is yet another example of an attribute described by Cornwell. *Id.*, ¶ [0042]. Accordingly, a difference in the number of bad blocks, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). For this additional reason, I believe that Cornwell renders obvious having a "difference exist" between memory profiles.

193. Annotated Figures 1 and 3, below, illustrate the **device profile** (i.e., the collection of attributes comprising device data 312) associated with both the **first memory location** and the **second memory location**:



194. For at least these reasons, it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious "said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.

- f. [Claim 29e]: "after identifying the command details, obtaining the first set of attributes;"

195. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that attributes comprising device data 312 can be obtained or “transmitted to the memory controller 104 via the I/O bus 110 *in response to a read command* issued by the memory controller.” Ex-1005, ¶ [0043] (emphasis added). Cornwell’s teaching that the attributes are transmitted “in response to a command” confirms that they are obtained after the command details are identified.

196. For at least these reasons, it is my opinion that Cornwell discloses “after identifying the command details, obtaining the first set of attributes.”

g. [Claim 29f]: “and using attributes from said first and second device profiles; and”

197. It is my opinion that Cornwell discloses this claim element. As explained above, Cornwell teaches that the first and second device profiles (i.e., collectively comprising device data 312) include a variety of attributes. *See supra* § X.A.4.e; Ex-1005, ¶ [0042] (describing attributes). Cornwell also teaches using the attributes from the first and second device profiles, explaining that “device data 312 can be transmitted to the memory controller 104 . . . *in response to a read command* issued by the by the memory controller” and “*used by the memory controller 104 and/or host system* to perform various memory management tasks.” Ex-1005, ¶ [0043] (emphasis added); *see also id.*, ¶ [0042] (explaining that “device data 312 can

be used to identify the memory device 106 and its manufacturer”); *id.*, ¶ [0046] (describing a memory management policy that “addresses how read and write operations should be performed”); *id.*, ¶ [0058] (explaining that the host processor “can request and receive memory device information (e.g., signatures, block size, interleave date, etc.) for one or more memory devices 106 over a standard IDE/ATA bus”).

198. For at least these reasons, it is my opinion that Cornwell discloses “using attributes from said first and second device profile.”

- h. [Claim 29g]: “selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

199. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches a memory transaction in which “the controller 308, *programs the page* into the memory array 304” Ex-1005, ¶ [0039] (emphasis added). As Cornwell further explains, a page has a specific “page size (1k, 2k, etc.).” *Id.*, ¶ [0042]; *see also, e.g., id.*, ¶ [0048] (“Memory access efficiency can be improved by *matching the average size of files* to be stored in the flash media *to the block size of the flash media.*”); *id.*, ¶ [0050] “[I]t may be advantageous to *tailor the size of files* such that the average file size is *proportional to the block size.*”) (all emphasis added). In this manner, Cornwell teaches selecting a transfer size for the

memory transaction.

200. Cornwell further discloses that the page size is a function of the device density of the memory transactions. *Id.*, ¶ [0042]. For example, Cornwell teaches varying “device density (e.g., 512 Mb, 1Gb, 2Gb, etc.).” *Id.*; *see also supra* § X.A.4.e. A POSITA would have understood that page size is the length of a row (number of columns) in the memory. A POSITA would have further understood that the length of a row, again the number of columns in a row, varies with the device density since the device density is the number of rows multiplied by the number of columns in a memory array multiplied by the number of arrays used in the memory chip.¹⁴ For example, a page size of 2k (i.e., there are 2,048 columns in a row), with 8k rows and 1k arrays would result in device density of 16Gb.

201. Cornwell also demonstrates that the transfer size is selected by the controller. As seen in Figure 1 of Cornwell, the host 102 communicates via bus 108 with the memory controller 104 in a format consistent with the IDE/ATA bus standard and its command, including the data size information (e.g., the number of logical sectors to be transferred), while the memory controller 104 communicates with the memory store via the I/O bus 110, using the signals required by the chips

¹⁴ *See supra* note 8.

used in the memory store and a transfer size, which is selected based on the data size information. As a result, the memory controller selects the transfer size for the memory transaction in response to a request from a host.

202. For instance, Cornwell teaches an example in which a media player sends a request to an operating system on a host. *Id.*, ¶ [0025]. This “request is received by the operating system, which formats the request into IDE/ATA signals, which are transmitted to the memory controller 104 on the IDE/ATA bus 108 by the host processor 102.” *Id.*¹⁵ The fact that host sends the requests to the memory controller—which must then communicate with the memory store—ensures that the memory controller selects the transfer size for the memory transactions, where the transfer size is a function of the data size of the memory transaction and the first set of attributes to retrieve the requested media file.

203. For at least these reasons, it is my opinion that Cornwell discloses “selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

5. Claim 36 Is Rendered Obvious by Cornwell in View of a

¹⁵ As Cornwell also explains, “ATA-6 is the latest version of the IDE/ATA standard, which was approved by the American National Standards Institute (ANSI) in 2001 under document NCITS 347-2001.” *Id.*, ¶ [0026].

POSITA's Knowledge

204. As explained below, it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious Claim 36 of the '190 patent.

- a. **[Claim 36]: “The computer readable medium of claim 29, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

205. It is my opinion that Cornwell renders obvious the computer readable medium of claim 29 for the reasons discussed above. *Supra* § X.A.4. It is also my opinion that Cornwell renders obvious the computer readable medium of claim 36.

206. As explained above, Cornwell teaches that the first and second device profiles (i.e., each set of device data 312) have shared attributes. *Supra* § X.A.4.e; *see also* Ex-1005, ¶ [0042]. For example, each device profile may have a “device type” attribute, “device density” attribute, “page size” attribute, etc. *Supra* § X.A.4.e; Ex-1005, ¶ [0042]. Further, as explained above, it is my opinion that having different attribute qualifiers for these shared attributes would have been obvious in view of Cornwell and a POSITA's knowledge. *Supra* § X.A.4.e. For example, the first memory location could have a “NAND” qualifier for the “device type” attribute, while the second memory location could have “NOR” qualifier for the “device type” attribute. Ex-1005, ¶ [0042]. Likewise, the first memory location could have a “512 Mb” qualifier for the “device density” attribute, while the second

memory location could have a “1Gb” qualifier for the “device density” attribute. *Id.* Cornwell discloses a variety of different attribute qualifiers for each shared attribute in the first and second device profiles, as illustrated below:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

207. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “computer readable medium of claim 29, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

6. Claim 37 Is Rendered Obvious by Cornwell in view of a POSITA’s Knowledge

208. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 37 of the ’190 patent.

a. [Claim 37]: “The computer readable medium of claim

36, wherein said different attribute qualifiers include different attribute values.”

209. It is my opinion that Cornwell renders obvious the method of claim 36 for the reasons discussed above. *Supra* § X.A.5.a. It is also my opinion that Cornwell renders obvious the method recited in claim 37.

210. As explained above, it is my opinion that having different attribute qualifiers would have been rendered obvious by Cornwell in view of a POSITA’s knowledge. *Supra* § X.A.4.e. For example, Cornwell teaches that the qualifiers associated with the “device density” attribute can have different values such as 512 Mb, 1 Gb, 2 Gb, etc. Ex-1005, ¶ [0042]. Likewise, Cornwell teaches that the qualifiers associated with the “page size” attribute can have different values such as 1k, 2K, etc. *Id.*; *see also supra* § X.A.4.e. Accordingly, it is my opinion that would have been obvious in view of Cornwell and a POSITA’s knowledge to have different attribute values for attribute qualifiers.

211. A variety of potential differences in attribute values are illustrated in the following excerpt from Cornwell:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., x8, x16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

212. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “computer readable medium of claim 36, wherein said different attribute qualifiers include different attribute values.”

7. Claim 45 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

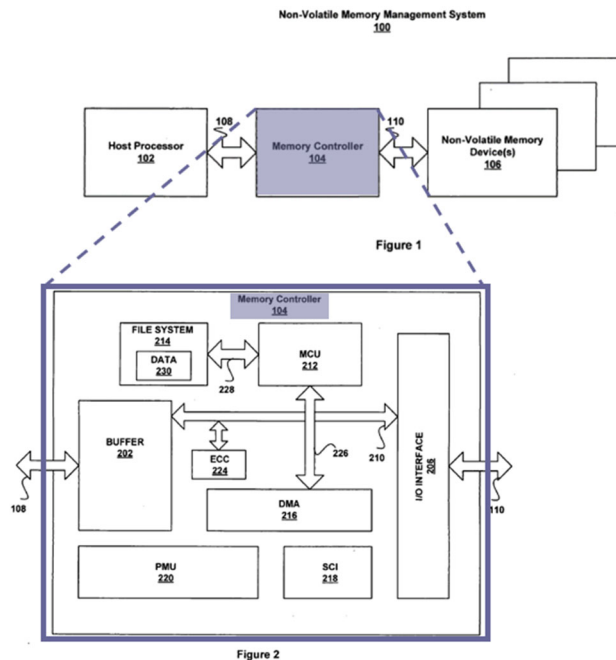
213. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 45 of the ’190 patent.

a. [Claim 45 pre]: “A memory controller for performing memory transactions, said memory controller comprising:”

214. To the extent the preamble of claim 45 is limiting, it is my opinion that Cornwell discloses this claim element. Cornwell teaches that “[a] memory controller is operatively coupled to the one or more non-volatile memory devices and

configurable to access the one or more non-volatile memory devices in accordance with a memory management policy.” Ex-1005, ¶ [0007]. The “memory controller 104 can be *any device that manages memory access*, including but not limited to: programmable memory controllers, flash disk controllers, direct memory access (DMA) controllers, logic devices, field-programmable gate arrays (FPGAs), central processing units (CPUs), etc.” *Id.*, ¶ [0019] (emphasis added); *see also id.*, ¶ [0021] (teaching that “the memory controller 104 recognizes control, address, and data signals transmitted on the bus 108 by the host processor” and “*translates the control, address and data signals into memory access requests* on memory devices 106”) (emphasis added); *id.* at ¶¶ [0028]-[0033] (describing the memory controller); *id.*, ¶ [0038] (explaining that the memory controller carries out “read and write operations on memory array 304”).

215. Annotated Figures 1 and 2, below, illustrate the **memory controller** (“memory controller 104”) for performing memory transactions:



216. For at least these reasons, it is my opinion that Cornwell discloses a “memory controller for performing memory transactions.”

- b. **[Claim 45a]: “means for performing a memory transaction comprising a means for addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile;”**

217. It is my opinion that Cornwell discloses this claim element. As explained above, this claim element contains two means plus function limitations. I address each of these means plus function limitations in turn.

218. The first means-plus-function element recited in the claim is a “means for performing a memory transaction.” As explained above, this term has a recited function of “performing a memory transaction.” *Supra* § VIII.B. Pursuant to

Petitioner’s proposed construction, the structure for performing this function is a “memory controller or similar structure configured to perform a memory transaction.”

Id. Cornwell discloses both the recited function and structure.

219. For example, Cornwell expressly teaches “[a] **memory controller** [that] is operatively coupled to the one or more non-volatile memory devices and **configurable to access the one or more non-volatile memory devices.**” Ex-1005, ¶ [0007] (emphasis added). Likewise, Cornwell teaches that the memory controller performs a memory transaction by “translat[ing] control, address and data signals into memory access requests on memory devices 106.” *Id.*, ¶ [0021]; *see also id.*, ¶¶ [0028]-[0033] (explaining the operation of the memory controller in data transactions).

220. Accordingly, it is my opinion that Cornwell discloses means for performing a memory transaction—i.e., a “memory controller or similar structure configured to perform a memory transaction.”

221. The second means-plus-function element recited in this claim is a “means for addressing a first memory location.” As explained above, this term has a recited function of “addressing a first memory location.” *Supra* § VIII.D. Pursuant to Petitioner’s construction, the structure for performing this function is “a memory device interface, a host interface, an interface controller or similar structure

configured to process memory transaction requests, and associated communication pathways.” *Id.* Cornwell discloses both the recited function and structure.

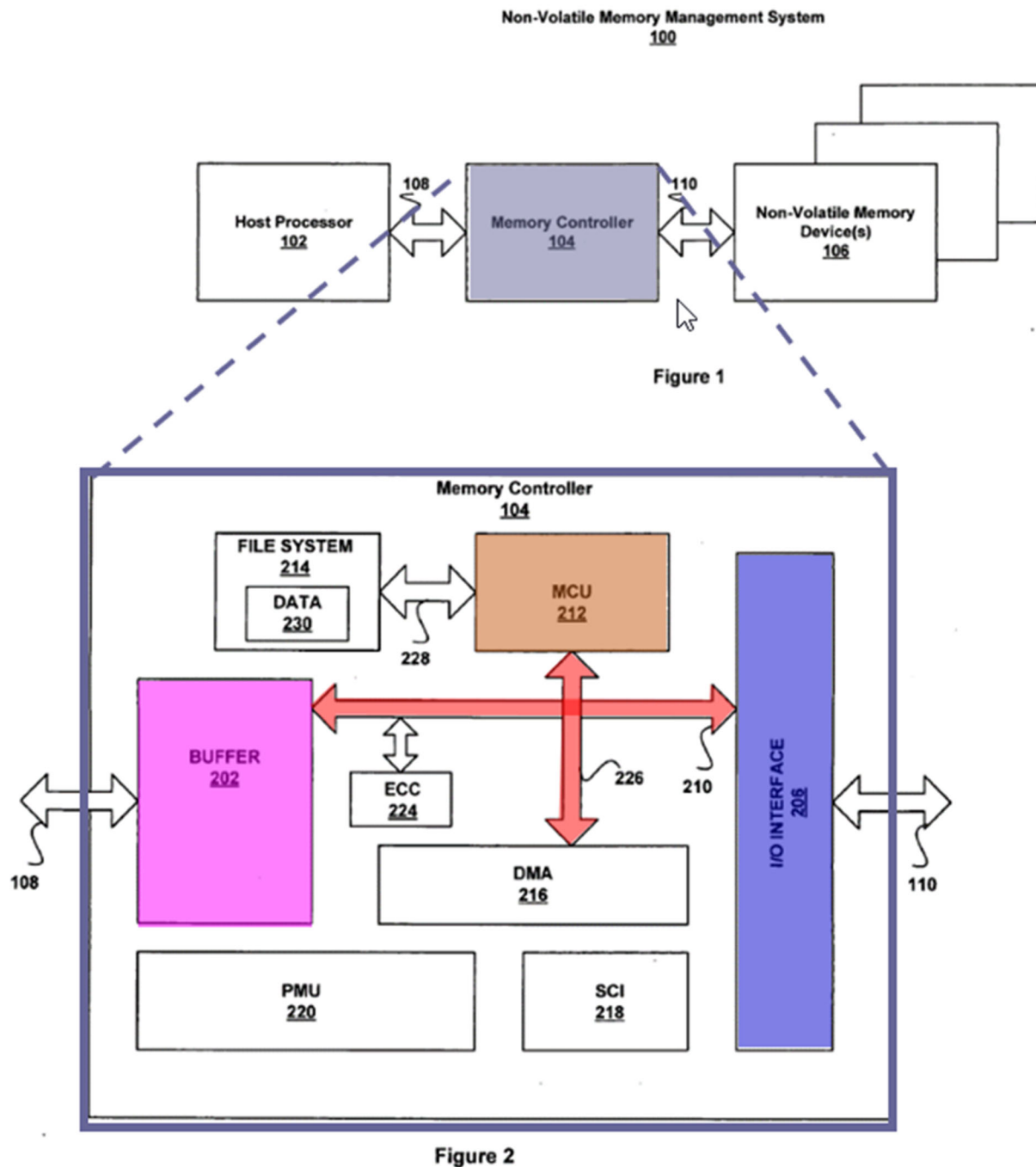
222. First, Cornwell teaches a memory device interface, which is associated with addressing a first memory location. For example, Cornwell discloses an “I/O interface 206” that “provides connectivity to the memory devices 106 through I/O bus 110, and includes circuitry for enabling read, program and erase operations to one or more memory devices. In some implementations, the I/O interface 206 is a multitasking interface that allows concurrent read, program and erase operations to multiple memory devices 106.” *Id.*, ¶ [0031]; *see also id.*, claim 16 (reciting a memory controller comprising “a first interface adapted for coupling to one or more non-volatile memory devices”).

223. Second, Cornwell also discloses a host interface that is associated with addressing a first memory location. For example, Cornwell that the “host processor is coupled to a memory controller and configured to retrieve . . . at least one attribute of at least one non-volatile memory device operatively coupled to the memory controller.” *Id.*, Abstract; *see also id.*, ¶ [0007]. Accordingly, Cornwell recites a memory controller comprising “a second interface adapted for coupling to a host processor, and configurable to receive a request from the host processor.”

224. Third, Cornwell discloses an interface controller or similar structure

configured to process memory transaction request. In particular, Cornwell discloses that the memory controller comprises a “micro- controller unit (MCU) 212.” *Id.*, ¶ [0028]. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029]. The file system also stores data that can “include an electronic signature or serial number for identifying the memory device 106 or its manufacturer, the block size of the memory controller 104, an identification of bad blocks, chip interleave depth, etc.” *Id.* Accordingly, Cornwell confirms that the MCU is an interface controller or similar structure configured to process memory transaction request.

225. Annotated Figures 1 and 2, below, illustrate Petitioner’s proposed structure of “a **memory device interface**, a **host interface**, an **interface controller** or similar structure configured to process memory transaction request, and **associated communication pathways**”:



226. Cornwell also teaches that the first and second memory locations are

respectively associated with a first device profile and a second device profile as explained above for element [1b]. *Supra* § X.A.1.c.

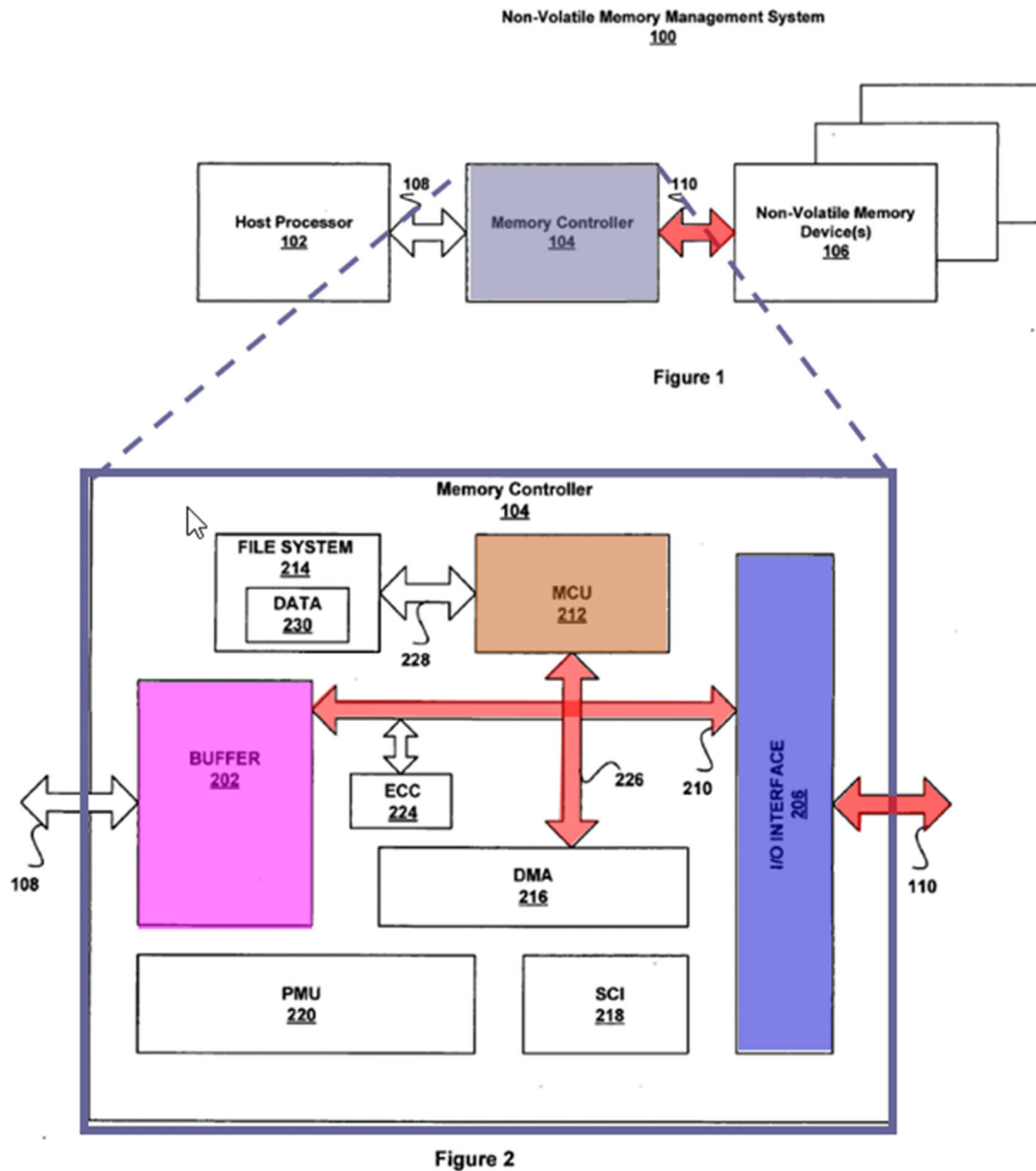
227. For at least these reasons, Cornwell discloses “means for performing a memory transaction comprising a means for addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile” pursuant to Petitioners’ proposed construction.

c. [Claim 45b]: “said means for addressing is directly coupled to the first memory location and second memory location;”

228. It is my opinion that Cornwell discloses this claim limitation. Cornwell teaches that the memory device interface (I/O interface 206), which is part of the structure for the means for addressing as noted above, “provides connectivity to the memory devices 106 through I/O bus 110, and includes circuitry for enabling read, program and erase operations to one or more memory devices. In some implementations, the I/O interface 206 is a multitasking interface that allows concurrent read, program and erase operations to multiple memory devices 106.” Ex-1005, ¶ [0031]; *see also id.*, claim 16 (reciting a memory controller comprising “a first interface adapted for coupling to one or more non-volatile memory devices”).

229. Annotated Figures 1 and 2 illustrated that the **memory device interface**

(I/O interface) is directly connected to the memory devices via **bus 110**:



230. For at least these reasons, it is my opinion that Cornwell discloses “said means for addressing is directly coupled to the first memory location and second memory location.”

d. [Claim 45c]: “wherein said means for addressing

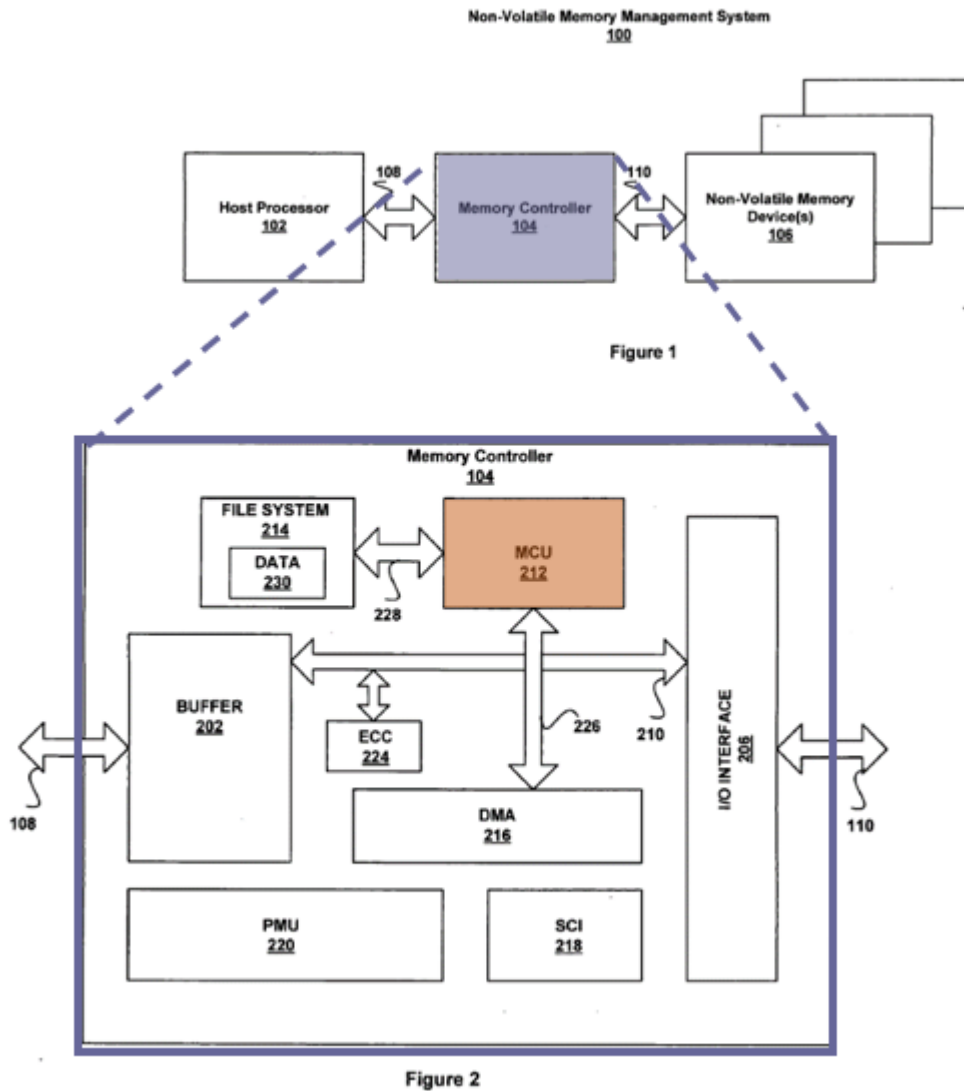
includes an interface controller means for processing memory transaction requests;”

231. It is my opinion that Cornwell discloses this claim limitation. As explained above, “interface controller means for processing memory transaction requests” is a means-plus-function term with a recited function of “processing memory transaction requests.” *Supra* § VIII.C. Pursuant to Petitioner’s proposed construction, the structure for performing this function is an “interface controller or similar structure configured to process memory transaction requests.” *Id.* Cornwell discloses both the recited function and structure.

232. As previously explained, Cornwell discloses that the memory controller comprises a “micro-controller unit (MCU) 212.” *Id.*, ¶ [0028]. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029]. The file system also stores data that can “include an electronic signature or serial number for identifying the memory device 106 or its manufacturer, the block size of the memory controller 104, an identification of bad blocks, chip interleave depth, etc.” *Id.* Accordingly, the MCU is an interface controller or similar structure configured to process memory

transaction requests.

233. The **interface controller** (MCU) is illustrated in annotated Figures 1 and 2 below:



234. For at least these reasons, it is my opinion that Cornwell discloses “wherein said means for addressing includes an interface controller means for processing memory transaction requests.”

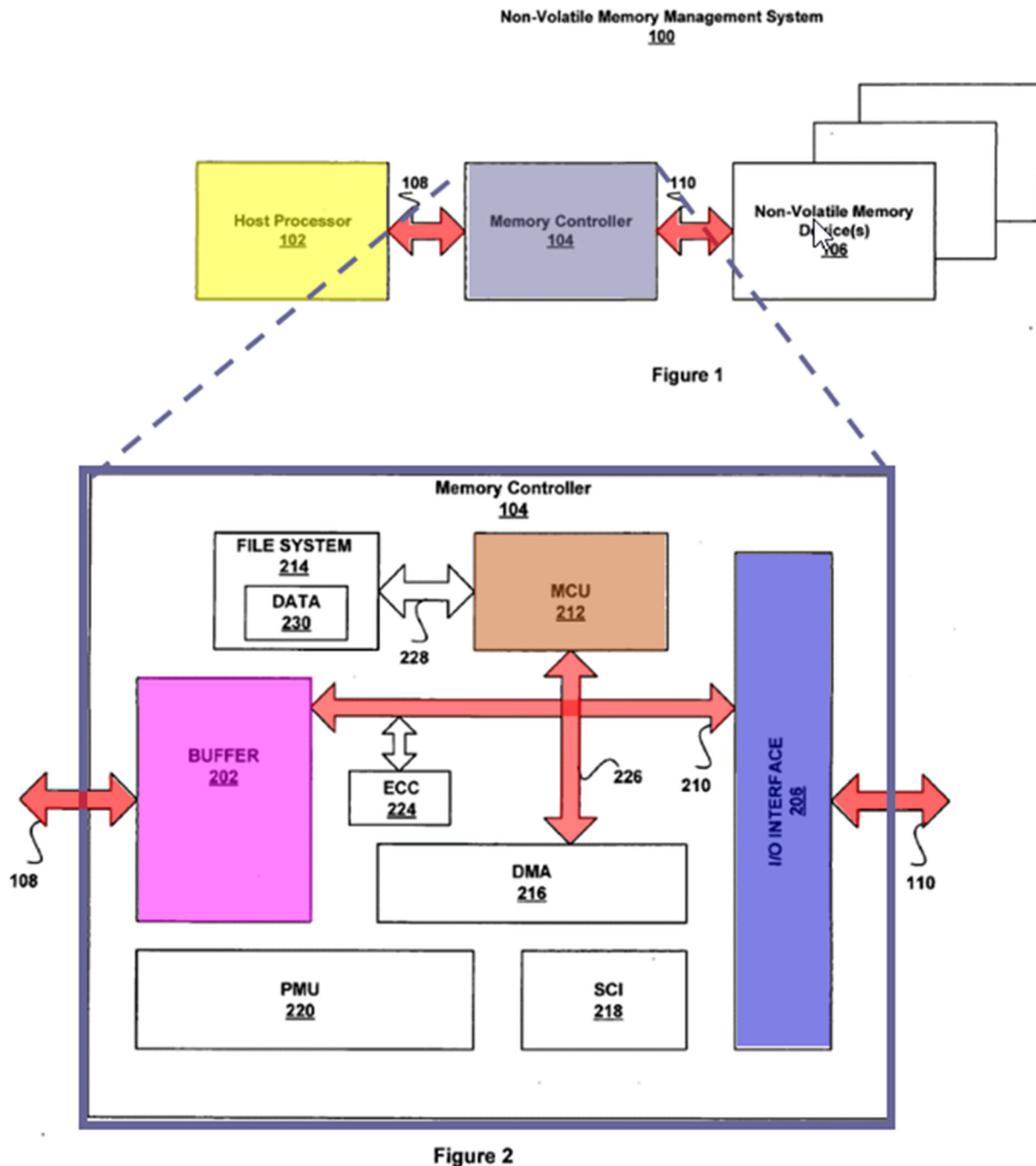
- e. **[Claim 45d]: “said means for addressing is directly coupled to a host as said means for addressing addresses the first memory location or second memory location;”**

235. It is my opinion that Cornwell discloses this claim limitation. As explained above, the memory device interface—which is a component of the means for addressing—is directly coupled to the host as the first and second memory locations are addressed. For example, Cornwell teaches that the memory controller is coupled to the host processor and is and “*configured to retrieve . . . at least one attribute of at least one non-volatile memory device* operatively coupled to the memory controller.” Ex-1005, Abstract; *see also id.*, ¶ [0007]; claim 16.

236. Further, Cornwell’s disclosure of the interface controller (MCU) confirms that it is coupled to the host through the host interface while the memory locations are addressed. For example, Cornwell teaches that “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029].

237. Annotated Figures 1 and 2, shown below, illustrate that Cornwell’s means for addressing—i.e., the **memory device interface**, a **host interface**, an

interface controller, and **associated communication pathways**—is directly coupled to the **host processor** as the first or second memory location are addressed:



238. For at least these reasons, it is my opinion that Cornwell discloses “said

means for addressing is directly coupled to a host as said means for addressing addresses the first memory location or second memory location.”

- f. **[Claim 45e]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”**

239. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. As explained above, Cornwell teaches a “device profile” in the form of a collection of “device data 312.” *See supra* § X.A.7.b. This device profile “can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. Information comprising the device profile may include “device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.*

240. A POSITA would have also understood that the device profiles taught in Cornwell are optimal for a data type subject to the memory transaction. For example, as explained in the preceding paragraph, Cornwell teaches that the device profile identifies whether the memory is NAND or NOR. *Id.* At the time Cornwell

was filed, a POSITA would have readily understood that NOR provides random access (i.e., access to a “random data type”) and NAND provides sequential access (i.e., access to a sequential data type”).¹⁶ Accordingly, it would have been obvious to a POSITA, based on Cornwell alone, that a profile for NOR memory would have been optimal for a random data type subject to a memory transaction, and that a profile for NAND memory would have been optimal for a sequential data type subject to a memory transaction.

241. Similarly, Cornwell teaches “block defining” as another way of ensuring that a device profile is optimal for a data type subject to the memory transactions. *See id.*, ¶¶ [0048]-[0050]. For example, Cornwell teaches that “[m]emory access efficiency can be improved by matching the average size of files to be stored in the flash media *to the block size of the flash media.*” *Id.*, ¶ [0048]; *see also id.*, ¶ [0050] (explaining that “it may be advantageous to *tailor the size of files* such that the *average file size is proportional to the block size.*”) (all emphasis added). As noted above, block size is one of the device profile attributes taught by Cornwell. The district court’s claim construction order states that such block size comparison information is another example of a device profile that is optimal for a

¹⁶ *See supra* note 6.

data type subject to the memory transaction.¹⁷

242. Moreover, Cornwell teaches that the data type comprises one of a random data type or a sequential data type, explaining that “[i]n some implementations a *random [data type] read command* may be issued first, followed by a page [*sequential data type*] read command.” *Id.*, ¶ [0040] (emphasis added). Thus, by matching block size to file size, and teaching that files contain random or sequential data types, Cornwell further teaches first device profile is optimal for a data type subject to the memory transaction.

243. For at least these reasons, it is my opinion that Cornwell in view of a POSITA’s knowledge renders obvious “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

g. [Claim 45f]: “said performing the memory transaction

¹⁷ As noted above, the district court’s claim construction order states that examples of optimal device profiles are “(1) a device profile that includes a block size attribute value that is the largest available from the other device profiles when the data type is sequential,” and “(2) a device profile that includes a block size attribute value that is the smallest available from the other device profiles” when the data type is random.” *Supra* note 7 (citing Ex-1007 at 44).

further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”

244. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the memory device “generally includes a command interface 302, a memory array 304 and a controller 308.” Ex-1005, ¶ [0034]. Cornwell further identifies a variety of commands that are “translated into signals which can be used by a controller and decoding logic in the memory device 106 to access a memory array.” *Id.*, ¶ [0027]. “For example, when the host processor 102 makes a read request, the “Read Sectors” opcode (20 h) is transmitted to the memory controller 104, together with address and control signals for accessing the sector(s).” *Id.*; *see also id.*, ¶ [0035] (describing a decoder that “decodes signals from the command interface logic 302 for reading or writing data into the memory line”); ¶ [0038] (explaining that the controller receives “one or more input addresses and command inputs, which are used by the controller 308 in combination with the control signals from the command interface logic 302 to carry out read and write operations on memory array 304”). Likewise, Cornwell teaches an “‘identify’ command that returns a chip ID,” which can then be used “to look-up information about the memory devices 106.” *Id.*, ¶ [0077]. As a result, Cornwell discloses identifying command details for causing the memory transaction to be performed, wherein the

command details comprise the first memory device.

245. Examples of such commands are provided in Table 1 of Cornwell, reproduced below:

TABLE I	
<u>Examples of Standard ATA-6 Commands</u>	
Opcode	Command
10h	Recalibrate
20h	Read Sectors
30h	Write Sectors
40h	Read Verify
B0h	SMART
C8h	Read DMA
CAh	Write DMA
E0h	Standby Immediate
E2h	Standby
E7h	Flush Cache
ECh	Identify
EFh	Set Features

246. The **command interface logic 302**, which identifies command details for causing the memory transaction to be performed, is also illustrated in annotated Figure 3 below:

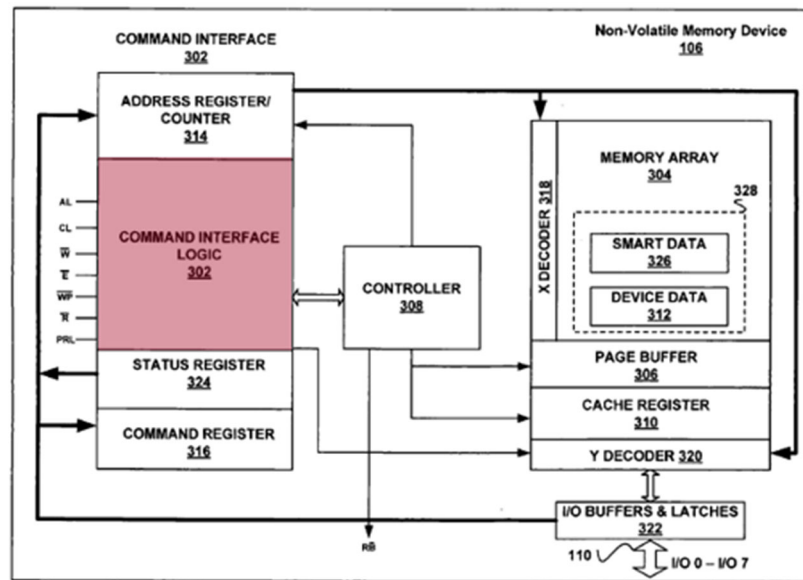


Figure 3

247. For at least these reasons, it is my opinion that Cornwell teaches “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- h. **[Claim 45g]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

248. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. Cornwell teaches that “the device data 312,” which collectively comprises the device profile, “can include an electronic signature or serial number that includes a manufacturer code and/or device code.” Ex-1005, ¶

[0042]. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer.” *Id.* For example, device data 312 can include the following:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Potential
differences
between first and
second device

Id. (highlighting and underlining added). Cornwell expressly states that this information “is associated with *attributes*, properties or characteristics of the memory device 106 (*collectively, referred to herein as “attributes”*).” *Id.*

249. Cornwell also describes that “differences exist” between memory profiles. For example, Cornwell explains that different memory devices 106 may have different densities depending on whether they utilize single-layer cell technology or multi-layer cell technology, teaching that “syncing with memory devices 106 that include multi-level (MLC) technology can be performed at a different frequency than with memory devices 106 that include single-level cell

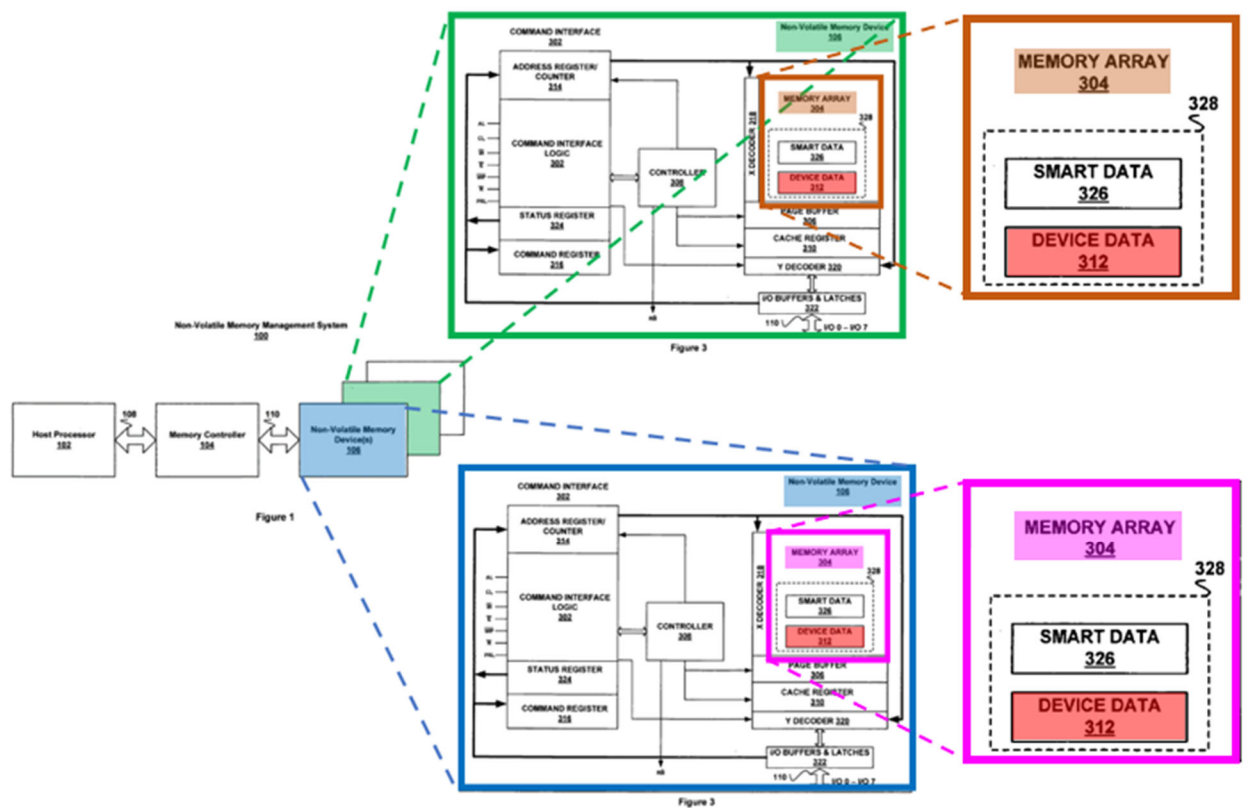
technology (SLC).” *Id.*, [0078]. This is one example of a difference between profiles.

250. In my opinion, having a “difference exist” in these profiles would also be obvious in view of Cornwell and a POSITA’s knowledge. As demonstrated above, Cornwell identifies a large number of potential differences in device attributes (describing possible differences in device type, device density, spare area, block size, etc.). *Id.*, ¶ [0042]. A difference in any of these attributes, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). As a result, I believe that Cornwell renders obvious having a “difference exist” between memory profiles.

251. Similarly, Cornwell explains that memory devices may have certain levels of bad blocks, which may increase over the lifetime of the device and which are identified so that the host system can avoid writing to them. *Id.*, ¶¶ [0054]-[0056]; *see also id.*, ¶ [0048] (explaining that flash memory “is available in a variety of block sizes” and memory access efficiency can be improved by “matching the average size of the files to be stored in the flash media to the block size of the flash media”); ¶ [0060] (describing different interleave levels for chips). As noted above, bad block identification is yet another example of an attribute described by Cornwell. *Id.*, ¶ [0042]. Accordingly, a difference in the number of bad blocks, as shown by

Cornwell, would constitute a difference in the device profile (chip data 312). For this additional reason, I believe that Cornwell renders obvious having a “difference exist” between memory profiles.

252. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of attributes comprising device data 312) associated with both the **first memory location** and the **second memory location**:



253. For at least these reasons, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious “said first device profile representing a first set of attributes of said first memory location, and said second device profile

representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

- i. **[Claim 45h]: “said means for addressing obtains the first set of attributes after identifying the command details;”**

254. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that attributes comprising device data 312 can be obtained or “transmitted to the memory controller 104 via the I/O bus 110 *in response to a read command* issued by the memory controller.” Ex-1005, ¶ [0043]. Cornwell’s teaching that the attributes are transmitted “in response to a command” confirms that they are obtained after the command details are identified.

255. Cornwell also discloses that the interface controller (MCU 212), which a structural component of the means for addressing, performs these functions. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029].

256. For at least these reasons, it is my opinion that Cornwell discloses “said means for addressing obtains the first set of attributes after identifying the command

details.”

- j. **[Claim 45i]: “and said means for addressing uses attributes from said first and second device profiles and selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

257. It is my opinion that Cornwell discloses this claim element. As explained above, Cornwell teaches that the first and second device profiles (i.e., collectively comprising device data 312) include a variety of attributes. *See supra* § X.A.7.b; Ex-1005, ¶ [0042] (describing attributes). Cornwell also teaches using the attributes from the first and second device profiles, explaining that “device data 312 can be transmitted to the memory controller 104” and “*used by the memory controller 104 and/or host system* to perform various memory management tasks, as described with respect to Fig. 4.” Ex-1005, ¶ [0043] (emphasis added); *see also id.*, ¶ [0042] (explaining that “device data 312 can be used to identify the memory device 106 and its manufacturer”); ¶ [0058] (explaining that the host processor “can request and receive memory device information (e.g., signatures, block size, interleave date, etc.) for one or more memory devices 106 over a standard IDE/ATA bus”).

258. Cornwell also teaches selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory

transaction and the first set of attributes. For example, Cornwell teaches a memory transaction in which “*the controller 308, programs the page* into the memory array 304” Ex-1005, ¶ [0039] (emphasis added). As Cornwell further explains, a page has a specific “page size (1k, 2k, etc.).” *Id.*, ¶ [0042]; *see also, e.g., id.*, ¶ [0048] (“Memory access efficiency can be improved by *matching the average size of files* to be stored in the flash media *to the block size of the flash media.*”); *id.*, ¶ [0050] “[I]t may be advantageous to *tailor the size of files* such that the average file size is *proportional to the block size.*”) (all emphasis added). In this manner, Cornwell teaches selecting a transfer size for the memory transaction.

259. Cornwell further discloses that the page size is a function of the device density of the memory transactions. *Id.*, ¶ [0042]. For example, Cornwell teaches varying “device density (e.g., 512 Mb, 1Gb, 2Gb, etc.).” *Id.*; *see also supra* § X.A.7.b. A POSITA would have understood that page size is the length of a row (number of columns) in the memory. A POSITA would have further understood that the length of a row, again the number of columns in a row, varies with the device density since the device density is the number of rows multiplied by the number of columns in a memory array multiplied by the number of arrays used in the memory

chip.¹⁸ For example, a page size of 2k (i.e., there are 2,048 columns in a row), with 8k rows and 1k arrays would result in device density of 16Gb.

260. Cornwell also demonstrates that the transfer size is selected by the controller. As seen in Figure 1 of Cornwell, the host 102 communicates via bus 108 with the memory controller 104 in a format consistent with the IDE/ATA bus standard and its command, including the data size information (e.g., the number of logical sectors to be transferred), while the memory controller 104 communicates with the memory store via the I/O bus 110, using the signals required by the chips used in the memory store and a transfer size, which is selected based on the data size information. As a result, the memory controller selects the transfer size for the memory transaction in response to a request from a host.

261. For instance, Cornwell teaches an example in which a media player sends a request to an operating system on a host. *Id.*, ¶ [0025]. This “request is received by the operating system, which formats the request into IDE/ATA signals, which are transmitted to the memory controller 104 on the IDE/ATA bus 108 by the host processor 102.” *Id.*¹⁹ The fact that host sends the requests to the memory

¹⁸ See *supra* note 8.

¹⁹ As Cornwell also explains, “ATA-6 is the latest version of the IDE/ATA standard,

controller—which must then communicate with the memory store—ensures that the memory controller selects the transfer size for the memory transactions, where the transfer size is a function of the data size of the memory transaction and the first set of attributes to retrieve the requested media file.

262. Cornwell also teaches that these functions are performed by the interface controller (MCU 212), which is a structural component of the means for addressing. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029]. The file system also stores data that can “include an electronic signature or serial number for identifying the memory device 106 or its manufacturer, the block size of the memory controller 104, an identification of bad blocks, chip interleave depth, etc.” *Id.*

263. For at least these reasons, it is my opinion that Cornwell discloses “said

which was approved by the American National Standards Institute (ANSI) in 2001 under document NCITS 347-2001.” *Id.*, ¶ [0026].

means for addressing uses attributes from said first and second device profiles and selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

8. Claim 51 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

264. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 51 of the ’190 patent.

- a. [Claim 51]: “The memory controller of claim 45, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

265. It is my opinion that Cornwell renders obvious the memory controller of claim 45 for the reasons discussed above. *Supra* § X.A.7. It is also my opinion that Cornwell renders obvious the memory controller recited in claim 51.

266. As explained above, Cornwell teaches that the first and second device profiles (i.e., each set of device data 312) have shared attributes. *Supra* § X.A.7.b; Ex-1005, ¶ [0042]. For example, each device profile may have a “device type” attribute, “device density” attribute, “page size” attribute, etc. *Supra* § X.A.7.b; Ex-1005, ¶ [0042]. Further, as explained above, it is my opinion that having different attribute qualifiers for these shared attributes would have been obvious in view of Cornwell and a POSITA’s knowledge. *Supra* § X.A.7.h. For example, the first

memory location could have a “NAND” qualifier for the “device type” attribute, while the second memory location could have “NOR” qualifier for the “device type” attribute. Ex-1005, ¶ [0042]. Likewise, the first memory location could have a “512 Mb” qualifier for the “device density” attribute, while the second memory location could have a “1Gb” qualifier for the “device density” attribute. *Id.* Cornwell discloses a variety of different attribute qualifiers for each shared attribute in the first and second device profiles, as illustrated below:

manufacturer code and/or device code. Chip data 312 can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

267. For at least these reasons, Cornwell in view of a POSITA’s knowledge renders obvious the “memory controller of claim 45, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

9. Claim 52 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge

268. As explained below, it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious Claim 52 of the '190 patent.

a. [Claim 52]: “The memory controller of claim 51, wherein said different attribute qualifiers include different attribute values.

269. It is my opinion that Cornwell renders obvious the memory controller of claim 51 for the reasons discussed above. *Supra* § X.A.8. It is also my opinion that Cornwell renders obvious the memory controller recited in claim 52.

270. As explained above, it is my opinion that having different attribute qualifiers would have been rendered obvious by Cornwell in view of a POSITA's knowledge. *Supra* § X.A.7.h. Cornwell also teaches that each attribute qualifier has an attribute value, which may be different. For example, the “device density” attribute can have different values such as 512 Mb, 1 Gb, 2 Gb, etc. Ex-1005, ¶ [0042]. Likewise, the qualifiers associated with the “page size” attribute can have different values such as 1k, 2K, etc. *Id.*; *see also supra* § X.A.7.b. Accordingly, it is my opinion that would have been obvious in view of Cornwell and a POSITA's knowledge to have different attribute values for attribute qualifiers.

271. A variety of potential differences in attribute values are illustrated in the following excerpt from Cornwell:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

272. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “memory controller of claim 51, wherein said different attribute qualifiers include different attribute values.”

10. Claim 59 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

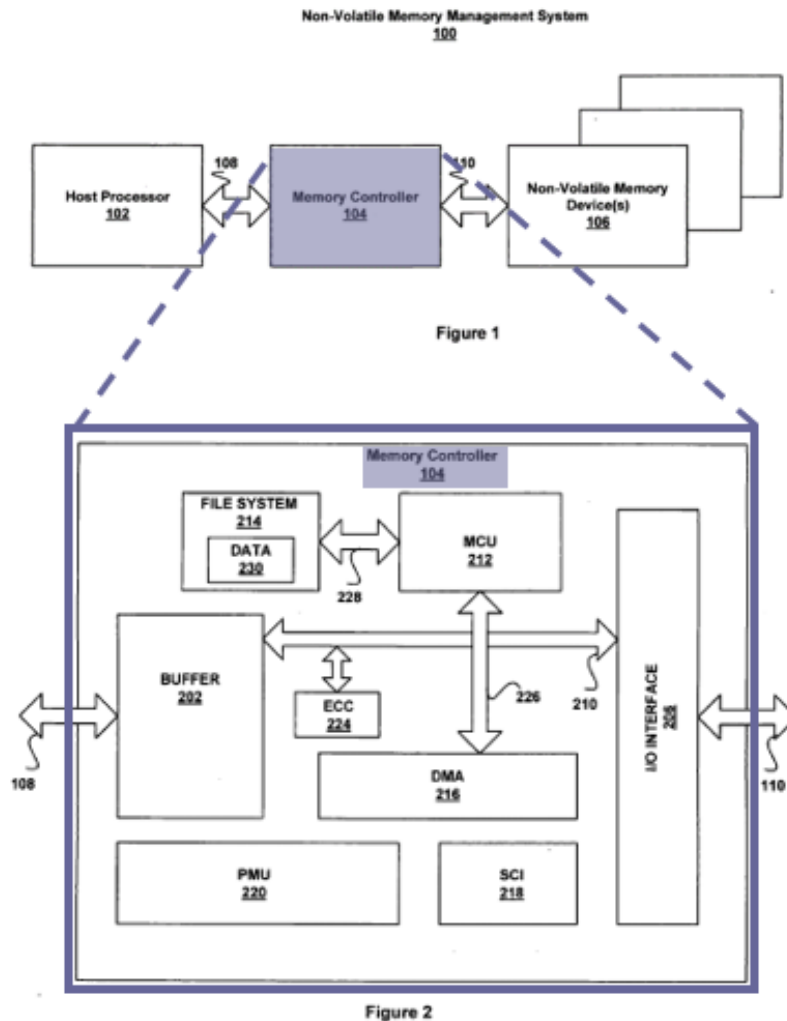
273. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 59 of the ’190 patent.

a. [Claim 59 pre]: “A memory controller comprising:”

274. To the extent the preamble of claim 59 is limiting, it is my opinion that Cornwell discloses this claim element. Cornwell teaches “[a] memory controller [that] is operatively coupled to the one or more non-volatile memory devices and configurable to access the one or more non-volatile memory devices in accordance

with a memory management policy.” Ex-1005, ¶ [0007]; *see also id.*, ¶ [0019] (noting that “memory controller 104 can be any device that manages memory access, including but not limited to: programmable memory controllers, flash disk controllers, direct memory access (DMA) controllers, logic devices, field-programmable gate arrays (FPGAs), central processing units (CPUs), etc.”); *id.*, ¶¶ [0028]-[0033] (describing the memory controller).

275. Annotated Figures 1 and 2 of Cornwell, below, illustrate the **memory controller** (“memory controller 104”):



276. For at least these reasons, it is my opinion that Cornwell discloses “[a] memory controller.”

- b. **[Claim 59a]: “an interface controller coupled to a memory device interface and an input/output (IO) device interface;”**

277. It is my opinion that Cornwell discloses this claim element. For example, as explained above, Cornwell discloses that the memory controller comprises a “micro- controller unit (MCU) 212.” Ex-1005, ¶ [0028]. “The MCU

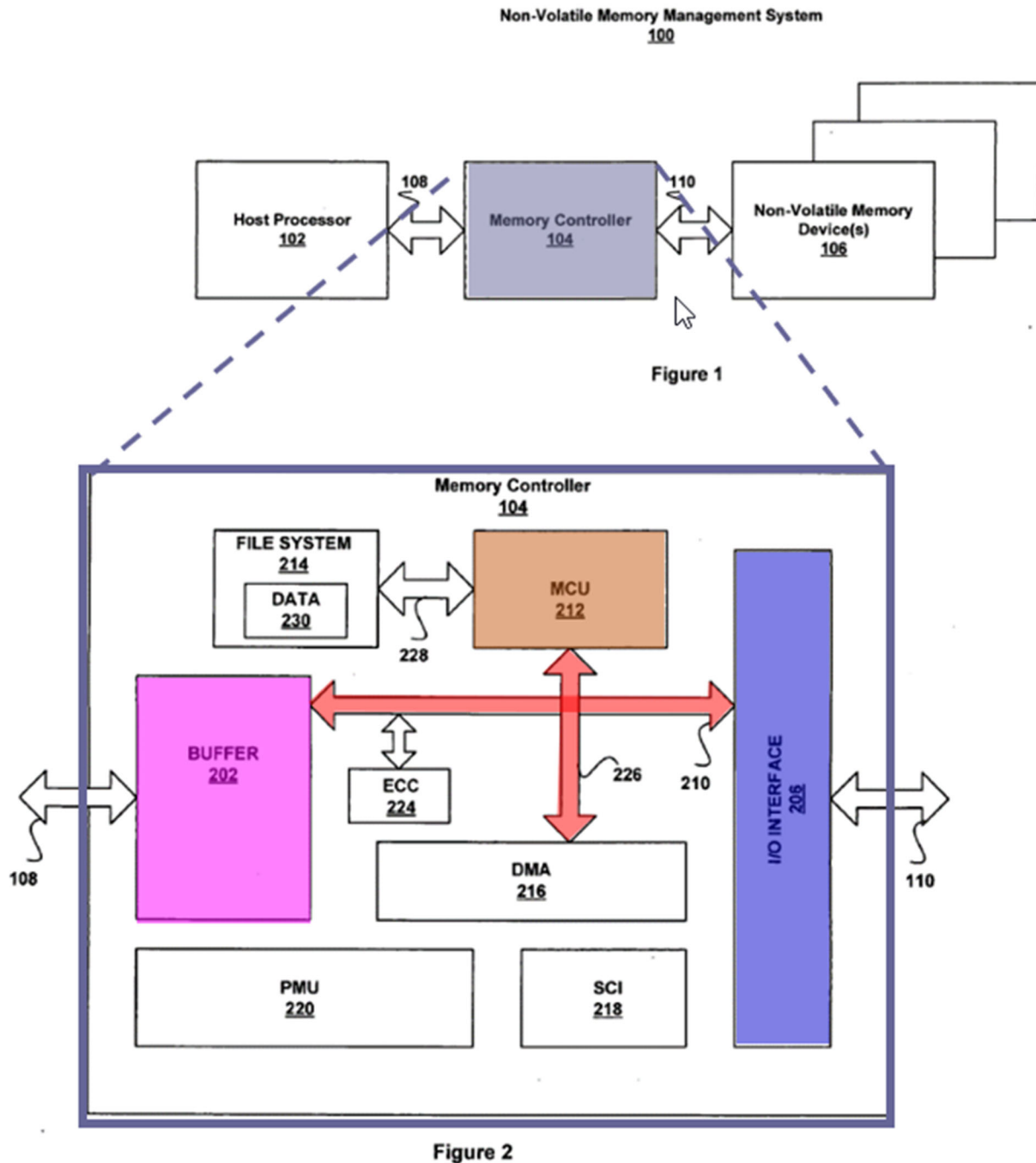
212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029]. Accordingly, the MCU is an interface controller.

278. The MCU is also coupled to a memory device interface. As Cornwell explains, “[d]uring operation, one or more memory devices 106 receive signals from the memory controller 104 over Input/Output (I/O) bus 110, which enables the memory devices 106 to perform memory access requests (e.g., read or write operations).” *Id.*, ¶ [0024]; *see also id.*, ¶ [0029] (explaining that the “MCU firmware can translate signals from the host processor 102 into memory read and write operations.”). In particular, “[t]he I/O interface 206 provides connectivity to the memory devices 106 through I/O bus 110, and includes circuitry for enabling read, program, and erase operations to one or more memory devices 106.” *Id.*, ¶ [0031]; *see also id.*, claim 16, (reciting a memory controller comprising “a controller operatively coupled” to “a first interface adapted for coupling to one or more non-volatile memory devices”) (all emphasis added).

279. The MCU is also coupled to the IO interface. Cornwell teaches that,

“[i]n some implementations, the memory controller 104 recognizes control, address, and data signals transmitted on bus 108 by the host processor 102.” *Id.*, ¶ [0021]; *see also id.*, ¶ [0029] (explaining that the “MCU firmware can translate signals from the host processor 102 into memory read and write operations.”). Indeed, Cornwell claims a memory controller comprising “a controller operatively coupled” to a “second interface adapted for coupling to a host processors, and configurable to receive a request from the host processor.” *Id.*, claim 16.

280. Annotated Figure 2, shown below, illustrates the **interface controller** (“MCU 212”) coupled to a **memory device interface** (“IO Interface 206”) and an **input/output device interface** (“buffer 202”). A POSITA would have understood that buffer 202 is an input/output interface that is used to buffer, or hold, input and output data that is transmitted between the Memory Controller 104 and the Host Processor 102.



281. For at least these reasons, it is my opinion that Cornwell discloses “an interface controller coupled to a memory device interface and an input/output (IO)

device interface.”

c. [Claim 59b]: “a memory store;”

282. It is my opinion that Cornwell discloses this claim element. Cornwell discloses a memory store comprising “one or more non-volatile memory devices 106,” which “can be discrete chips, chipsets and or memory modules.” Ex-1005, ¶¶ [0018], [0020]. The disclosed memory devices may “include but are not limited to: NAND and/or NOR flash media, read-only memory (ROM), erasable, programmable ROM (EPROM), electrically-erasable, programmable ROM (EEPROM), Ferroelectric RAM (FeRAM), magnetoresistive RAM (MRAM), non-volatile static RAM (nvSRAM), and any other memory device that does not need its memory contents periodically refreshed and/or can retain memory without power.” *Id.*, ¶ [0020].

283. The **memory store**, comprising a plurality of memory devices 106, is illustrated in annotated Figure 1 below:

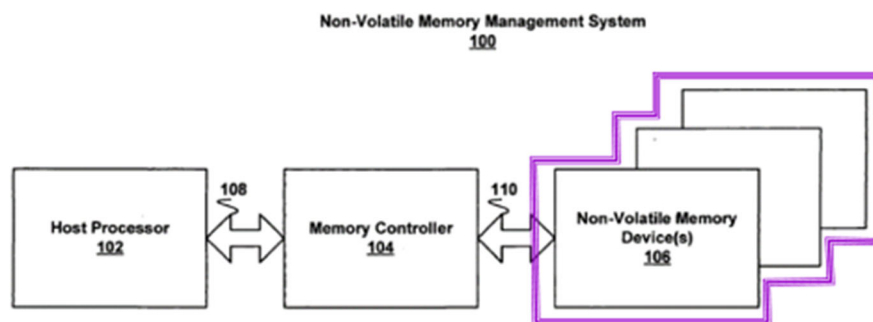


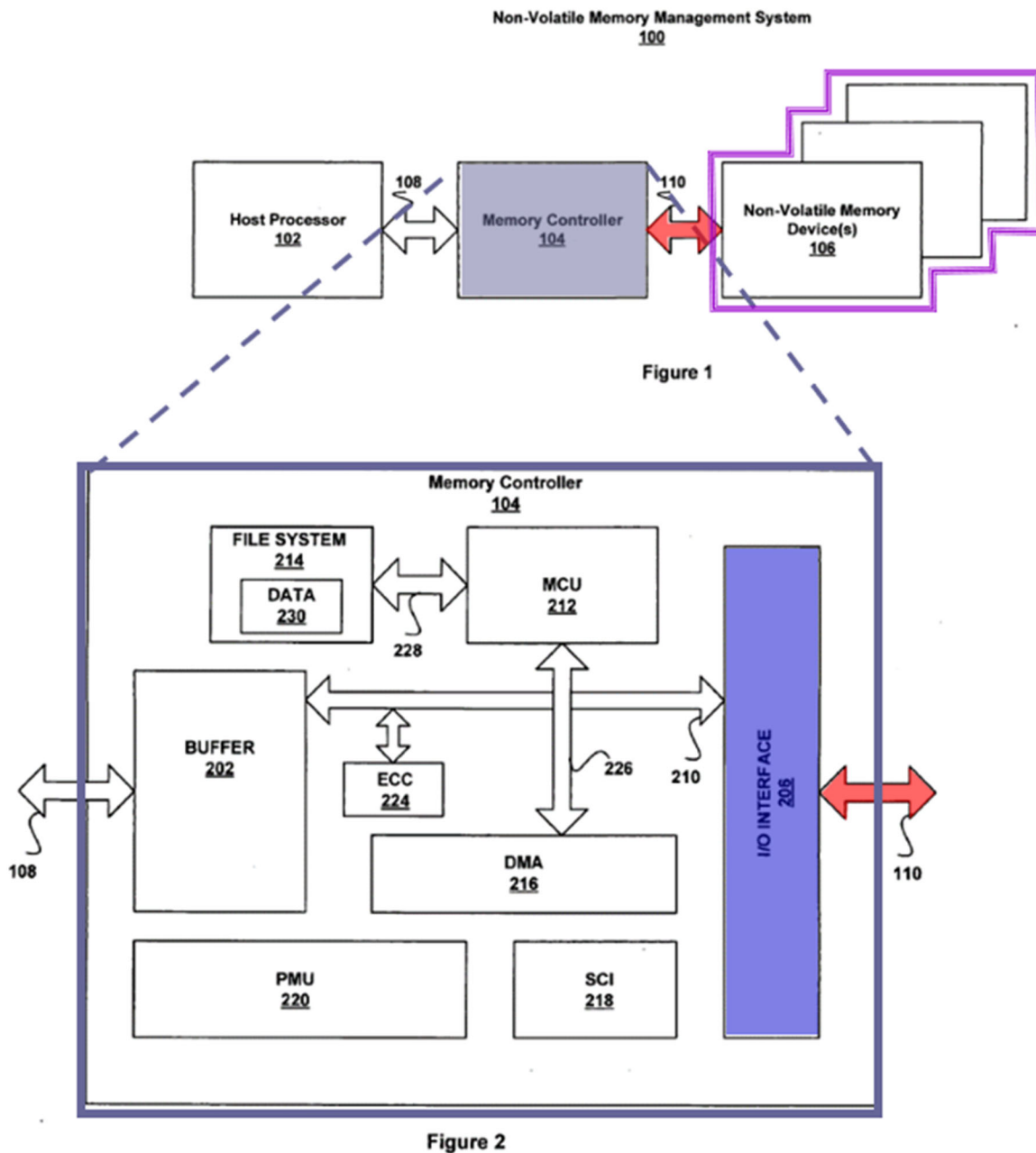
Figure 1

284. For at least these reasons, it is my opinion that Cornwell discloses “a memory store.”

d. [Claim 59c]: “wherein the memory device interface is directly coupled to the memory store;”

285. It is my opinion that Cornwell discloses this limitation. As explained above, Cornwell discloses a memory device interface. *Supra* § X.A.10.b. Cornwell also discloses that the memory device interface (“IO Interface 206”) is directly coupled to the memory store. For example, Cornwell teaches that the “I/O interface provides connectivity to the memory devices 106 through I/O bus 110.” Ex-1005, ¶ [0031]. Likewise, Cornwell explains that “[d]uring operation, one or more *memory devices 106 receive signals from the memory controller 104 over Input/Output (I/O) bus 110*, which enables the memory devices 106 to perform memory access requests (e.g., read or write operations).” *Id.*, ¶ [0024].

286. Annotated Figures 1 and 2, shown below, demonstrate that the **memory device interface** (“IO Interface 206”) is directly coupled to the **memory store** via **bus 110**:



287. For at least these reasons, it is my opinion that Cornwell discloses “wherein the memory device interface is directly coupled to the memory store.”

- e. [Claim 59d]: “said interface controller disposed to perform a memory transaction by addressing a first

memory location in the memory store,”

288. It is my opinion that Cornwell discloses this claim element. Cornwell discloses a memory store comprising “one or more non-volatile memory devices 106,” which “can be discrete chips, chipsets and or memory.” Ex-1005, ¶¶ [0018], [0020]. The disclosed memory devices may “include but are not limited to: NAND and/or NOR flash media, read-only memory (ROM), erasable, programmable ROM (EPROM), electrically-erasable, programmable ROM (EEPROM), Ferroelectric RAM (FeRAM), magnetoresistive RAM (MRAM), non-volatile static RAM (nvSRAM), and any other memory device that does not need its memory contents periodically refreshed and/or can retain memory without power.” *Id.*, ¶ [0020].

289. The **memory store**, comprising a plurality of memory devices 106, is illustrated in annotated Figure 1 below:

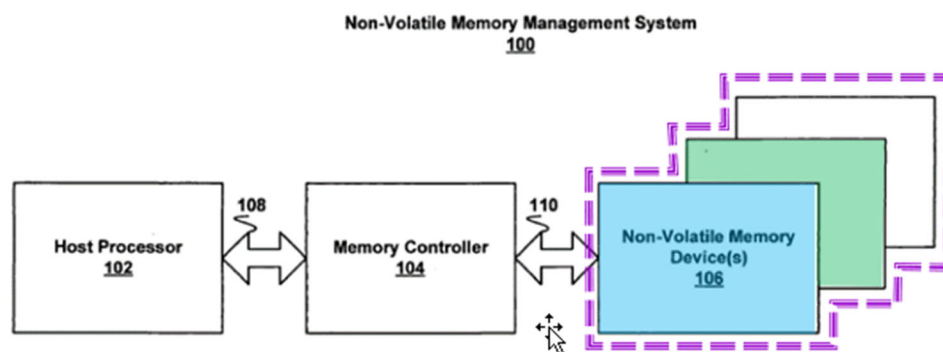


Figure 1

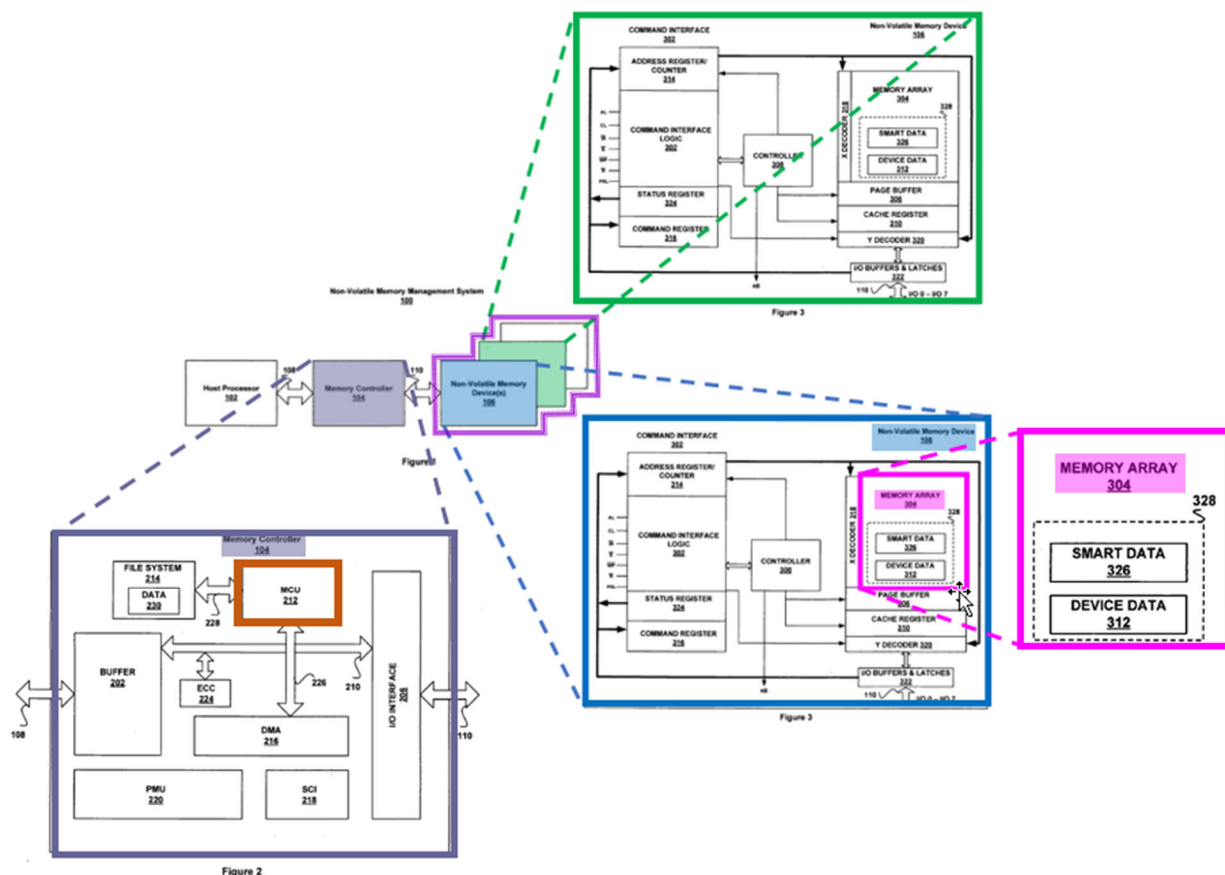
290. Cornwell also teaches performing a memory transaction comprising addressing a first memory location in a first memory device. Cornwell explains that

“the memory controller 104 recognizes control, **address**, and data signals transmitted . . . by the host processor 102” and “translates the control, **address** and data **signals into memory access requests** on memory devices 106.” *Id.*, ¶ [0021] (emphasis added); *see also id.* (explaining that an IDE/ATA bus may translate “control, **address** and data signals into memory access requests”); *id.*, ¶ [0035] (noting that the memory device has a decoder that “decodes input **addresses**”); *id.*, ¶ [0037] (“The address register/counter 314 is configured to receive **addresses** from the memory controller 104 via I/O bus 110.”) (all emphasis added). A POSITA would readily understand that an address for accessing data on a memory device would include information addressing the location in the memory device where that data is located.

291. Cornwell also discloses that the interface controller (MCU 212) is disposed to perform these functions. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029]. The file system also stores data that can “include an electronic signature or serial number for identifying the memory device

106 or its manufacturer, the block size of the memory controller 104, an identification of bad blocks, chip interleave depth, etc.” *Id.*

292. Annotated Figures 1 and 3, below, illustrate the **interface controller** (“MCU 212”) disposed to perform a memory transaction by addressing a **first memory location** (memory array 304 in a **first memory device 106**) in the **memory store**.



293. For at least these reasons, it is my opinion that Cornwell discloses “said

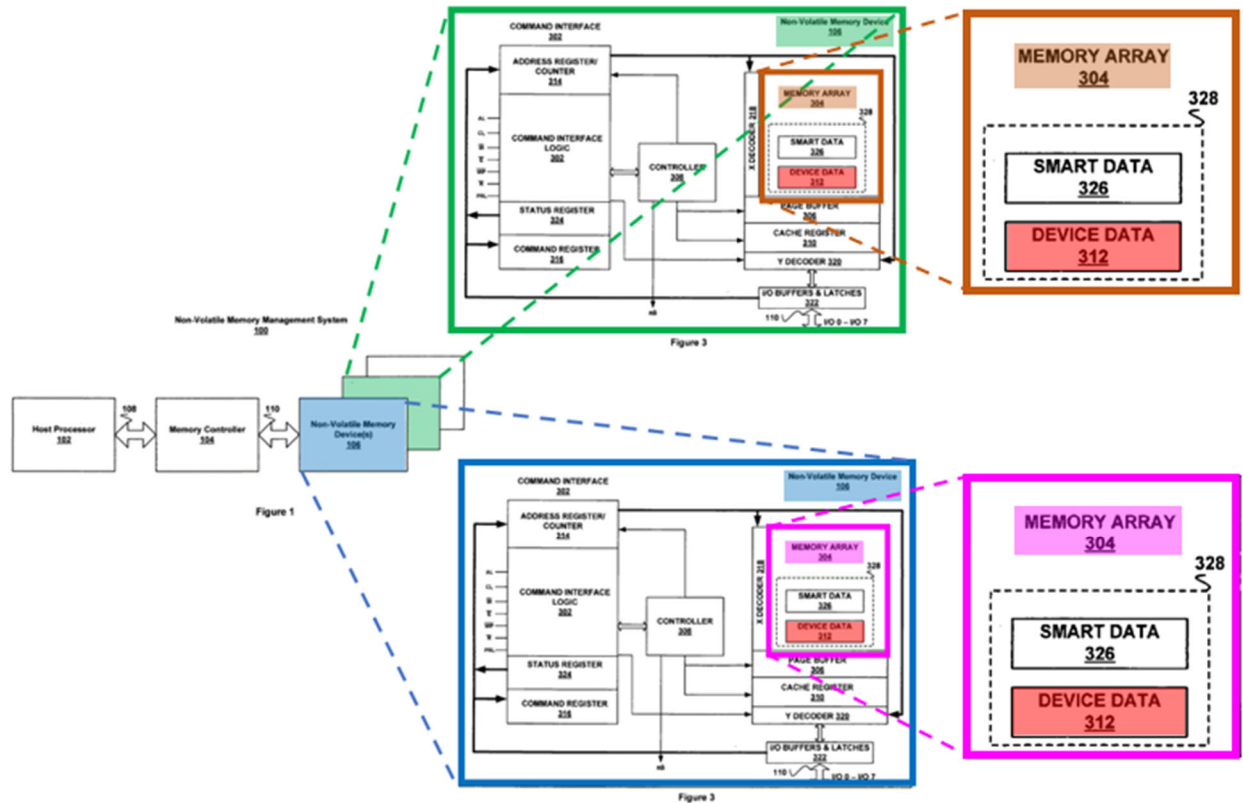
interface controller disposed to perform a memory transaction by addressing a first memory location in the memory store.”

f. [Claim 59e]: “said first memory location and a second memory location respectively associated with a first device profile and a second device profile;”

294. It is my opinion that Cornwell discloses this claim element. Cornwell discloses that each memory device 106 may include “device data 312 [that] can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. The device data “can include an electronic signature or serial number that includes a manufacturer code and/or device code,” as well as information such “as device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.* Accordingly, Cornwell’s collective set of “device data” comprises a “device profile.”²⁰

²⁰ See *supra* note 5 (explaining characteristics of device profiles).

295. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of device data 312) associated with both the **first memory location** and the **second memory location**:



296. For at least these reasons, Cornwell discloses that “said first memory location and a second memory location respectively associated with a first device profile and a second device profile.”

- g. [Claim 59f]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”**

297. It is my opinion that Cornwell renders this claim element obvious in

view of a POSITA's knowledge. As explained above, Cornwell teaches a "device profile" in the form of a collection of "device data 312." *See supra* § X.A.10.f. The device profile "can be used to identify the memory device 106 and its manufacturer." Ex-1005, ¶ [0042]. Information comprising the device profile may include "device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as "attributes")." *Id.*

298. A POSITA would have also understood that the device profiles taught in Cornwell are optimal for a data type subject to the memory transaction. For example, as explained in the preceding paragraph, Cornwell teaches that the device profile identifies whether the memory is NAND or NOR. *Id.* At the time Cornwell was filed, a POSITA would have readily understood that NOR provides random access (i.e., access to a "random data type") and NAND provides sequential access (i.e., access to a sequential data type").²¹ Accordingly, it would have been obvious

²¹ *See supra* note 6.

to a POSITA, based on Cornwell alone, that a profile for NOR memory would have been optimal for a random data type subject to a memory transaction, and that a profile for NAND memory would have been optimal for a sequential data type subject to a memory transaction.

299. Similarly, Cornwell teaches “block defining” as another way of ensuring that a device profile is optimal for a data type subject to the memory transactions. *See id.*, ¶¶ [0048]-[0050]. For example, Cornwell teaches that “[m]emory access efficiency can be improved by matching the average size of files to be stored in the flash media *to the block size of the flash media.*” *Id.*, ¶ [0048]; *see also id.*, ¶ [0050] (explaining that “it may be advantageous to *tailor the size of files* such that the *average file size is proportional to the block size.*”) (all emphasis added). As noted above, block size is one of the device profile attributes taught by Cornwell. The district court’s claim construction order states that such block size comparison information is another example of a device profile that is optimal for a data type subject to the memory transaction.²²

²² As noted above, the district court’s claim construction order states that examples of optimal device profiles are “(1) a device profile that includes a block size attribute value that is the largest available from the other device profiles when the data type

300. Moreover, Cornwell teaches that the data type comprises one of a random data type or a sequential data type, explaining that “[i]n some implementations a ***random [data type] read command*** may be issued first, followed by a page [***sequential data type***] read command.” *Id.*, ¶ [0040] (emphasis added). Thus, by matching block size to file size, and teaching that files contain random or sequential data types, Cornwell further teaches first device profile is optimal for a data type subject to the memory transaction.

301. For at least these reasons, it is my opinion that Cornwell in view of a POSITA’s knowledge renders obvious “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- h. [Claim 59g]: “said interface controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

302. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the memory device “generally includes a command interface 302, a

is sequential,” and “(2) a device profile that includes a block size attribute value that is the smallest available from the other device profiles” when the data type is random.” *Supra* note 7 (citing Ex-1007 at 44).

memory array 304 and a controller 308.” Ex-1005, ¶ [0034]. Cornwell further identifies, for example, a variety of commands that are “translated into signals which can be used by a controller and decoding logic in the memory device 106 to access a memory array.” *Id.*, ¶ [0027]. “For example, when the host processor 102 makes a read request, the “Read Sectors” opcode (20 h) is transmitted to the memory controller 104, together with address and control signals for accessing the sector(s).” *Id.*; *see also id.*, ¶ [0035] (describing a decoder that “decodes signals from the command interface logic 302 for reading or writing data into the memory line”); *id.*, ¶ [0038] (explaining that the controller receives “one or more input addresses and command inputs, which are used by the controller 308 in combination with the control signals from the command interface logic 302 to carry out read and write operations on memory array 304”). Likewise, Cornwell teaches an “‘identify’ command that returns a chip ID,” which can then be used “to look-up information about the memory devices 106.” *Id.*, ¶ [0077]. As a result, Cornwell discloses identifying command details for causing the memory transaction to be performed, wherein the command details comprise the first memory device.

303. Examples of such commands are provided in Table 1 of Cornwell, reproduced below:

TABLE I

Examples of Standard ATA-6 Commands	
Opcode	Command
10h	Recalibrate
20h	Read Sectors
30h	Write Sectors
40h	Read Verify
B0h	SMART
C8h	Read DMA
CAh	Write DMA
E0h	Standby Immediate
E2h	Standby
E7h	Flush Cache
ECh	Identify
EFh	Set Features

304. The **command interface logic 302**, which identifies command details for causing the memory transaction to be performed, is also illustrated in annotated Figure 3 below:

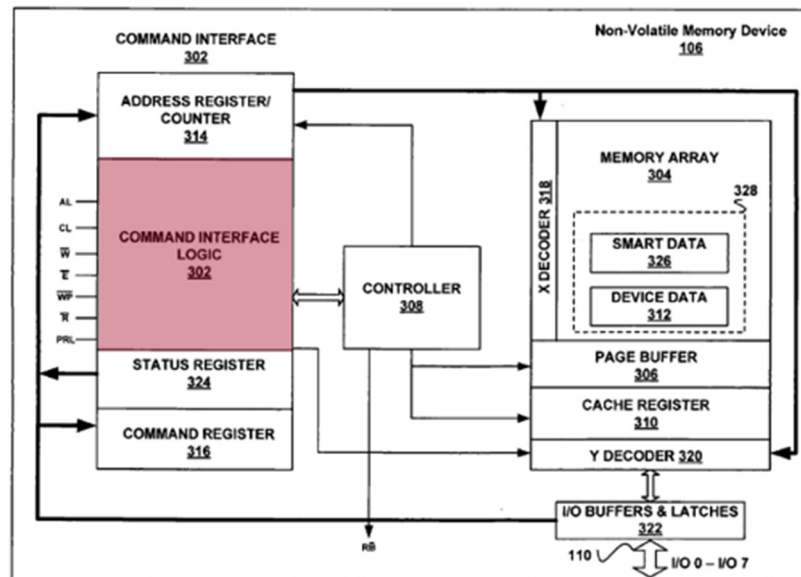


Figure 3

305. Cornwell also discloses that the interface controller (MCU 212) performs these functions. “The MCU 212 translates IDE/ATA commands into data

and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029].

306. For at least these reasons, it is my opinion that Cornwell discloses “said interface controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- i. **[Claim 59h]: “said device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

307. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. Cornwell teaches that “the device data 312,” which collectively comprises the device profile, “can include an electronic signature or serial number that includes a manufacturer code and/or device code.” Ex-1005, ¶ [0042]. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer.” *Id.* For example, device data 312 can include the following:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Potential
differences
between first and
second device

Id. (highlighting and underlining added). Cornwell expressly states that this information “is associated with *attributes*, properties or characteristics of the memory device 106 (*collectively, referred to herein as “attributes”*).” *Id.*

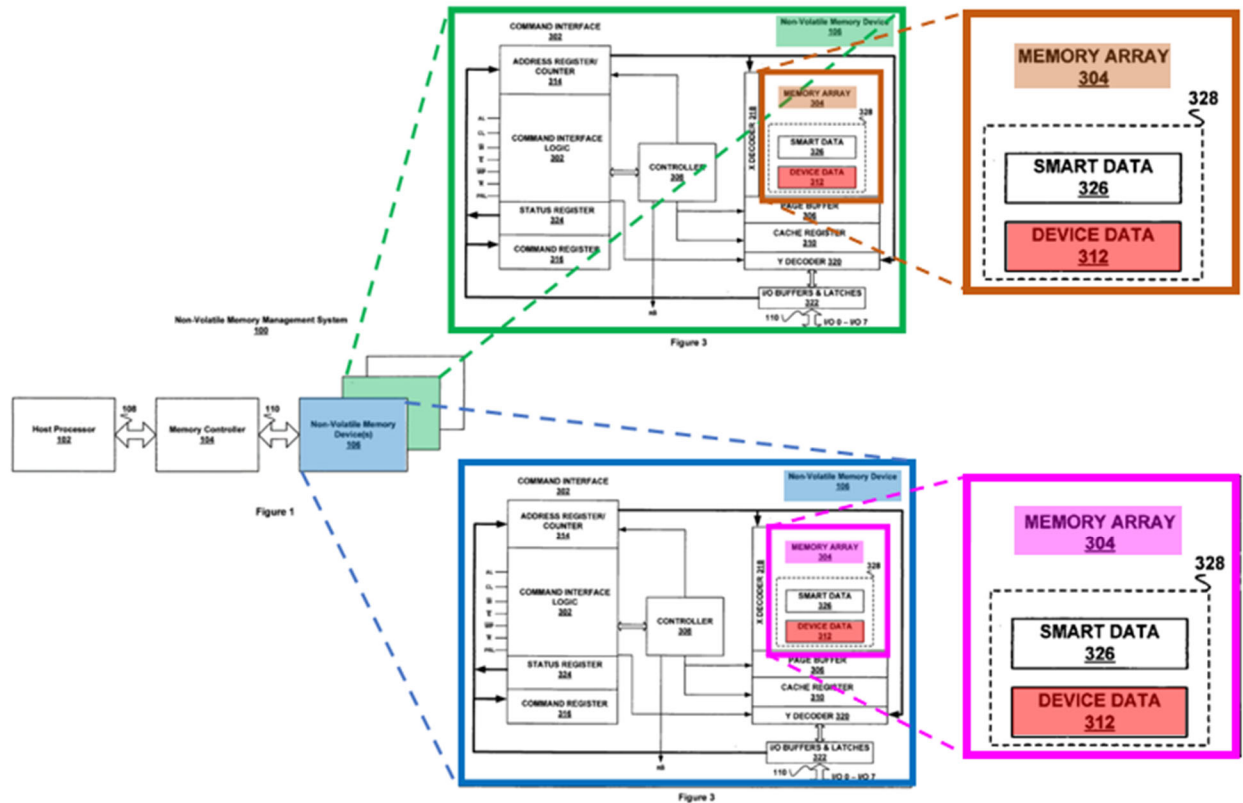
308. Cornwell also describes that “differences exist” between memory profiles. For example, Cornwell explains that different memory devices 106 may have different densities depending on whether they utilize single-layer cell technology or multi-layer cell technology, teaching that “syncing with memory devices 106 that include multi-level (MLC) technology can be performed at a different frequency than with memory devices 106 that include single-level cell technology (SLC).” *Id.*, [0078]. This is one example of a difference between profiles.

309. In my opinion, Cornwell having a “difference exist” in these profiles

would also be obvious in view of Cornwell and a POSITA's knowledge. As demonstrated above, Cornwell identifies a large number of potential differences in device attributes (describing possible differences in device type, device density, spare area, block size, etc.). *Id.*, ¶ [0042]. A difference in any of these attributes, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). As a result, I believe that Cornwell renders obvious having a "difference exist" between memory profiles.

310. Similarly, Cornwell explains that memory devices may have certain levels of bad blocks, which may increase over the lifetime of the device and which are identified so that the host system can avoid writing to them. *Id.*, ¶¶ [0054]-[0056]; *see also id.*, ¶ [0048] (explaining that flash memory "is available in a variety of block sizes" and memory access efficiency can be improved by "matching the average size of the files to be stored in the flash media to the block size of the flash media"); ¶ [0060] (describing different interleave levels for chips). As noted above, bad block identification is yet another example of an attribute described by Cornwell. *Id.*, ¶ [0042]. Accordingly, a difference in the number of bad blocks, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). For this additional reason, I believe that Cornwell renders obvious having a "difference exist" between memory profiles.

311. Annotated Figures 1 and 3, below, illustrate the **device profile** (i.e., the collection of attributes comprising device data 312) associated with both the **first memory location** and the **second memory location**:



312. For at least these reasons it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious "said device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles."

- j. [Claim 59i]: "said interface controller obtaining the first set of attributes after identifying the command

details;”

313. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that attributes comprising device data 312 can be obtained or “transmitted to the memory controller 104 via the I/O bus 110 *in response to a read command* issued by the memory controller.” Ex-1005, ¶ [0043]. Cornwell’s teaching that the attributes are transmitted “in response to a command” confirms that they are obtained after the command details are identified.

314. Cornwell also discloses that the interface controller (MCU 212) performs these functions. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029].

315. For at least these reasons, it is my opinion that Cornwell discloses “said interface controller obtaining the first set of attributes after identifying the command details.”

- k. **[Claim 59j]: “and said addressing of said first memory location includes using said attributes from said first device profile; and”**

316. It is my opinion that Cornwell discloses this claim element. As explained above, Cornwell teaches that the first and second device profiles (i.e., collectively comprising device data 312) include a variety of attributes. *See supra* § X.A.10.f; Ex-1005, ¶ [0042] (describing attributes). Cornwell also teaches using the attributes from the first and second device profiles, explaining that “device data 312 can be transmitted to the memory controller 104 . . . *in response to a read command* issued by the by the memory controller” and “*used by the memory controller 104 and/or host system* to perform various memory management tasks.” Ex-1005, ¶ [0043] (emphasis added); *see also id.*, ¶ [0042] (explaining that “device data 312 can be used to identify the memory device 106 and its manufacturer”); *id.*, ¶ [0046] (describing a memory management policy that “addresses how read and write operations should be performed”); *id.*, ¶ [0058] (explaining that the host processor “can request and receive memory device information (e.g., signatures, block size, interleave date, etc.) for one or more memory devices 106 over a standard IDE/ATA bus”).

317. For at least these reasons, it is my opinion that Cornwell discloses “said addressing of said first memory location includes using said attributes from said first device profile.”

- I. **[Claim 59k]: “said addressing of said first memory location includes selecting a transfer size for the**

memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

318. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches a memory transaction in which “the controller 308, *programs the page* into the memory array 304” Ex-1005, ¶ [0039] (emphasis added). As Cornwell further explains, a page has a specific “page size (1k, 2k, etc.).” *Id.*, ¶ [0042]; *see also, e.g., id.*, ¶ [0048] (“Memory access efficiency can be improved by *matching the average size of files* to be stored in the flash media *to the block size of the flash media.*”); *id.*, ¶ [0050] “[I]t may be advantageous to *tailor the size of files* such that the average file size is *proportional to the block size.*”) (all emphasis added). In this manner, Cornwell teaches selecting a transfer size for the memory transaction.

319. Cornwell further discloses that the page size is a function of the device density of the memory transactions. *Id.*, ¶ [0042]. For example, Cornwell teaches varying “device density (e.g., 512 Mb, 1Gb, 2Gb, etc.).” *Id.*; *see also supra* § X.A.10.f. A POSITA would have understood that page size is the length of a row (number of columns) in the memory. A POSITA would have further understood that the length of a row, again the number of columns in a row, varies with the device density since the device density is the number of rows multiplied by the number of

columns in a memory array multiplied by the number of arrays used in the memory chip.²³ For example, a page size of 2k (i.e., there are 2,048 columns in a row), with 8k rows and 1k arrays would result in device density of 16Gb.

320. Cornwell also demonstrates that the transfer size is selected by the controller. As seen in Figure 1 of Cornwell, the host 102 communicates via bus 108 with the memory controller 104 in a format consistent with the IDE/ATA bus standard and its command, including the data size information (e.g., the number of logical sectors to be transferred), while the memory controller 104 communicates with the memory store via the I/O bus 110, using the signals required by the chips used in the memory store and a transfer size, which is selected based on the data size information. As a result, the memory controller selects the transfer size for the memory transaction in response to a request from a host.

321. For instance, Cornwell teaches an example in which a media player sends a request to an operating system on a host. *Id.*, ¶ [0025]. This “request is received by the operating system, which formats the request into IDE/ATA signals, which are transmitted to the memory controller 104 on the IDE/ATA bus 108 by the

²³ See *supra* note 8.

host processor 102.” *Id.*²⁴ The fact that host sends the requests to the memory controller—which must then communicate with the memory store—ensures that the memory controller selects the transfer size for the memory transactions, where the transfer size is a function of the data size of the memory transaction and the first set of attributes to retrieve the requested media file.

322. For at least these reasons, it is my opinion that Cornwell discloses “said addressing of said first memory location includes selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

11. Claim 71 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

323. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious Claim 71 of the ’190 patent.

- a. **[Claim 71]: “The memory controller of claim 59, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

²⁴ As Cornwell also explains, “ATA-6 is the latest version of the IDE/ATA standard, which was approved by the American National Standards Institute (ANSI) in 2001 under document NCITS 347-2001.” *Id.*, ¶ [0026].

324. It is my opinion that Cornwell renders obvious the memory controller of claim 59 for the reasons discussed above. *Supra* § X.A.10. It is also my opinion that Cornwell renders obvious the memory controller recited in claim 71.

325. As explained above, Cornwell teaches that the first and second device profiles (i.e., each set of device data 312) have shared attributes. *Supra* § X.A.10.f; Ex-1005, ¶ [0042]. For example, each device profile may have a “device type” attribute, “device density” attribute, “page size” attribute, etc. *Supra* § X.A.10.f; Ex-1005, ¶ [0042]. Further, as explained above, it is my opinion that having different attribute qualifiers for these shared attributes would have been obvious in view of Cornwell and a POSITA’s knowledge. *Supra* § X.A.10.i. For example, the first memory location could have a “NAND” qualifier for the “device type” attribute, while the second memory location could have “NOR” qualifier for the “device type” attribute. Ex-1005, ¶ [0042]. Likewise, the first memory location could have a “512 Mb” qualifier for the “device density” attribute, while the second memory location could have a “1Gb” qualifier for the “device density” attribute. *Id.* Cornwell discloses a variety of different attribute qualifiers for each shared attribute in the first and second device profiles, as illustrated below:

manufacturer code and/or device code. Chip data 312 can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

326. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “memory controller of claim 59, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

12. Claim 76 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

327. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious claim 76 of the ’190 patent.

- a. **[Claim 76]: “The memory controller of claim 71, wherein said different attribute qualifiers include different attribute values.”**

328. It is my opinion that Cornwell renders obvious the memory controller of claim 71 for the reasons discussed above. *Supra* § X.A.11. It is also my opinion

that Cornwell renders obvious the memory controller recited in claim 76.

329. As explained above, it is my opinion that having different attribute qualifiers would have been rendered obvious by Cornwell in view of a POSITA's knowledge. *Supra* § X.A.10.i. Cornwell also teaches that each attribute qualifier has an attribute value, which may be different. For example, the qualifiers associated with the "device density" attribute can have different values such as 512 Mb, 1 Gb, 2 Gb, etc. Ex-1005, ¶ [0042]. Likewise, the qualifiers associated with the "page size" attribute can have different values such as 1k, 2K, etc. *Id.*; *see also supra* § X.A.10.f. Accordingly, it is my opinion that would have been obvious in view of Cornwell and a POSITA's knowledge to have different attribute values for attribute qualifiers.

330. A variety of potential differences in attribute values are illustrated in the following excerpt from Cornwell:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

331. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “memory controller of claim 71, wherein said different attribute qualifiers include different attribute values.”

13. Claim 80 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

332. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious claim 80 of the ’190 patent.

a. [Claim 80 pre]: “An electronic storage device, comprising:”

333. To the extent the preamble is limiting, it is my opinion that Cornwell discloses this claim element. Cornwell teaches a “memory management system” that *“can be part of a host system. A host system can be any electronic or*

computing device that uses non-volatile memory, including but not limited to: flash drives, portable and desktop computers, clients, servers, consumer electronics, calculators, network appliances, media players/recorders, game consoles, mobile phones, email devices, personal digital assistants (PDAs), embedded devices, televisions, system-on-chip (SoC), set-top boxes, audio recorders, handheld data collection scanners, monitoring devices, etc.” Ex-1005, ¶ [0018] (emphasis added); *see also id.*, ¶¶ [0081]-[0088] (describing “hardware architecture for a host system 600 that includes the memory management system 100” and explaining that the disclosed implementations “can be realized in any device capable of presenting a user interface on a display device”). Cornwell’s disclosure of a host system thus teaches “an electronic storage device.”

334. Figure 1, shown below, illustrates an exemplary non-volatile memory management system, while Figure 6b, also shown below, illustrates the hardware architecture for a host system that includes the memory management system shown in Figure 1:

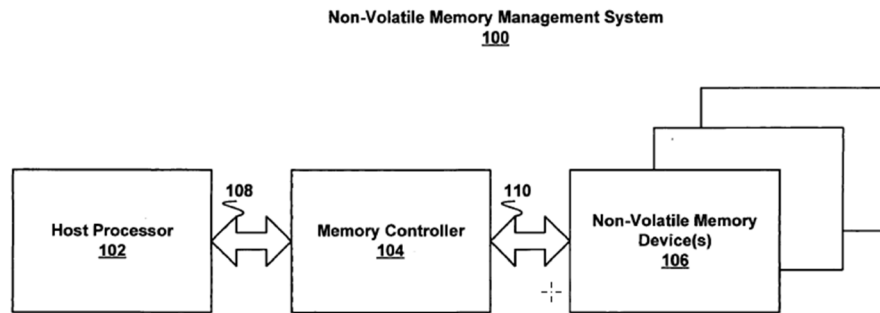


Figure 1

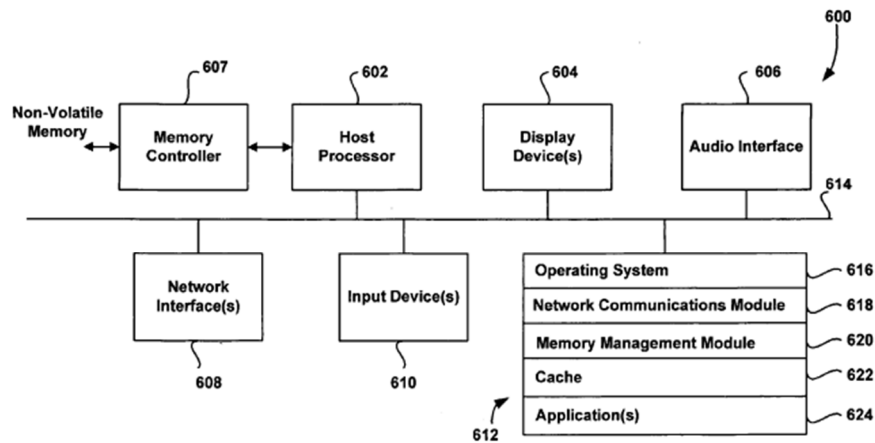


Figure 6B

335. For at least these reasons, it is my opinion that Cornwell discloses an “electronic storage device.”

- b. [Claim 80a]: “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile;”

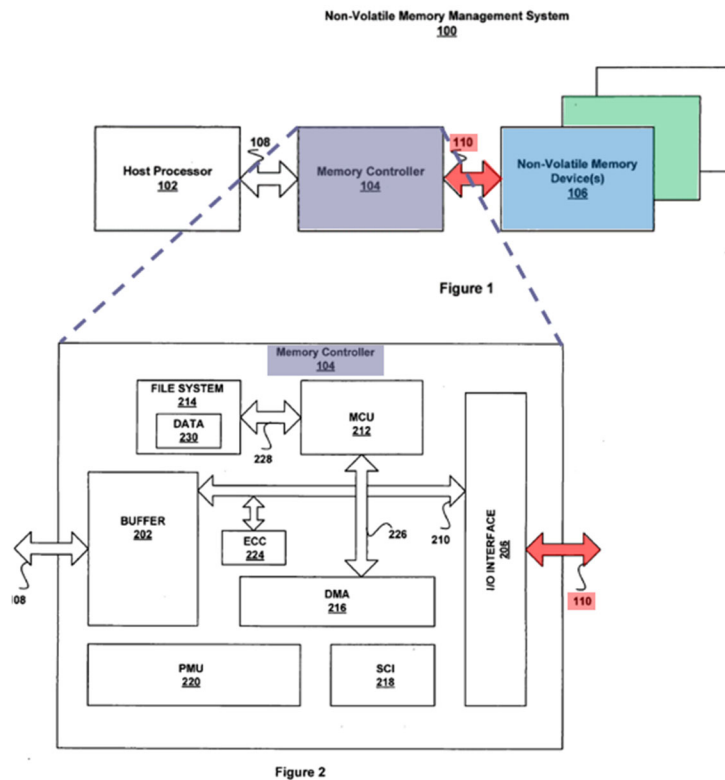
336. It is my opinion that Cornwell discloses this claim limitation. Cornwell teaches “[a] memory controller [that] is operatively *coupled to the one or more non-*

volatile memory devices and configurable to access the one or more non-volatile memory devices in accordance with a memory management policy.” Ex-1005, ¶ [0007] (emphasis added); *see also id.*, ¶ [0024] (“During operation, one or more memory devices 106 receive signals from the memory controller 104 over Input/Output (I/O) bus 110, which enables the memory devices 106 to perform memory access requests (e.g., read or write operations).” More specifically, “[t]he I/O interface 206 [of memory controller 106] provides connectivity to the memory devices 106 through I/O bus 110, and includes circuitry for enabling read, program, and erase operations to one or more memory devices 106.” *Id.*, ¶ [0031]; *see also id.*, claim 16, (reciting a memory controller comprising “a controller operatively coupled” to “a first interface adapted for coupling to one or more non-volatile memory devices”) (emphasis added).

337. Cornwell also teaches that the memory controller is disposed to address a first memory location and a second memory location. Cornwell explains that “the memory controller 104 recognizes control, *address*, and data signals transmitted . . . by the host processor 102” and “translates the control, *address* and data *signals into memory access requests* on memory devices 106.” *Id.*, ¶ [0021] (emphasis added); *see also id.* (explaining that an IDE/ATA bus may translate “control, *address* and data signals into memory access requests”); *id.*, ¶ [0035] (noting that the memory

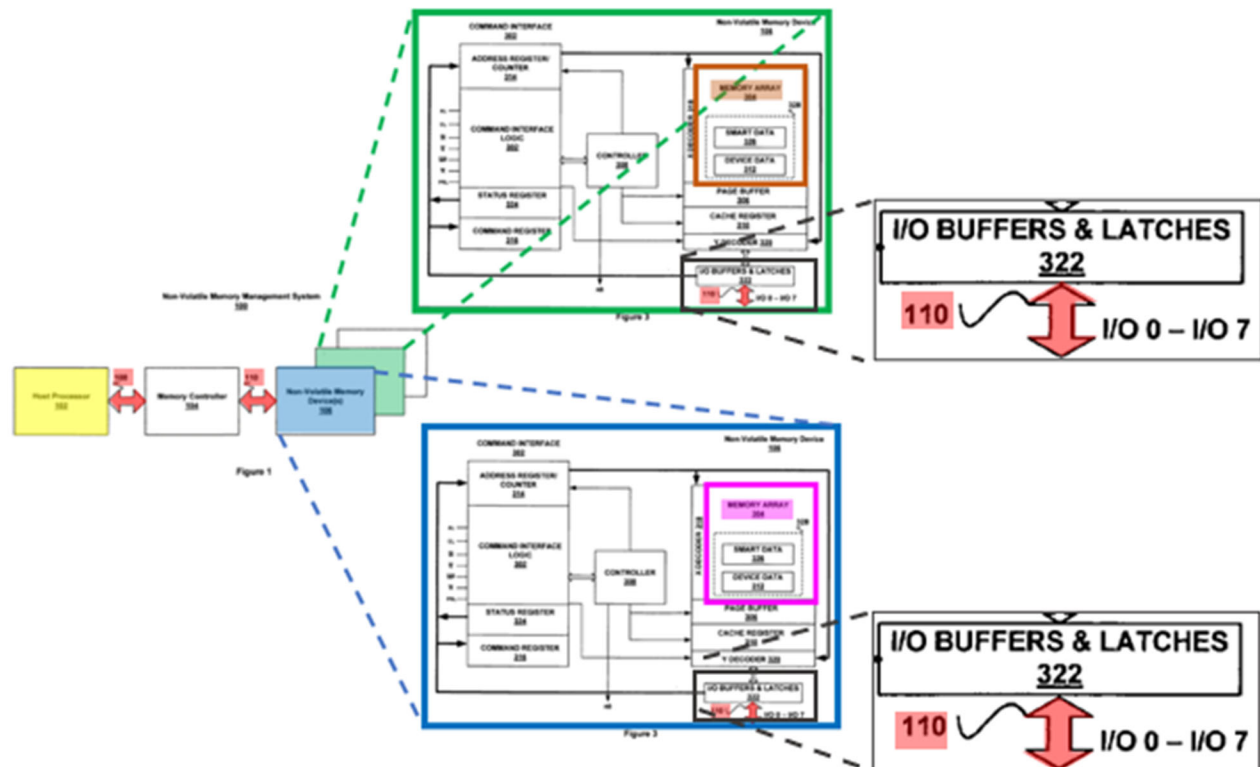
device has a decoder that “decodes input *addresses*”); *id.*, ¶ [0037] (“The address register/counter 314 is configured to receive *addresses* from the memory controller 104 via I/O bus 110.”) (all emphasis added).

338. As illustrated below, annotated Figures 1 and 2 confirm that Cornwell’s **memory controller** (“memory controller 104”) is directly coupled to and disposed to address a **first memory location** and a **second memory location** via **bus 110**:



339. Annotated Figures 1 and 3, below, further demonstrate that Cornwell’s **memory controller** (“memory controller 104”) is directly coupled to and disposed

to address a **first memory location** and a **second memory location** via **bus 110**:

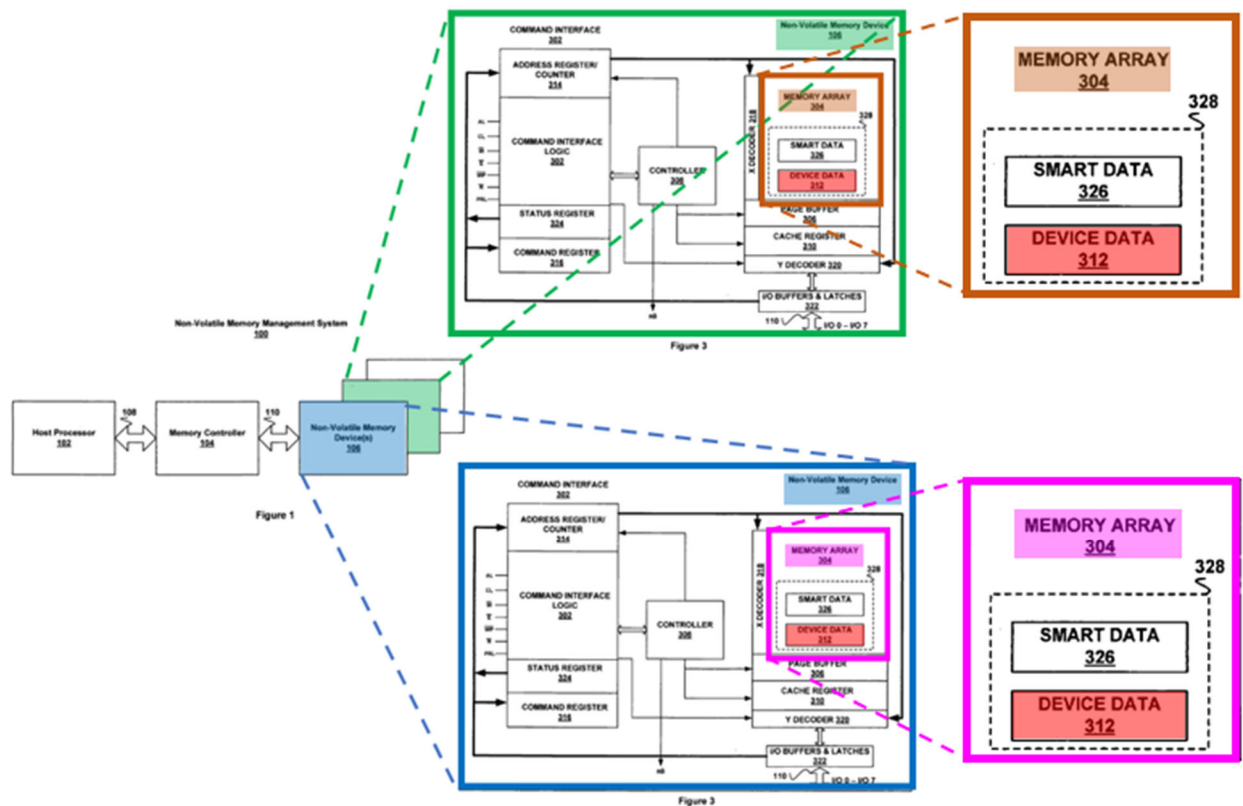


340. Cornwell also discloses that the first and second memory locations are respectively associated with a first device profile and a second device profile. Cornwell discloses that each memory device 106 may include “device data 312 [that] can be used to identify the memory device 106 and its manufacturer.” *Id.*, ¶ [0042]. The device data “can include an electronic signature or serial number that includes a manufacturer code and/or device code,” as well as information such “as device

type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., $\times 8$, $\times 16$, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.* Accordingly, Cornwell’s collective set of “device data” comprises a “device profile.”²⁵

341. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of device data 312) associated with both the **first memory location** and the **second memory location**:

²⁵ See *supra* note 5 (explaining characteristics of device profiles).



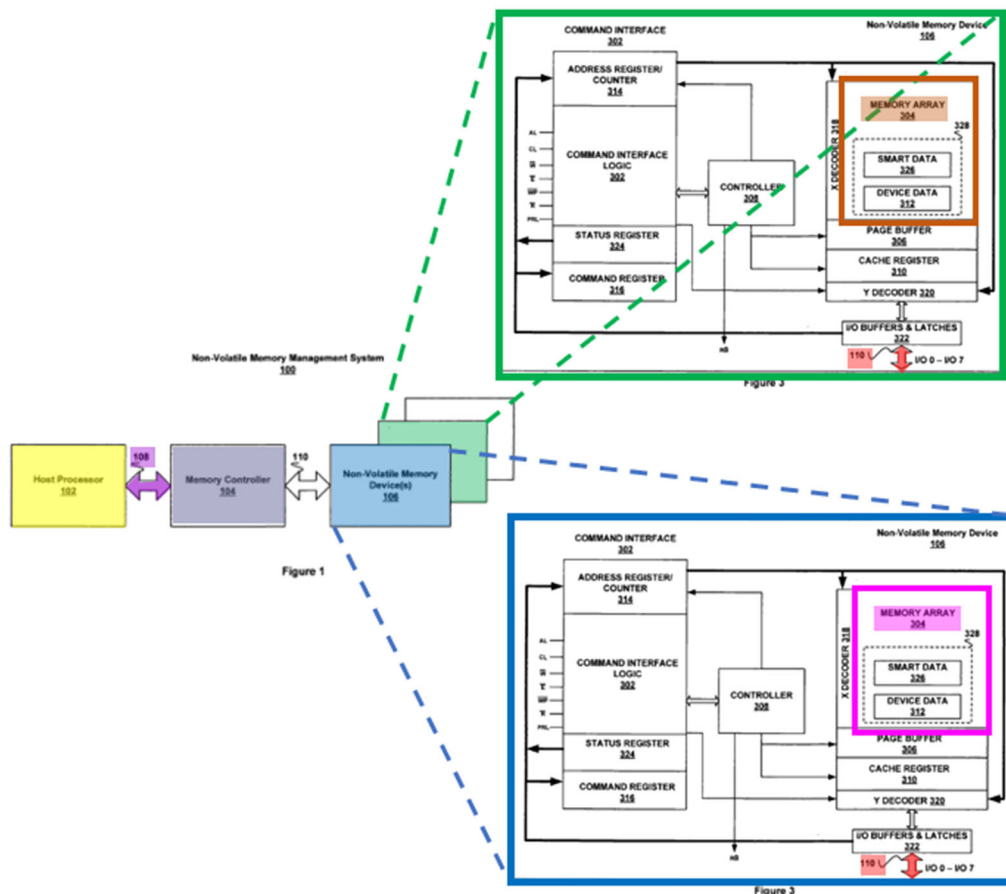
342. For at least these reasons, it is my opinion that Cornwell discloses “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile.”

- c. [Claim 80b]: “said memory controller directly coupled to a host as said memory controller addresses the first memory location or second memory location;”

343. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that “the memory controller 104 recognizes control, address, and data *signals transmitted on bus 108 by the host processor 102*” and

“*translates* the control, address and data signals *into memory access requests on memory devices 106.*” Ex-1005, ¶ [0021] (emphasis added); *see also id.*, ¶ [0025] (explaining that data requests “are transmitted to the memory controller 104 on the IDE/ATA *bus 108* by the host processor 102”).

344. Annotated Figures 1 and 3, below, further illustrate that the **host processor** (highlighted in yellow) is directly coupled, **via bus 108**, to the **memory controller 104** as it addresses the **first memory location** or **second memory location**:



345. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller directly coupled to a host as said memory controller addresses the first memory location or second memory location.”

- d. [Claim 80c]: “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”**

346. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. As explained above, Cornwell teaches a “device profile” in the form of a collection of “device data 312.” *See supra* § X.A.13.b. The device profile “can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. Information comprising the device profile may include “device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.*

347. A POSITA would have also understood that the device profiles taught in Cornwell are optimal for a data type subject to the memory transaction. For example, as explained in the preceding paragraph, Cornwell teaches that the device

profile identifies whether the memory is NAND or NOR. *Id.* At the time Cornwell was filed, a POSITA would have readily understood that NOR provides random access (i.e., access to a “random data type”) and NAND provides sequential access (i.e., access to a sequential data type”).²⁶ Accordingly, it would have been obvious to a POSITA, based on Cornwell alone, that a profile for NOR memory would have been optimal for a random data type subject to a memory transaction, and that a profile for NAND memory would have been optimal for a sequential data type subject to a memory transaction.

348. Similarly, Cornwell teaches “block defining” as another way of ensuring that a device profile is optimal for a data type subject to the memory transactions. *See id.*, ¶¶ [0048]-[0050]. For example, Cornwell teaches that “[m]emory access efficiency can be improved by matching the average size of files to be stored in the flash media *to the block size of the flash media.*” *Id.*, ¶ [0048]; *see also id.*, ¶ [0050] (explaining that “it may be advantageous to *tailor the size of files* such that the *average file size is proportional to the block size.*”) (all emphasis added). As noted above, block size is one of the device profile attributes taught by Cornwell. The district court’s claim construction order states that such block size

²⁶ *See supra* note 6.

comparison information is another example of a device profile that is optimal for a data type subject to the memory transaction.²⁷

349. Moreover, Cornwell teaches that the data type comprises one of a random data type or a sequential data type, explaining that “[i]n some implementations a **random [data type] read command** may be issued first, followed by a page [**sequential data type**] read command.” *Id.*, ¶ [0040] (emphasis added). Thus, by matching block size to file size, and teaching that files contain random or sequential data types, Cornwell further teaches first device profile is optimal for a data type subject to the memory transaction.

350. For at least these reasons, it is my opinion that Cornwell in view of a POSITA’s knowledge renders obvious “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises

²⁷ As noted above, the district court’s claim construction order states that examples of optimal device profiles are “(1) a device profile that includes a block size attribute value that is the largest available from the other device profiles when the data type is sequential,” and “(2) a device profile that includes a block size attribute value that is the smallest available from the other device profiles” when the data type is random.” *Supra* note 7 (citing Ex-1007 at 44).

one of a random data type or a sequential data type.”

- e. **[Claim 80d]: “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

351. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the memory device “generally includes a command interface 302, a memory array 304 and a controller 308.” Ex-1005, ¶ [0034]. Cornwell further identifies, for example, a variety of commands that are “translated into signals which can be used by a controller and decoding logic in the memory device 106 to access a memory array.” *Id.*, ¶ [0027]. “For example, when the host processor 102 makes a read request, the “Read Sectors” opcode (20 h) is transmitted to the memory controller 104, together with address and control signals for accessing the sector(s).” *Id.*; *see also id.*, ¶ [0035] (describing a decoder that “decodes signals from the command interface logic 302 for reading or writing data into the memory line”); *id.*, ¶ [0038] (explaining that the controller receives “one or more input addresses and command inputs, which are used by the controller 308 in combination with the control signals from the command interface logic 302 to carry out read and write operations on memory array 304”). Likewise, Cornwell teaches an ““identify” command that returns a chip ID,” which can then be used “to look-up information about the memory devices 106.” *Id.*, ¶ [0077]. As a result, Cornwell discloses

identifying command details for causing the memory transaction to be performed, wherein the command details comprise the first memory device.

352. Cornwell also discloses that the interface controller (MCU 212) within the memory controller performs these functions. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example, the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029].

353. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- f. [Claim 80e]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

354. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. Cornwell teaches that “the device data 312,” which collectively comprises the device profile, “can include an electronic signature or serial number that includes a manufacturer code and/or device code.” Ex-1005, ¶

[0042]. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer.” *Id.* For example, device data 312 can include the following:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Potential
differences
between first and
second device

Id. (highlighting and underlining added). Cornwell expressly states that this information “is associated with *attributes*, properties or characteristics of the memory device 106 (*collectively, referred to herein as “attributes”*).” *Id.*

355. Cornwell also describes that “differences exist” between memory profiles. For example, Cornwell explains that different memory devices 106 may have different densities depending on whether they utilize single-layer cell technology or multi-layer cell technology, teaching that “syncing with memory devices 106 that include multi-level (MLC) technology can be performed at a different frequency than with memory devices 106 that include single-level cell

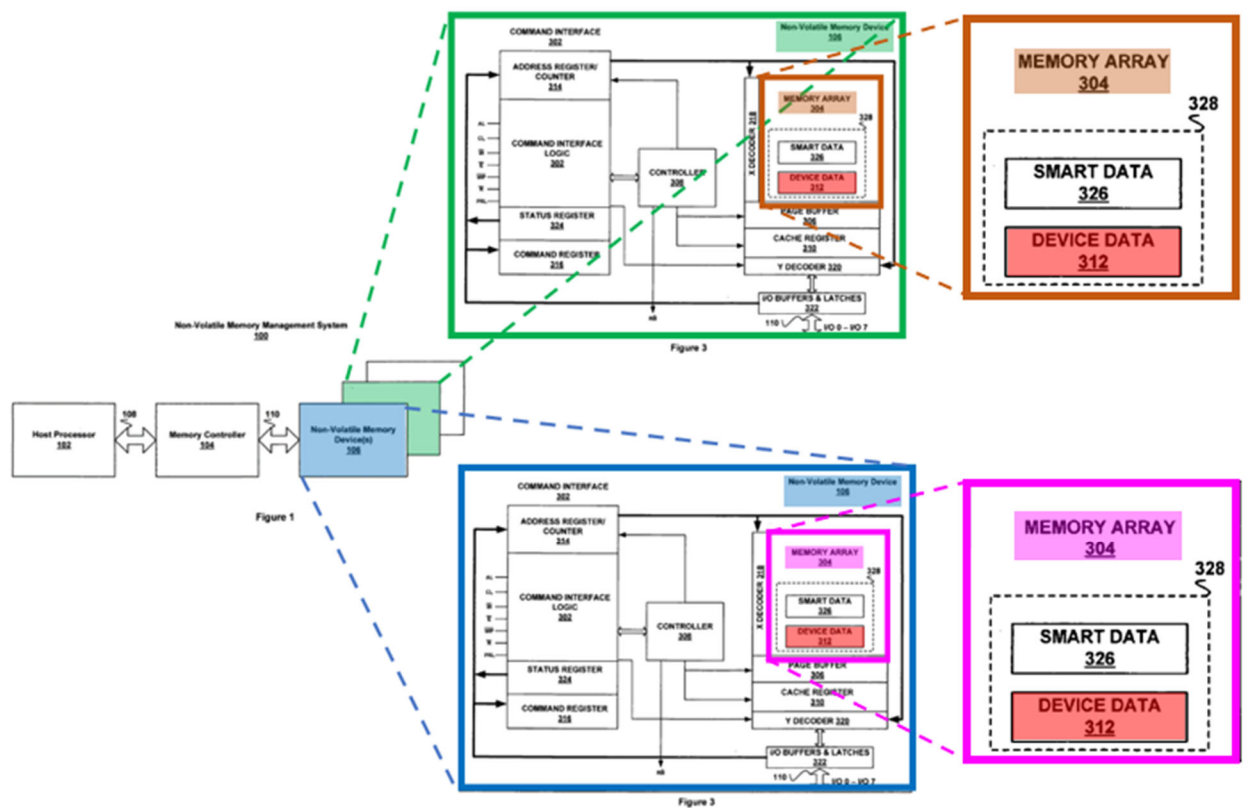
technology (SLC).” *Id.*, [0078]. This is one example of a difference between profiles.

356. In my opinion, having a “difference exist” in these profiles would also be obvious in view of Cornwell and a POSITA’s knowledge. As demonstrated above, Cornwell identifies a large number of potential differences in device attributes (describing differences in device type, device density, spare area, block size, etc.). *Id.*, ¶ [0042]. A difference in any of these attributes, as shown by Cornwell, would constitute a difference in the device profile (chip data 312). As a result, I believe that Cornwell renders obvious having a “difference exist” between memory profiles.

357. Similarly, Cornwell explains that memory devices may have certain levels of bad blocks, which may increase over the lifetime of the device and which are identified so that the host system can avoid writing to them. *Id.*, ¶¶ [0054]-[0056]; *see also id.*, ¶ [0048] (explaining that flash memory “is available in a variety of block sizes” and memory access efficiency can be improved by “matching the average size of the files to be stored in the flash media to the block size of the flash media”); ¶ [0060] (describing different interleave levels for chips). As noted above, bad block identification is yet another example of an attribute described by Cornwell. *Id.*, ¶ [0042]. Accordingly, a difference in the number of bad blocks, as shown by

Cornwell, would constitute a difference in the device profile (chip data 312). For this additional reason, I believe that Cornwell renders obvious having a “difference exist” between memory profiles.

358. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of attributes comprising device data 312) associated with both the **first memory location** and the **second memory location**:



359. For at least these reasons, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious “said first device profile representing a first set of attributes of said first memory location, and said second device profile

representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

g. [Claim 80f]: “said memory controller obtaining the first set of attributes after identifying the command details;”

360. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that attributes comprising device data 312 can be obtained or “transmitted to the memory controller 104 via the I/O bus 110 *in response to a read command* issued by the memory controller.” Ex-1005, ¶ [0043]. Cornwell’s teaching that the attributes are transmitted “in response to a command” confirms that they are obtained after the command details are identified.

361. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller obtaining the first set of attributes after identifying the command details.”

h. [Claim 80g]: “and said memory controller using attributes from said first and second device profiles; and”

362. It is my opinion that Cornwell discloses this claim element. As explained above, Cornwell teaches that the first and second device profiles (i.e., collectively comprising device data 312) include a variety of attributes. *See supra* § X.A.13.b; Ex-1005, ¶ [0042] (describing attributes). Cornwell also teaches using

the attributes from the first and second device profiles, explaining that “device data 312 can be transmitted to the memory controller 104 . . . *in response to a read command* issued by the by the memory controller” and “*used by the memory controller 104 and/or host system* to perform various memory management tasks.” Ex-1005, ¶ [0043] (emphasis added); *see also id.*, ¶ [0042] (explaining that “device data 312 can be used to identify the memory device 106 and its manufacturer”); *id.*, ¶ [0046] (describing a memory management policy that “addresses how read and write operations should be performed”); *id.*, ¶ [0058] (explaining that the host processor “can request and receive memory device information (e.g., signatures, block size, interleave date, etc.) for one or more memory devices 106 over a standard IDE/ATA bus”).

363. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller using attributes from said first and second device profiles.”

- i. **[Claim 80h]: “said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

364. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches a memory transaction in which “*the controller 308, programs the page* into the memory array 304” Ex-1005, ¶ [0039] (emphasis added). As Cornwell further explains, a page has a specific “page size (1k, 2k, etc.).”

Id., ¶ [0042]; *see also, e.g., id.*, ¶ [0048] (“Memory access efficiency can be improved by ***matching the average size of files*** to be stored in the flash media ***to the block size of the flash media.***”); *id.*, ¶ [0050] “[I]t may be advantageous to ***tailor the size of files*** such that the average file size is ***proportional to the block size.***”) (all emphasis added). In this manner, Cornwell teaches selecting a transfer size for the memory transaction.

365. Cornwell further discloses that the page size is a function of the device density of the memory transactions. *Id.*, ¶ [0042]. For example, Cornwell teaches varying “device density (e.g., 512 Mb, 1Gb, 2Gb, etc.).” *Id.*; *see also supra* § X.A.13.b. A POSITA would have understood that page size is the length of a row (number of columns) in the memory. A POSITA would have further understood that the length of a row, again the number of columns in a row, varies with the device density since the device density is the number of rows multiplied by the number of columns in a memory array multiplied by the number of arrays used in the memory chip.²⁸ For example, a page size of 2k (i.e., there are 2,048 columns in a row), with 8k rows and 1k arrays would result in device density of 16Gb.

366. Cornwell also demonstrates that the transfer size is selected by the

²⁸ *See supra* note 8.

controller. As seen in Figure 1 of Cornwell, the host 102 communicates via bus 108 with the memory controller 104 in a format consistent with the IDE/ATA bus standard and its command, including the data size information (e.g., the number of logical sectors to be transferred), while the memory controller 104 communicates with the memory store via the I/O bus 110, using the signals required by the chips used in the memory store and a transfer size, which is selected based on the data size information. As a result, the memory controller selects the transfer size for the memory transaction in response to a request from a host.

367. For instance, Cornwell teaches an example in which a media player sends a request to an operating system on a host. *Id.*, ¶ [0025]. This “request is received by the operating system, which formats the request into IDE/ATA signals, which are transmitted to the memory controller 104 on the IDE/ATA bus 108 by the host processor 102.” *Id.*²⁹ The fact that host sends the requests to the memory controller—which must then communicate with the memory store—ensures that the memory controller selects the transfer size for the memory transactions, where the

²⁹ As Cornwell also explains, “ATA-6 is the latest version of the IDE/ATA standard, which was approved by the American National Standards Institute (ANSI) in 2001 under document NCITS 347-2001.” *Id.*, ¶ [0026].

transfer size is a function of the data size of the memory transaction and the first set of attributes to retrieve the requested media file.

368. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

14. Claim 84 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

369. As explained below, it is my opinion that Cornwell, in view of a POSITA’s knowledge, renders obvious claim 84 of the ’190 patent.

- a. **[Claim 84]: “The electronic storage device of claim 80, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

370. It is my opinion that Cornwell renders obvious the electronic storage device of claim 80 for the reasons discussed above. *Supra* § X.A.13. It is also my opinion that Cornwell renders obvious the electronic storage device recited in claim 84.

371. As explained above, Cornwell teaches that the first and second device profiles (i.e., each set of device data 312) have shared attributes. *Supra* § X.A.13.b; Ex-1005, ¶ [0042]. For example, each device profile may have a “device type”

attribute, “device density” attribute, “page size” attribute, etc. *Id.*; ¶ [0042]. Further, as explained above, it is my opinion that having different attribute qualifiers for these shared attributes would have been obvious in view of Cornwell and a POSITA’s knowledge. *Supra* § X.A.13.f. For example, the first memory location could have a “NAND” qualifier for the “device type” attribute, while the second memory location could have “NOR” qualifier for the “device type” attribute. Ex-1005, ¶ [0042]. Likewise, the first memory location could have a “512 Mb” qualifier for the “device density” attribute, while the second memory location could have a “1Gb” qualifier for the “device density” attribute. *Id.* Cornwell discloses a variety of different attribute qualifiers for each shared attribute in the first and second device profiles, as illustrated below:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

372. For at least these reasons, it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious "[t]he electronic storage device of claim 80, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles."

15. Claim 85 Is Rendered Obvious by Cornwell in View of a POSITA's Knowledge

373. As explained below, it is my opinion that Cornwell, in view of a POSITA's knowledge, renders obvious claim 85 of the '190 patent.

- a. [Claim 85]: "The electronic storage device of claim 84, wherein said different attribute qualifiers include different attribute values."**

374. It is my opinion that Cornwell renders obvious the electronic storage device of claim 84 for the reasons discussed above. *Supra* § X.A.14. It is also my opinion that Cornwell renders obvious the electronic storage device recited in claim 85.

375. As explained above, it is my opinion that having different attribute qualifiers would have been rendered obvious by Cornwell in view of a POSITA's knowledge. *Supra* § X.A.13.f. Cornwell also teaches that each attribute qualifier has an attribute value, which may be different. For example, the qualifiers associated with the "device density" attribute can have different values such as 512 Mb, 1 Gb, 2 Gb, etc. Ex-1005, ¶ [0042]. Likewise, the qualifiers associated with the "page

size” attribute can have different values such as 1k, 2K, etc. *Id.*; *see also supra* § X.A.13.b. Accordingly, it is my opinion that would have been obvious in view of Cornwell and a POSITA’s knowledge to have different attribute values for attribute qualifiers.

376. A variety of potential differences in attribute values are illustrated in the following excerpt from Cornwell:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added).

377. For at least these reasons, it is my opinion that Cornwell, in view of the knowledge of a POSITA, renders obvious the “electronic storage device of claim 84, wherein said different attribute qualifiers include different attribute values.”

16. Claim 98 Is Rendered Obvious by Cornwell in View of a POSITA’s Knowledge

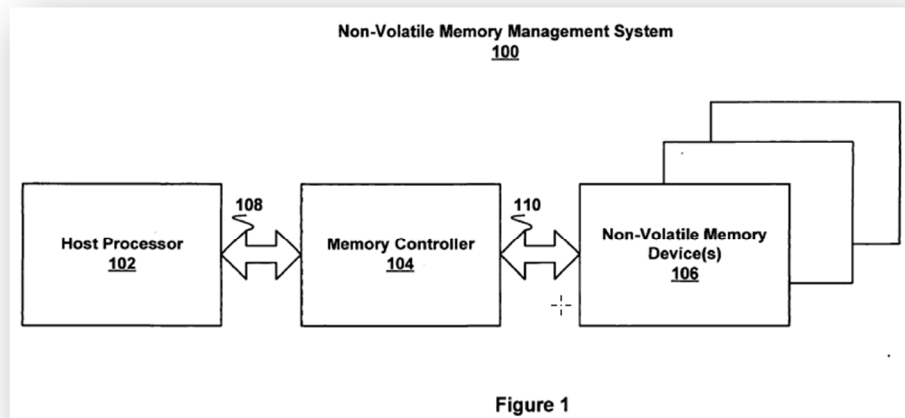
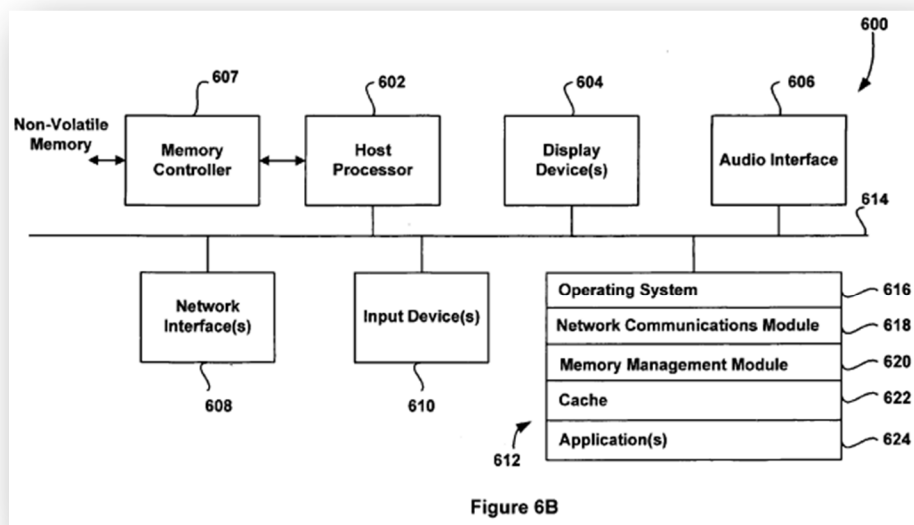
378. As explained below, it is my opinion that Cornwell, in view of a

POSITA's knowledge, renders obvious claim 98 of the '190 patent.

a. [Claim 98 pre]: “An electronic storage device, comprising:”

379. To the extent the preamble is limiting, it is my opinion that Cornwell discloses this claim element. Cornwell teaches a “memory management system” that *can be part of a host system. A host system can be any electronic or computing device that uses non-volatile memory*, including but not limited to: flash drives, portable and desktop computers, clients, servers, consumer electronics, calculators, network appliances, media players/recorders, game consoles, mobile phones, email devices, personal digital assistants (PDAs), embedded devices, televisions, system-on-chip (SoC), set-top boxes, audio recorders, handheld data collection scanners, monitoring devices, etc.” Ex-1005, ¶ [0018] (emphasis added); *see also id.*, ¶¶ [0081]-[0088] (describing “hardware architecture for a host system 600 that includes the memory management system 100” and explaining that the disclosed implementations “can be realized in any device capable of presenting a user interface on a display device”).

380. Figure 1 of Cornwell, shown below, illustrates an exemplary non-volatile memory management system, while Figure 6b, also shown below, illustrates the hardware architecture for a host system that includes the memory management system shown in Figure 1:



381. For at least these reasons, it is my opinion that Cornwell discloses an “electronic storage device.”

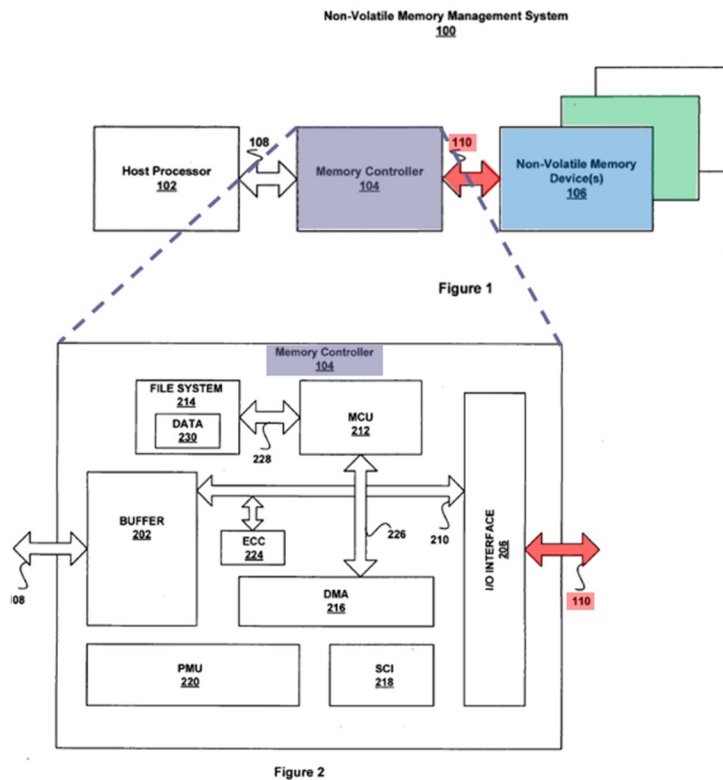
- b. [Claim 98a]: “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile;”

382. It is my opinion that Cornwell discloses this claim limitation. Cornwell states that, “[d]uring operation, *one or more memory devices 106 receive signals from the memory controller 104 over Input/Output (I/O) bus 110*, which enables the memory devices 106 to perform memory access requests (e.g., read or write operations).” Ex-1005, ¶ [0024]. More specifically, “[t]he I/O interface 206 [of memory controller 106] provides connectivity to the memory devices 106 through I/O bus 110, and includes circuitry for enabling read, program, and erase operations to one or more memory devices 106.” *Id.*, ¶ [0031]; *see also id.*, claim 16, (reciting a memory controller comprising “a controller operatively coupled” to “a first interface adapted for coupling to one or more non-volatile memory devices”) (emphasis added).

383. Cornwell also teaches that the memory controller is disposed to address a first memory location and a second memory location. Cornwell explains that “the memory controller 104 recognizes control, *address*, and data signals transmitted . . . by the host processor 102” and “translates the control, *address* and data *signals into memory access requests* on memory devices 106.” *Id.*, ¶ [0021] (emphasis added); *see also id.* (explaining that an IDE/ATA bus may translate “control, *address* and data signals into memory access requests”); *id.*, ¶ [0035] (noting that the memory device has a decoder that “decodes input *addresses*”); *id.*, ¶ [0037] (“The address

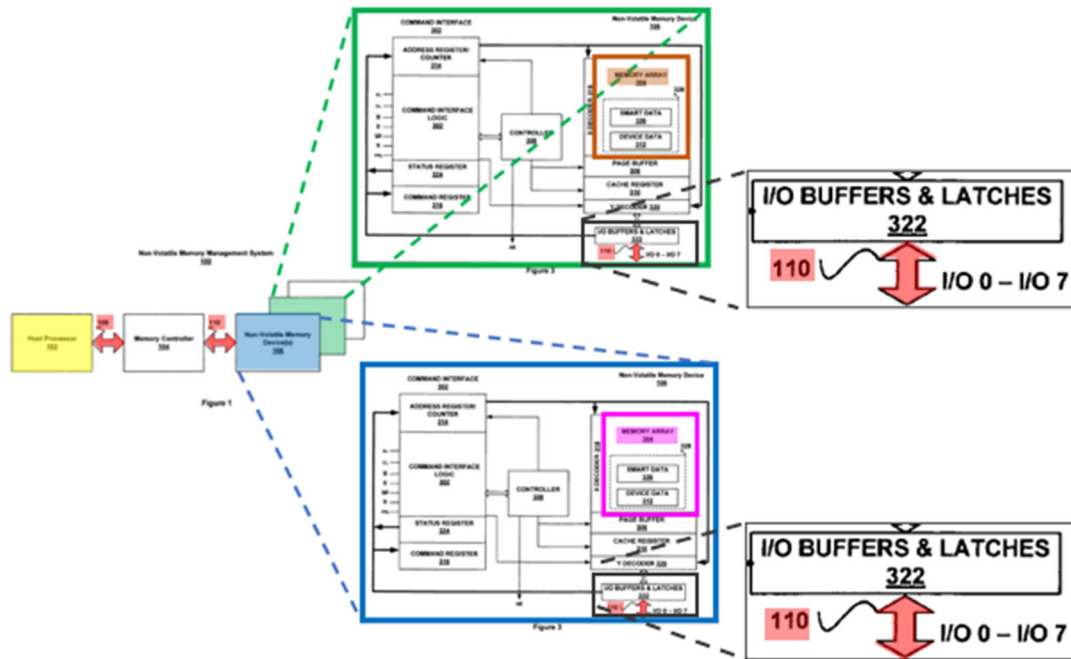
register/counter 314 is configured to receive **addresses** from the memory controller 104 via I/O bus 110.”) (all emphasis added).

384. As illustrated below, annotated Figures 1 and 2 demonstrate that Cornwell’s **memory controller** (“memory controller 104”) is directly coupled to and disposed to address a **first memory location** and a **second memory location** via **bus 110**:



385. Annotated Figures 1 and 3 further confirm that Cornwell’s **memory controller** (“memory controller 104”) is directly coupled to and disposed to address

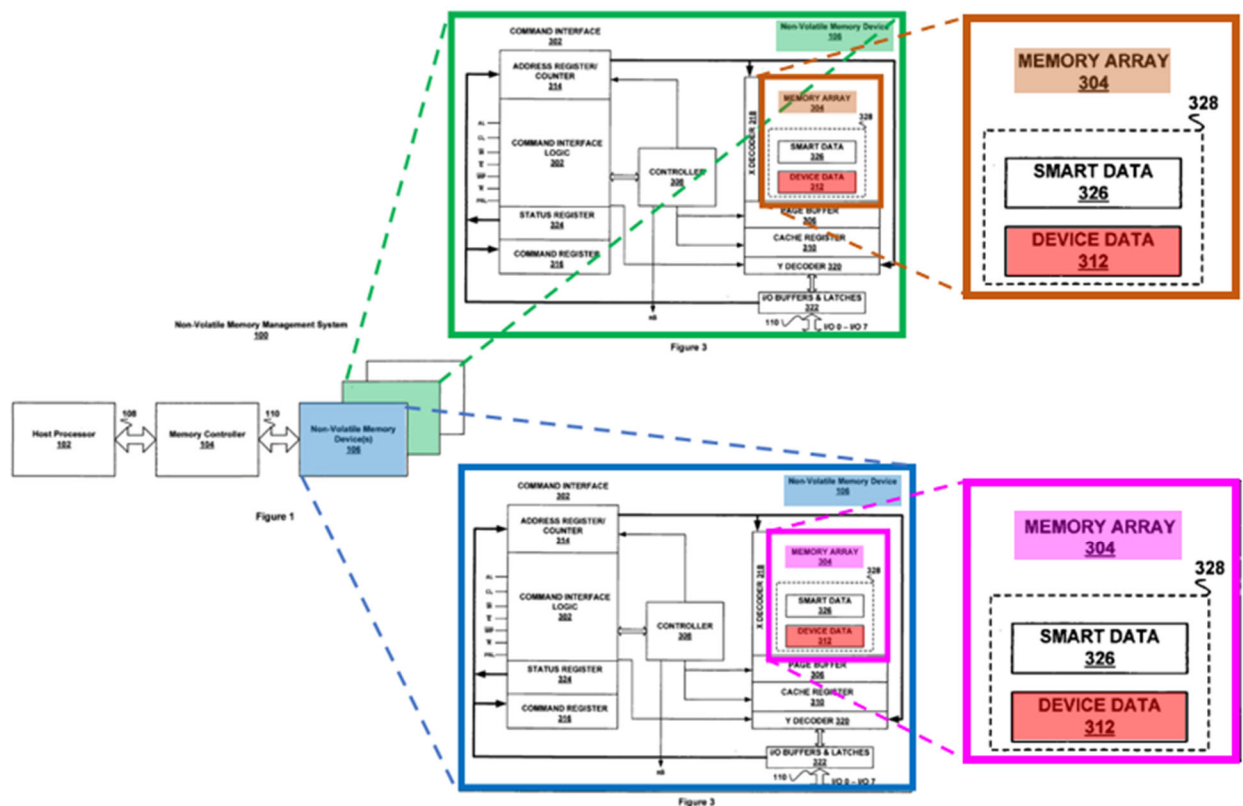
a **first memory location** and a **second memory location** via **bus 110**:



386. Cornwell also discloses that the first and second memory locations are respectively associated with a first device profile and a second device profile. Cornwell discloses that each memory device 106 may include “device data 312 [that] can be used to identify the memory device 106 and its manufacturer.” *Id.*, ¶ [0042]. The device data “can include an electronic signature or serial number that includes a manufacturer code and/or device code,” as well as information such “as device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and

any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.* Accordingly, Cornwell’s collective set of “device data” comprises a “device profile.”³⁰

387. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of device data 312) associated with both the **first memory location** and the **second memory location**:



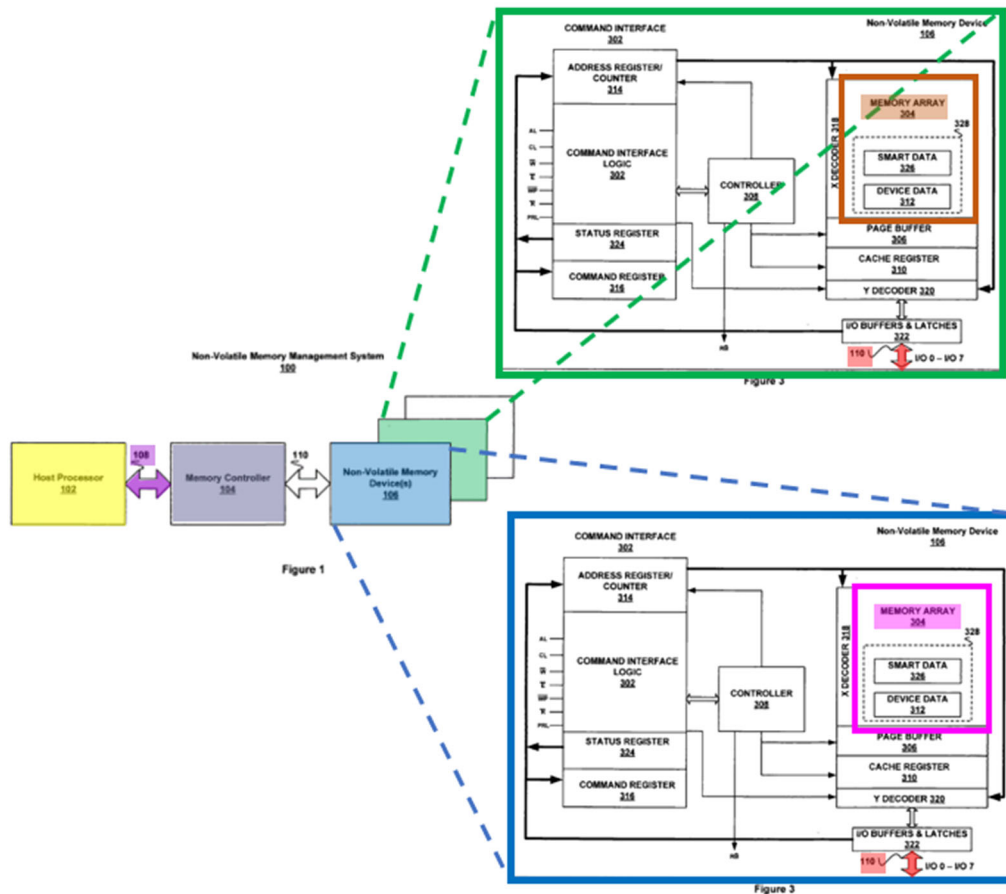
³⁰ See *supra* note 5 (explaining characteristics of device profiles).

388. For at least these reasons, it is my opinion that Cornwell discloses “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile.”

- c. **[Claim 98b]: “wherein the memory controller is directly coupled to a host as the memory controller addresses the first memory location or the second memory location;”**

389. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that “the memory controller 104 recognizes control, address, and data *signals transmitted on bus 108 by the host processor 102*” and “*translates* the control, address and data signals *into memory access requests on memory devices 106*.” Ex-1005, ¶ [0021] (emphasis added); *see also id.*, ¶ [0025] (explaining that data requests “are transmitted to the memory controller 104 on the IDE/ATA *bus 108* by the host processor 102”).

390. Annotated Figures 1 and 3, below, further illustrate that the **host processor (highlighted in yellow)** is directly coupled, **via bus 108**, to the **memory controller 104** as it addresses the **first memory location** or **second memory location**:



391. For at least these reasons, it is my opinion that Cornwell discloses “wherein the memory controller is directly coupled to a host as the memory controller addresses the first memory location or the second memory location.”

- d. [Claim 98c]: “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”

392. It is my opinion that Cornwell renders this claim element obvious in view of a POSITA’s knowledge. As explained above, Cornwell teaches a “device profile” in the form of a collection of “device data 312.” *See supra* § X.A.16.b. The

device profile “can be used to identify the memory device 106 and its manufacturer.” Ex-1005, ¶ [0042]. Information comprising the device profile may include “device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device 106 (collectively, referred to herein as “attributes”).” *Id.*

393. A POSITA would have also understood that the device profiles taught in Cornwell are optimal for a data type subject to the memory transaction. For example, as explained in the preceding paragraph, Cornwell teaches that the device profile identifies whether the memory is NAND or NOR. *Id.* At the time Cornwell was filed, a POSITA would have readily understood that NOR provides random access (i.e., access to a “random data type”) and NAND provides sequential access (i.e., access to a sequential data type”).³¹ Accordingly, it would have been obvious to a POSITA, based on Cornwell alone, that a profile for NOR memory would have been optimal for a random data type subject to a memory transaction, and that a

³¹ See *supra* note 6.

profile for NAND memory would have been optimal for a sequential data type subject to a memory transaction.

394. Similarly, Cornwell teaches “block defining” as another way of ensuring that a device profile is optimal for a data type subject to the memory transactions. *See id.*, ¶¶ [0048]-[0050]. For example, Cornwell teaches that “[m]emory access efficiency can be improved by matching the average size of files to be stored in the flash media *to the block size of the flash media.*” *Id.*, ¶ [0048]; *see also id.*, ¶ [0050] (explaining that “it may be advantageous to *tailor the size of files* such that the *average file size is proportional to the block size.*”) (all emphasis added). As noted above, block size is one of the device profile attributes taught by Cornwell. The district court’s claim construction order states that such block size comparison information is another example of a device profile that is optimal for a data type subject to the memory transaction.³²

³² As noted above, the district court’s claim construction order states that examples of optimal device profiles are “(1) a device profile that includes a block size attribute value that is the largest available from the other device profiles when the data type is sequential,” and “(2) a device profile that includes a block size attribute value that is the smallest available from the other device profiles” when the data type is

395. Moreover, Cornwell teaches that the data type comprises one of a random data type or a sequential data type, explaining that “[i]n some implementations a ***random [data type] read command*** may be issued first, followed by a page [***sequential data type***] read command.” *Id.*, ¶ [0040] (emphasis added). Thus, by matching block size to file size, and teaching that files contain random or sequential data types, Cornwell further teaches first device profile is optimal for a data type subject to the memory transaction.

396. For at least these reasons, it is my opinion that Cornwell in view of a POSITA’s knowledge renders obvious “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- e. **[Claim 98d]: “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

397. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that the memory device “generally includes a command interface 302, a memory array 304 and a controller 308.” Ex-1005, ¶ [0034]. Cornwell further identifies, for example, a variety of commands that are “translated into signals which

random.” *Supra* note 7 (citing Ex-1007 at 44).

can be used by a controller and decoding logic in the memory device 106 to access a memory array.” *Id.*, ¶ [0027]. “For example, when the host processor 102 makes a read request, the “Read Sectors” opcode (20 h) is transmitted to the memory controller 104, together with address and control signals for accessing the sector(s).” *Id.*; *see also id.*, ¶ [0035] (describing a decoder that “decodes signals from the command interface logic 302 for reading or writing data into the memory line”); *id.*, ¶ [0038] (explaining that the controller receives “one or more input addresses and command inputs, which are used by the controller 308 in combination with the control signals from the command interface logic 302 to carry out read and write operations on memory array 304”). Likewise, Cornwell teaches an ““identify” command that returns a chip ID,” which can then be used “to look-up information about the memory devices 106.” *Id.*, ¶ [0077]. As a result, Cornwell discloses identifying command details for causing the memory transaction to be performed, wherein the command details comprise the first memory device.

398. Cornwell also discloses that the interface controller (MCU 212) within the memory controller performs these functions. “The MCU 212 translates IDE/ATA commands into data and control signals required for memory operations. The MCU 212 is coupled via internal bus 228 to the file system 214 which contains MCU firmware for performing various tasks file management tasks. For example,

the MCU firmware can translate signals from the host processor 102 into memory read and write operations.” *Id.*, ¶ [0029].

399. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

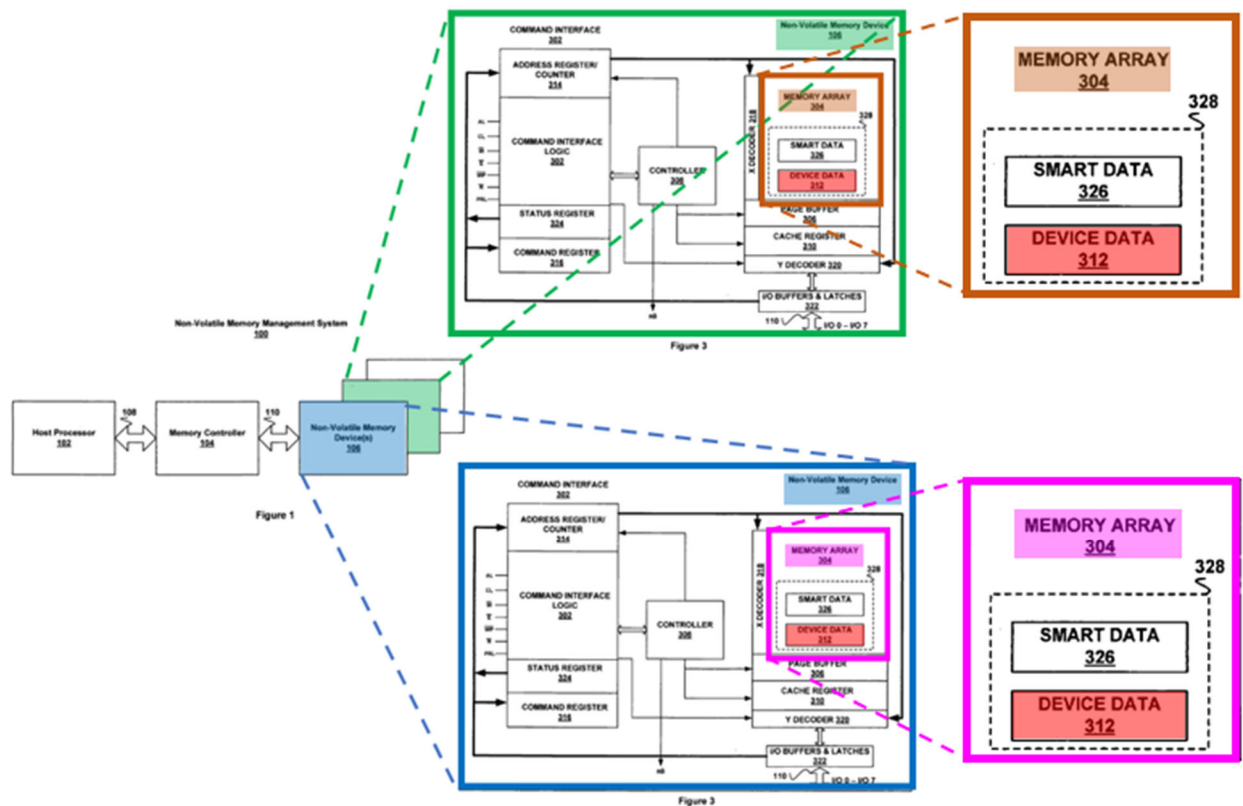
- f. **[Claim 98e]: “the first device profile representing a first set of attributes of the first memory location, and the second device profile representing a second set of attributes of the second memory location;”**

400. It is my opinion that Cornwell discloses this claim element. Cornwell teaches that “the device data 312,” which collectively comprises the device profile, “can include an electronic signature or serial number that includes a manufacturer code and/or device code.” Ex-1005, ¶ [0042]. Cornwell also teaches that the device data 312 “can be used to identify the memory device 106 and its manufacturer.” *Id.* For example, device data 312 can include the following:

manufacturer code and/or device code. Chip data **312** can also include but is not limited to: device type (e.g., NAND, NOR, etc.), device density (e.g., 512 Mb, 1 Gb, 2 Gb, etc.), device operating voltage (e.g., 3.3 volts), page size (1 k, 2K, etc.), spare area size (e.g., 8, 16 bytes, etc.), sequential access time (e.g., 30, 50 nanoseconds, etc.), block size (e.g., 64 k, 128 k, etc.), bus width (e.g., ×8, ×16, etc.), bad block identification, and any other information that is associated with attributes, properties or characteristics of the memory device **106** (collectively, referred to herein as “attributes”).

Id. (highlighting and underlining added). Cornwell states that this information “is associated with *attributes*, properties or characteristics of the memory device 106 (*collectively, referred to herein as “attributes”*).”

401. Annotated Figures 1 and 3, below, illustrate a **device profile** (i.e., the collection of attributes comprising device data 312) associated with both the **first memory location** and the **second memory location**:



402. For at least these reasons, it is my opinion that Cornwell discloses “the first device profile representing a first set of attributes of the first memory location, and the second device profile representing a second set of attributes of the second memory location.”

- g. **[Claim 98f]: “said memory controller obtaining the first set of attributes after identifying the command details;”**

403. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches that attributes comprising device data 312 can be obtained or “transmitted to the memory controller 104 via the I/O bus 110 *in*

response to a read command issued by the memory controller.” Ex-1005, ¶ [0043]. Cornwell’s teaching that the attributes are transmitted “in response to a command” confirms that they are obtained after the command details are identified.

404. For at least these reasons, it is my opinion that Cornwell discloses “said memory controller obtaining the first set of attributes after identifying the command details.”

h. [Claim 98g]: “and wherein said memory controller uses attributes from the first and second device profiles;”

405. It is my opinion that Cornwell discloses this claim element. As explained above, Cornwell teaches that the first and second device profiles (i.e., collectively comprising device data 312) include a variety of attributes. *See supra* § X.A.16.b; Ex-1005, ¶ [0042] (describing attributes). Cornwell also teaches using the attributes from the first and second device profiles, explaining that “device data 312 can be transmitted to the memory controller 104 . . . *in response to a read command* issued by the by the memory controller” and “*used by the memory controller 104 and/or host system* to perform various memory management tasks.” Ex-1005, ¶ [0043] (emphasis added); *see also id.*, ¶ [0042] (explaining that “device data 312 can be used to identify the memory device 106 and its manufacturer”); *id.*, ¶ [0046] (describing a memory management policy that “addresses how read and

write operations should be performed”); *id.*, ¶ [0058] (explaining that the host processor “can request and receive memory device information (e.g., signatures, block size, interleave date, etc.) for one or more memory devices 106 over a standard IDE/ATA bus”).

406. For at least these reasons, it is my opinion that Cornwell discloses “wherein said memory controller uses attributes from the first and second device profiles.”

- i. **[Claim 98h]: “wherein said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

407. It is my opinion that Cornwell discloses this claim element. For example, Cornwell teaches a memory transaction in which “*the controller 308, programs the page* into the memory array 304” Ex-1005, ¶ [0039] (emphasis added). As Cornwell further explains, a page has a specific “page size (1k, 2k, etc.).” *Id.*, ¶ [0042]; *see also, e.g., id.*, ¶ [0048] (“Memory access efficiency can be improved by *matching the average size of files* to be stored in the flash media *to the block size of the flash media.*”); *id.*, ¶ [0050] “[I]t may be advantageous to *tailor the size of files* such that the average file size is *proportional to the block size.*”) (all emphasis added). In this manner, Cornwell teaches selecting a transfer size for the memory transaction.

408. Cornwell further discloses that the page size is a function of the device density of the memory transactions. *Id.*, ¶ [0042]. For example, Cornwell teaches varying “device density (e.g., 512 Mb, 1Gb, 2Gb, etc.).” *Id.*; see also *supra* § X.A.16.b. A POSITA would have understood that page size is the length of a row (number of columns) in the memory. A POSITA would have further understood that the length of a row, again the number of columns in a row, varies with the device density since the device density is the number of rows multiplied by the number of columns in a memory array multiplied by the number of arrays used in the memory chip. For example, a page size of 2k (i.e., there are 2,048 columns in a row), with 8k rows and 1k arrays would result in device density of 16Gb.

409. Cornwell also demonstrates that the transfer size is selected by the controller. As seen in Figure 1 of Cornwell, the host 102 communicates via bus 108 with the memory controller 104 in a format consistent with the IDE/ATA bus standard and its command, including the data size information (e.g., the number of logical sectors to be transferred), while the memory controller 104 communicates with the memory store via the I/O bus 110, using the signals required by the chips used in the memory store and a transfer size, which is selected based on the data size information. As a result, the memory controller selects the transfer size for the memory transaction in response to a request from a host.

410. For instance, Cornwell teaches an example in which a media player sends a request to an operating system on a host. *Id.*, ¶ [0025]. This “request is received by the operating system, which formats the request into IDE/ATA signals, which are transmitted to the memory controller 104 on the IDE/ATA bus 108 by the host processor 102.” *Id.*³³ The fact that host sends the requests to the memory controller—which must then communicate with the memory store—ensures that the memory controller selects the transfer size for the memory transactions, where the transfer size is a function of the data size of the memory transaction and the first set of attributes to retrieve the requested media file.

411. For at least these reasons, it is my opinion that Cornwell discloses “wherein said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

B. Each of the Challenged Claims Is Rendered Obvious by Ware in View of a POSITA’s Knowledge

412. It is my opinion that claims 1, 5, 6, 29, 36, 37, 45, 51, 52, 59, 71, 76,

³³ As Cornwell also explains, “ATA-6 is the latest version of the IDE/ATA standard, which was approved by the American National Standards Institute (ANSI) in 2001 under document NCITS 347-2001.” *Id.*, ¶ [0026].

80, 84, 85, and 98 of the '190 patent are rendered obvious by Ware in view of a POSITA's knowledge.

1. Claim 1 Is Rendered Obvious by Ware in View of a POSITA's Knowledge

413. It is my opinion that claim 1 of the '190 patent is rendered obvious by Ware in view of a POSITA's knowledge.

a. [Claim 1 Pre]: "A method of performing transactions on a set of memory locations that includes a first memory location and a second memory location, the method comprising:"

414. To the extent the preamble is limiting, it is my opinion that Ware discloses a method of performing transactions on a set of memory locations that includes a first memory location and a second memory location. For example, Ware discloses "memory systems and methods" utilizing "a memory controller for a non-homogeneous memory system." Ex-1006 at 1:6-8; *see also id.*, claims 23-31 (reciting a "method of managing memory in a non-homogenous memory system"). A method of performing read and write transactions on set of memory locations is also illustrated in Figure 3, reproduced below, which provides a flow diagram of "a non-homogeneous memory management process." *Id.* at 11:62-63 (teaching a method for performing read and write transactions on a set of memory locations that includes different types of memory).

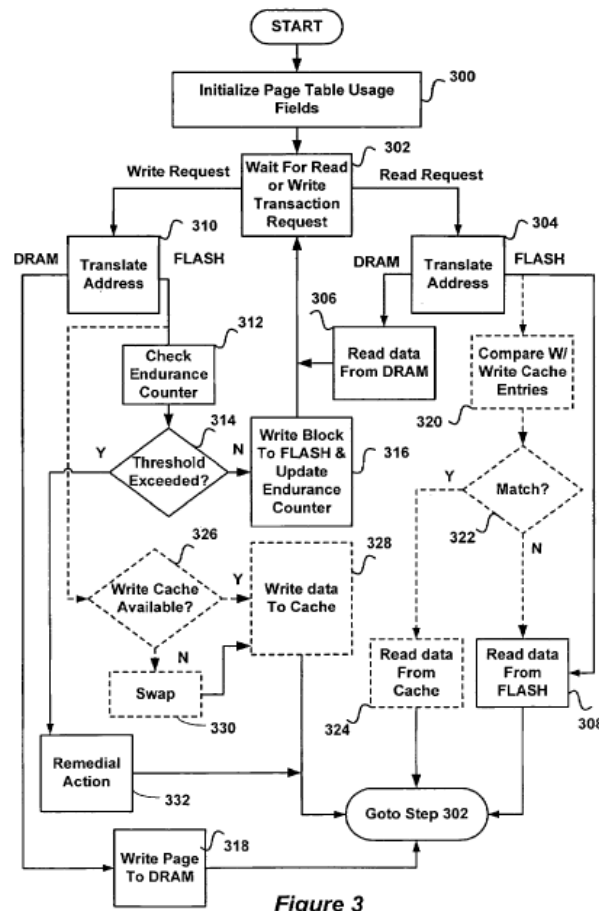
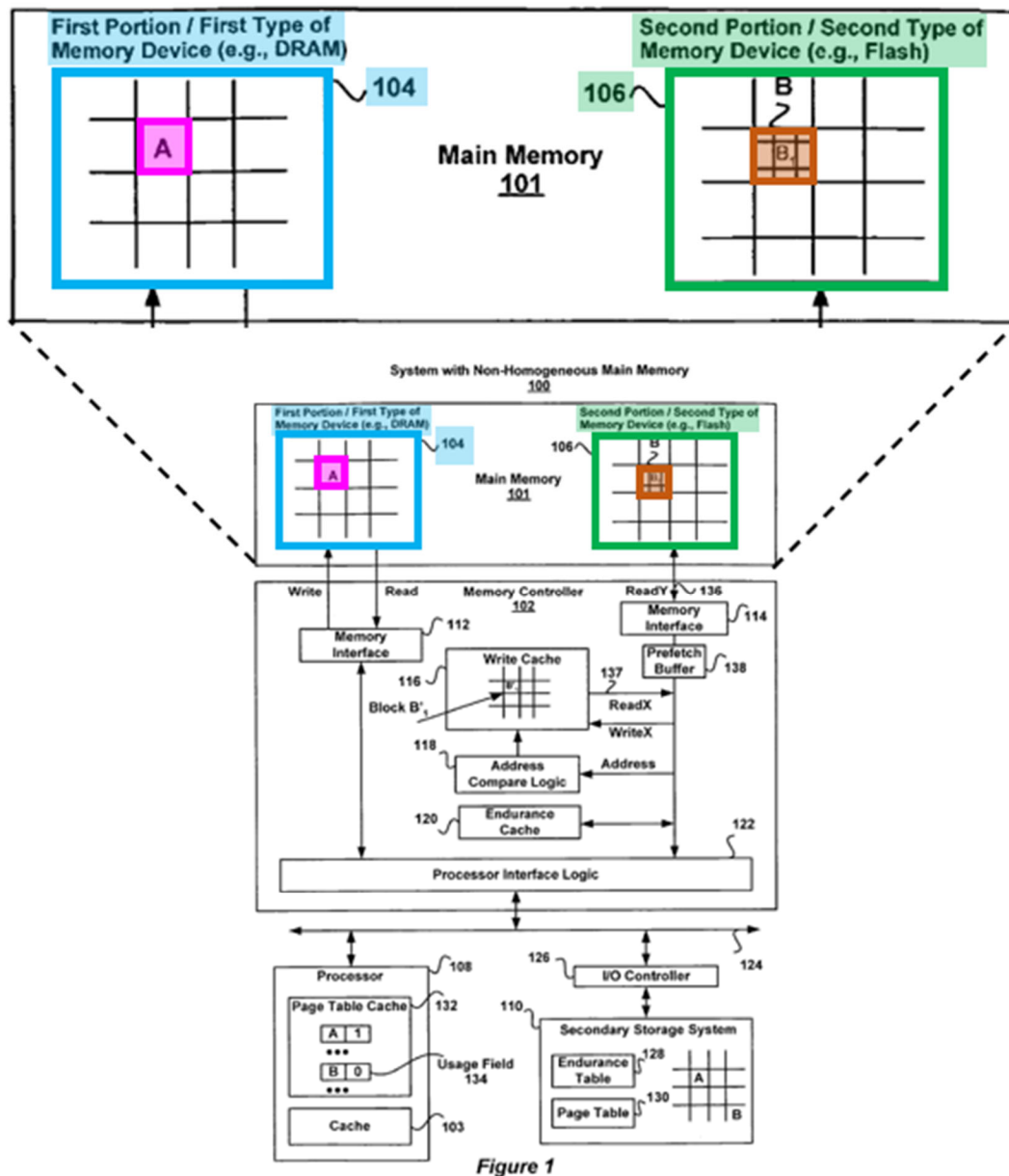


Figure 3

415. Ware also discloses performing memory transactions on a set of memory locations that includes a first memory location and a second memory location. For example, Figure 1, reproduced in annotated form below, discloses a “main memory 101 [that] includes *a first portion 104* implemented with one or more memory devices of a first type (*e.g., DRAM*), and a *second portion 106* implemented with one or more memory devices of a second type (*e.g., NOR Flash, often simply called Flash memory*).” *Id.* at 3:56-61 (emphasis added). Ware further teaches that “a usage field value of ‘A’ represents *a page mapped to the first portion 104 of*

main memory 101, while a value of ‘B’ represents *a page mapped to the second portion 106* of main memory 101.” *Id.* at 5:57-61 (emphasis added).

416. Annotated Figure 1, shown below, illustrates the **first memory location** (field value A in a **first memory portion 104**) and a **second memory location** (field value B in a **second memory portion 106**):



417. For at least these reasons, it is my opinion that Ware discloses “[a] method of performing transactions on a set of memory locations that includes a first memory location and a second memory location.”

- b. [Claim 1a]: “performing a memory transaction comprising addressing a first memory location in a

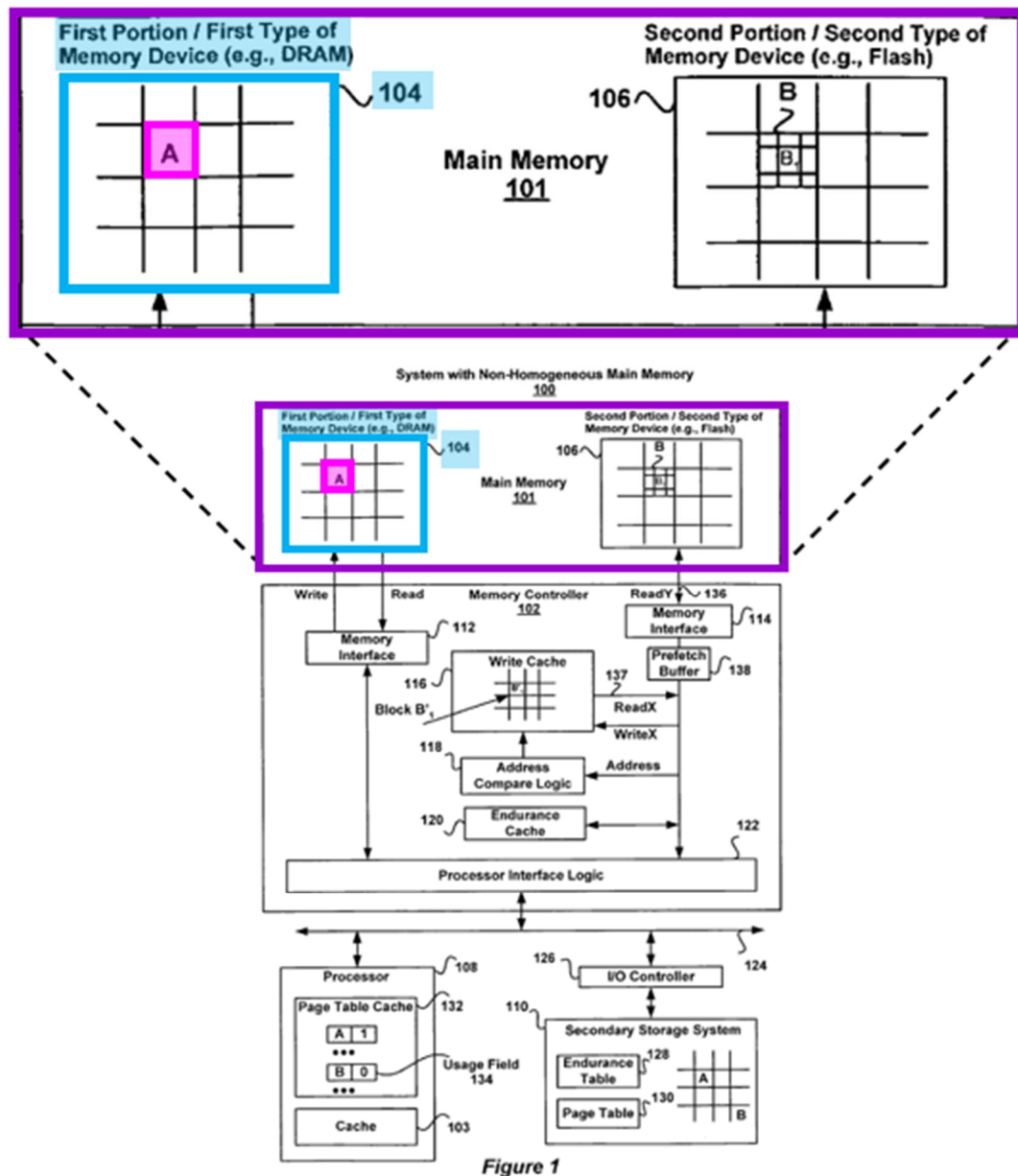
first memory device in a memory store,”

418. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a memory store comprising a first memory location (first memory portion 104) and a second memory location (second memory portion 106). Ex-1006 at 3:56-61; *supra* § X.B.1.a. The first memory portion may be “implemented using one or more DRAM memory devices,” while the second memory portion may be “implemented using one or more Flash memory devices (e.g., NOR Flash memory devices).” Ex-1006 at 4:38-42; *see also id.* at 7:22-25 (noting that a memory controller includes interfaces for coupling memory “in two or more distinct portions (104, 106)”; *supra* § X.B.1.a.

419. Ware also teaches performing a memory transaction comprising addressing a first memory location in the first memory device. For example, Ware discloses that “[d]ata can be distributed among the first and second portions 104, 106 of main memory 101.” Ex-1006 at 3:64-67. Ware also teaches that “a usage field value of ‘A’ represents a page mapped to the first portion 104 of main memory”—i.e., the first memory location. *Id.* at 5:57-61. A POSITA would readily understand that performing read and write transactions to the first memory location in the first memory device, as described by Ware, would require addressing that location. *See id.* at 7:22-25 (explaining that the memory device interfaces “include

signal conditioning circuitry and other devices for transmitting addresses, data, and control signals to and from the memory devices in main memory portions 104 and 106”); *id.* at 7:46-49 (noting that the processor interface logic may include circuitry “for transmitting data, address, and control signals”).

420. Annotated Figure 1, below, illustrates the **memory store**, which comprises a **first memory location** (field value A in a **first memory portion 104**), that is addressed in a read or write memory transaction:



421. For at least these reasons, it is my opinion that Ware discloses “performing a memory transaction comprising addressing a first memory location in a first memory device in a memory store.”

- c. [Claim 1b]: “said first memory location and a second memory location respectively associated with a first

device profile and a second device profile,”

422. It is my opinion that Ware discloses this claim element. For example, Ware teaches “one or more first memory devices of a first memory type having a *first set of attributes*, and a second memory interface adapted to be coupled to one or more second memory devices of a second memory type having a *second set of attributes*.” Ex-1006, Abstract (emphasis added); claim 15 (reciting “one or more first memory devices . . . having a first set of attributes” and “one or more second memory devices . . . having a second set of attributes”). More specifically, Ware discloses a “usage field 208” that “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212*.” *Id.* at 11:1-4 (emphasis added). In one embodiment of Ware, such as Figure 2, the usage model is a one-bit flag that indicates whether the requested page is expected to be read-only or read-write. However, Ware discloses other embodiments: “In the above example, the usage field 208 is a one-bit flag that indicates whether the requested page is expected to be read-only or whether it is to be read-Write. It should be apparent, however, that the usage field 208 can represent other page usage models, and include more or fewer bits.” *Id.* at 11:19-23. In this manner, Ware teaches a first device profile, or set of attributes, associated with a first memory location (the

first memory portion comprising DRAM) and a second device profile, or set of attributes, associated with a second memory location (the second memory portion comprising Flash).

423. Annotated Figure 2, shown below, illustrates the **device profile** (i.e., the usage field used to store a first and second sets of attributes), which is associated with both the **first memory location** (in the first memory device) and the **second memory location** (in the second memory device):

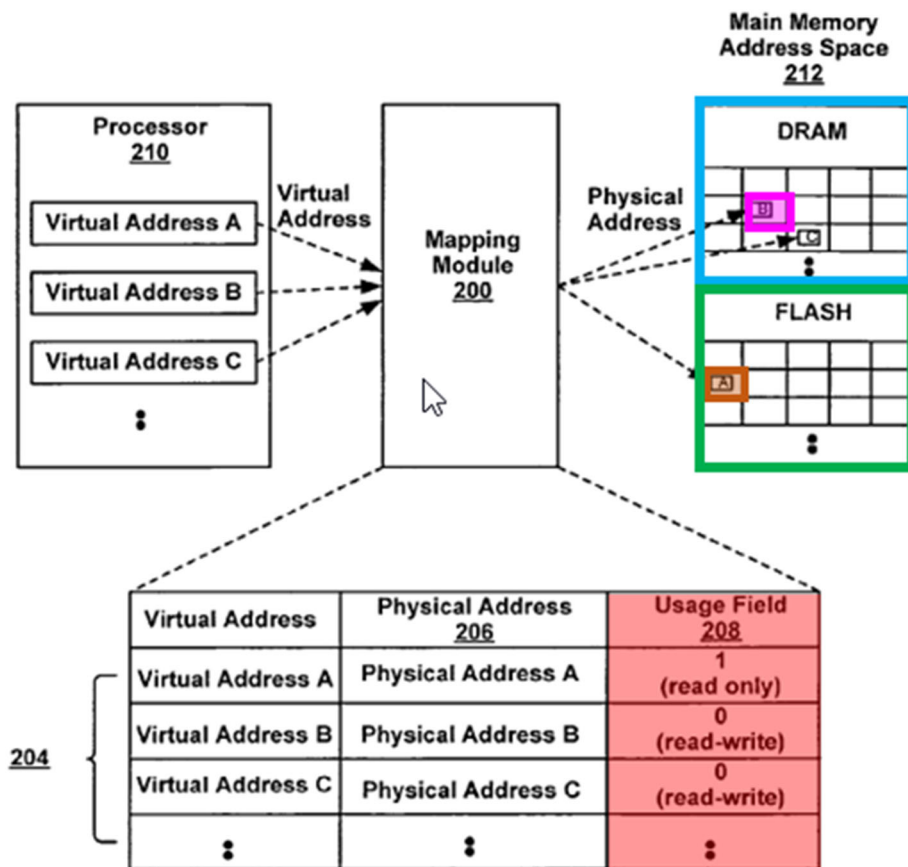


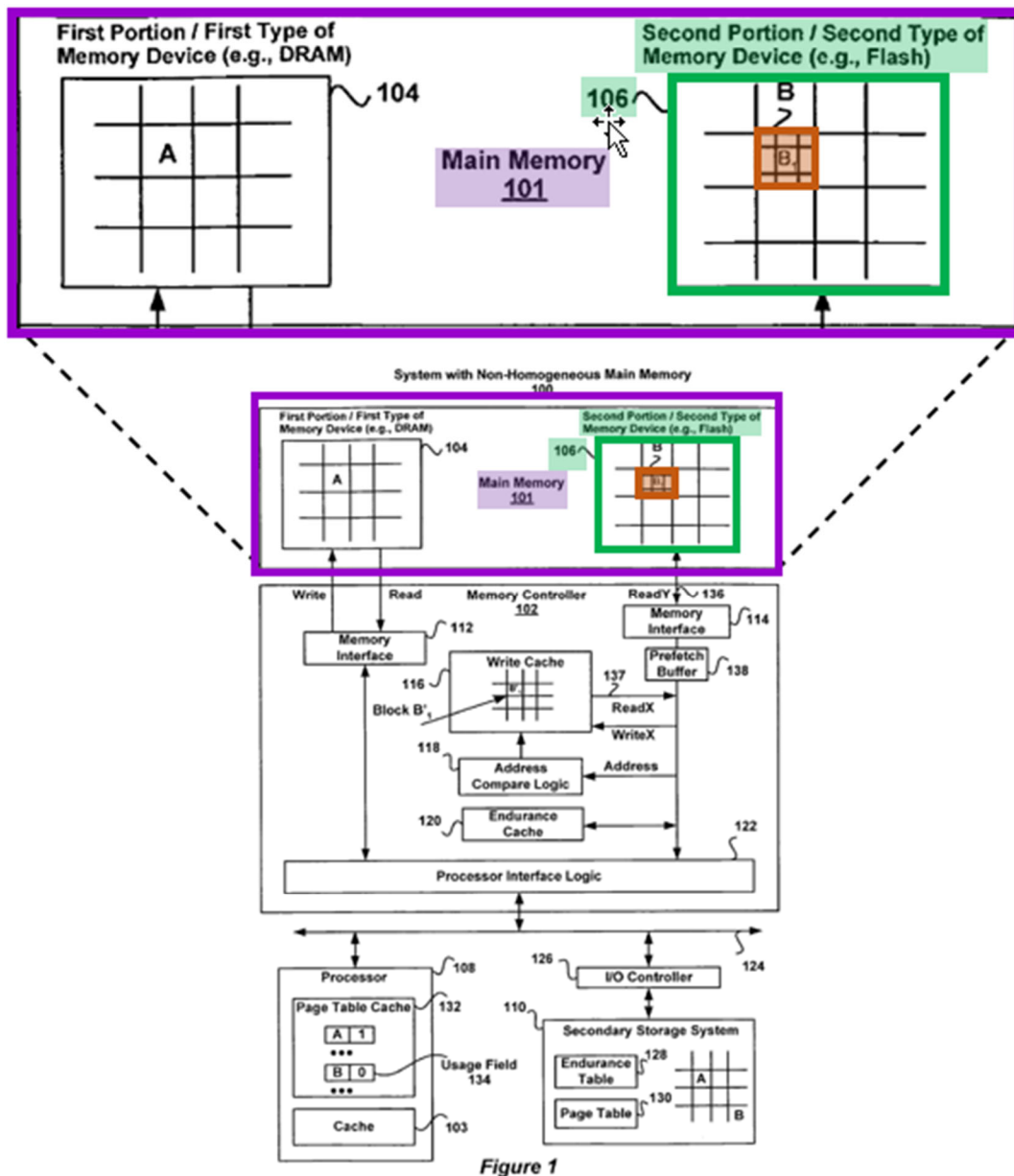
Figure 2

424. For at least these reasons, Ware discloses that “said first memory location and a second memory location respectively associated with a first device profile and a second device profile.”

- d. [Claim 1c]: “wherein said second memory location is in a second memory device in the memory store,”

425. It is my opinion that Ware discloses this claim element. As explained

above, the second memory location (i.e., usage field “B”) is in the second memory device (second portion 106) in the memory store. *See supra* § X.B.1.c. This is further illustrated in annotated Figure 1, below, which shows that the **second memory location** (usage field “B”) is in the **second memory device** (second portion 106) in the **memory store** (main memory 100):



426. For at least these reasons, it is my opinion that Ware discloses that “said second memory location is in a second memory device in the memory store.”

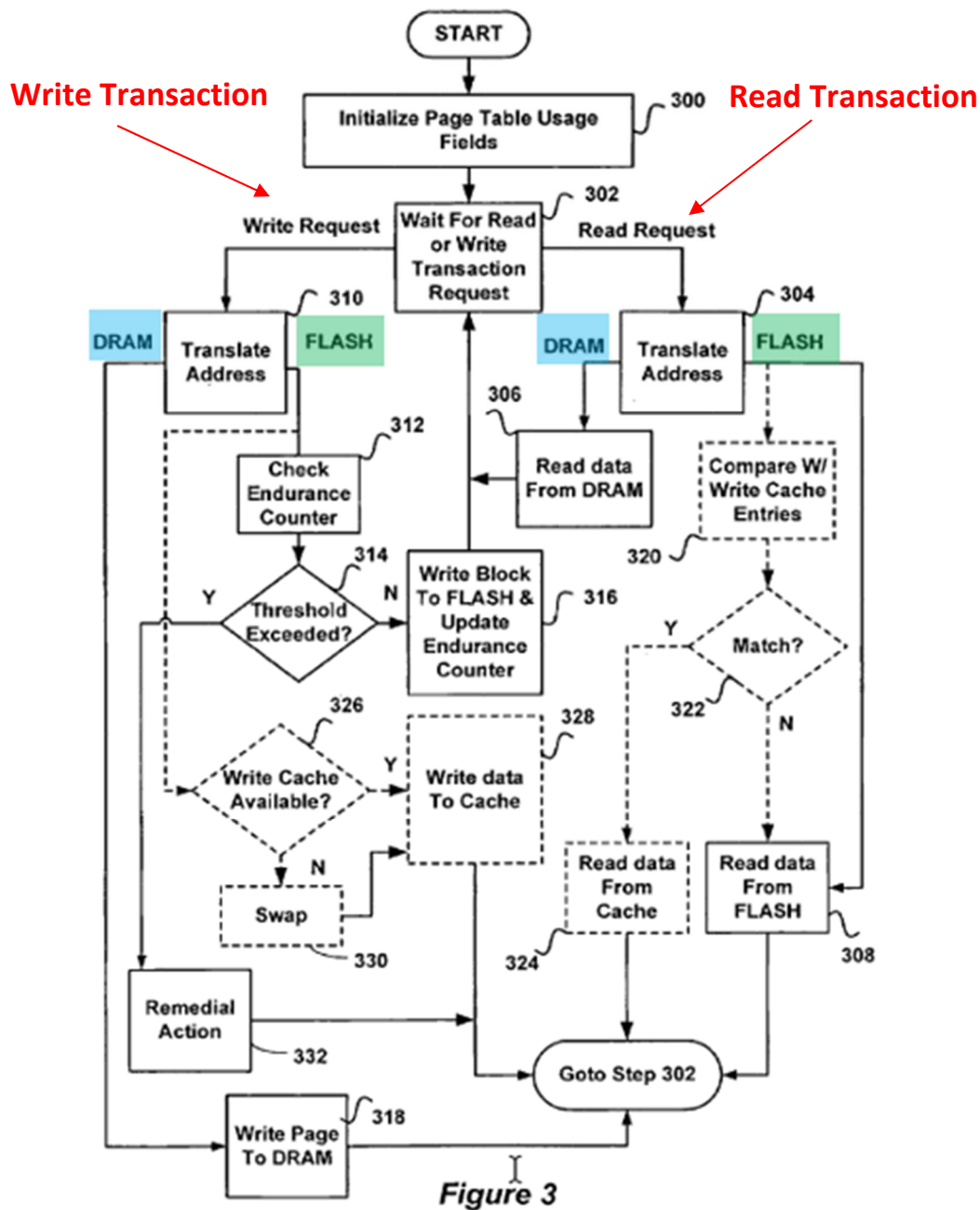
- e. [Claim 1d]: “wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location,”

427. It is my opinion that Ware discloses this limitation. For example, Ware teaches that “[a]fter the initialization phase is complete, the memory controller waits 302 for a ***read or write transaction request***. If a ***read request*** is received, the virtual address is translated into a physical address at step 304. . . . If the usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be ***read from DRAM***, then the data (which may contain program code) ***is read at step 306 from DRAM*** and returned to the requestor. The process then returns to step 302 to wait for ***another read or write transaction request***.” Ex-1006 at 12:12-24 (emphasis added).

428. Likewise, Ware teaches that “[i]f the transaction request is a ***write request***, the virtual address is translated into a physical address at step 310, which once again includes reading the usage field for the page containing the specified address. If the usage field indicates that the ***write data should [be] written to DRAM*** (i.e., to a first portion of main memory), then ***the write data is written to DRAM*** at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45 (emphasis added); *see also id.* at 2:30-36 (“[V]irtual memory pages that are read-write, especially pages that may be written a large number of times, are mapped to a first portion of main memory implemented using one or more memory devices of a first type, while virtual memory pages that are

read-only are mapped to a second portion of main memory, implemented using one or memory devices of a second type.”). Ware similarly teaches reading and writing to the second location in flash memory. *Id.* at 7:55-67; 8:24-28; 9:48-57.

429. This is further illustrated in annotated Figure 3, below, which shows that a memory read transaction (read request) and memory write transaction (write request) are performed on either a **first memory location** (in DRAM) or a **second memory location** (in Flash memory):



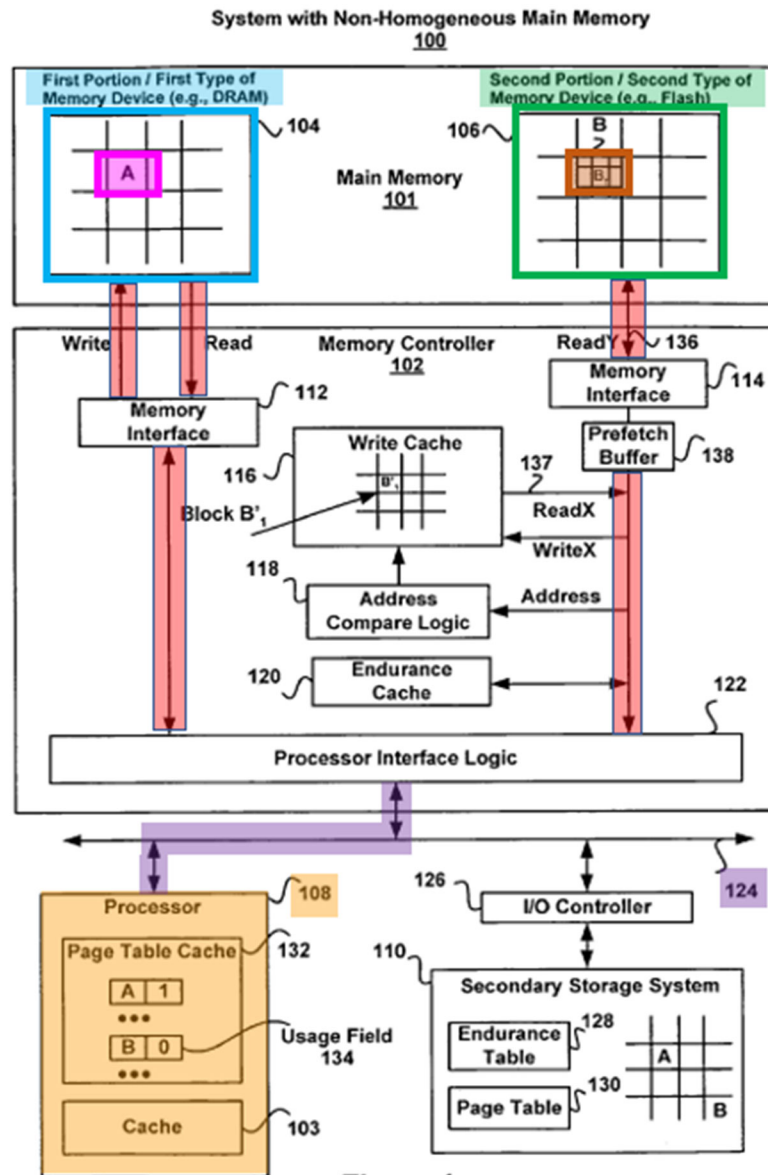
430. For at least these reasons, it is my opinion that Ware teaches “wherein a memory read transaction or a memory write transaction is performed on at least one of the first memory location or second memory location.”

- f. **[Claim 1e]: “wherein data is transferred to the first memory location or second memory location from a host for a memory write transaction, wherein data is transferred from the first memory location or second memory location to the host for a memory read transaction;”**

431. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches that data is read from a first or second memory location in a read transaction and is written to a first or second memory location in a write transaction. *Supra* § X.B.1.e. Ware also teaches that the read and write transactions occur between a host and the first or second memory locations. For example, Ware teaches a host computer system comprising a processor 108, which “is coupled to the memory controller 102 via bus 124 and interface logic 122.” Ex-1006 at 3:39-44, 7:9-10. The “programs and data used by [host] processor 108” are stored in the main memory, which is comprised of the first memory location (DRAM) and the second memory location (Flash). *Id.* at 3:61-63. As a result, Ware teaches that data is transferred to the first or second memory locations for a write transaction, and is transferred to the host from the first or second memory locations for a read transaction.

432. Annotated Figure 1, shown below, illustrates data being transferred by the **host processor** (processor 108) over **bus 124** and **communication paths** to and from the **first memory location** (usage field “A” in the first memory device) and

second memory location (usage field “B” in the second memory device):



433. For at least these reasons, it is my opinion that Ware teaches “wherein data is transferred to the first memory location or second memory location from a host for a memory write transaction, wherein data is transferred from the first

memory location or second memory location to the host for a memory read transaction.”

- g. **[Claim 1f]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”**

434. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a first memory location comprising DRAM having a device profile. A POSITA would have understood that a random read command results in accessing a byte of data at some random location in the memory because that is an essential operation of any random read command. Thus, a POSITA would understand that the device profile for DRAM would be optimal for a random data type. Indeed, Ware teaches that “[m]ain memory is typically implemented with Dynamic Random [data type] Access Memory (DRAM),” which “has a near-*optimal combination of operational attributes*”—comprising a device profile—“for implementing main memory.” Ex-1006, 1:65-2:6; *see also id.* at 11:1-4 (disclosing a “usage field 208” or profile that “stores data indicative of the usage model of the page, which can be used by processor 210 to *exploit the unique attributes of a physical memory device*,” such as DRAM’s suitability for random data) (all

emphasis added).³⁴

435. For at least these reasons, it is my opinion that Ware discloses “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- h. [Claim 1g]: “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

436. It is my opinion that Ware discloses this claim element. Ware teaches that, “[a]fter the initialization phase is complete, the memory controller waits [at] 302 for a read or write transaction request.” Ex-1006 at 12:12-14. A POSITA would readily understand that performing the read or write transaction request requires identifying command details for causing the memory transaction to be performed.

437. For example, Ware teaches that “[i]f a read request is received, the

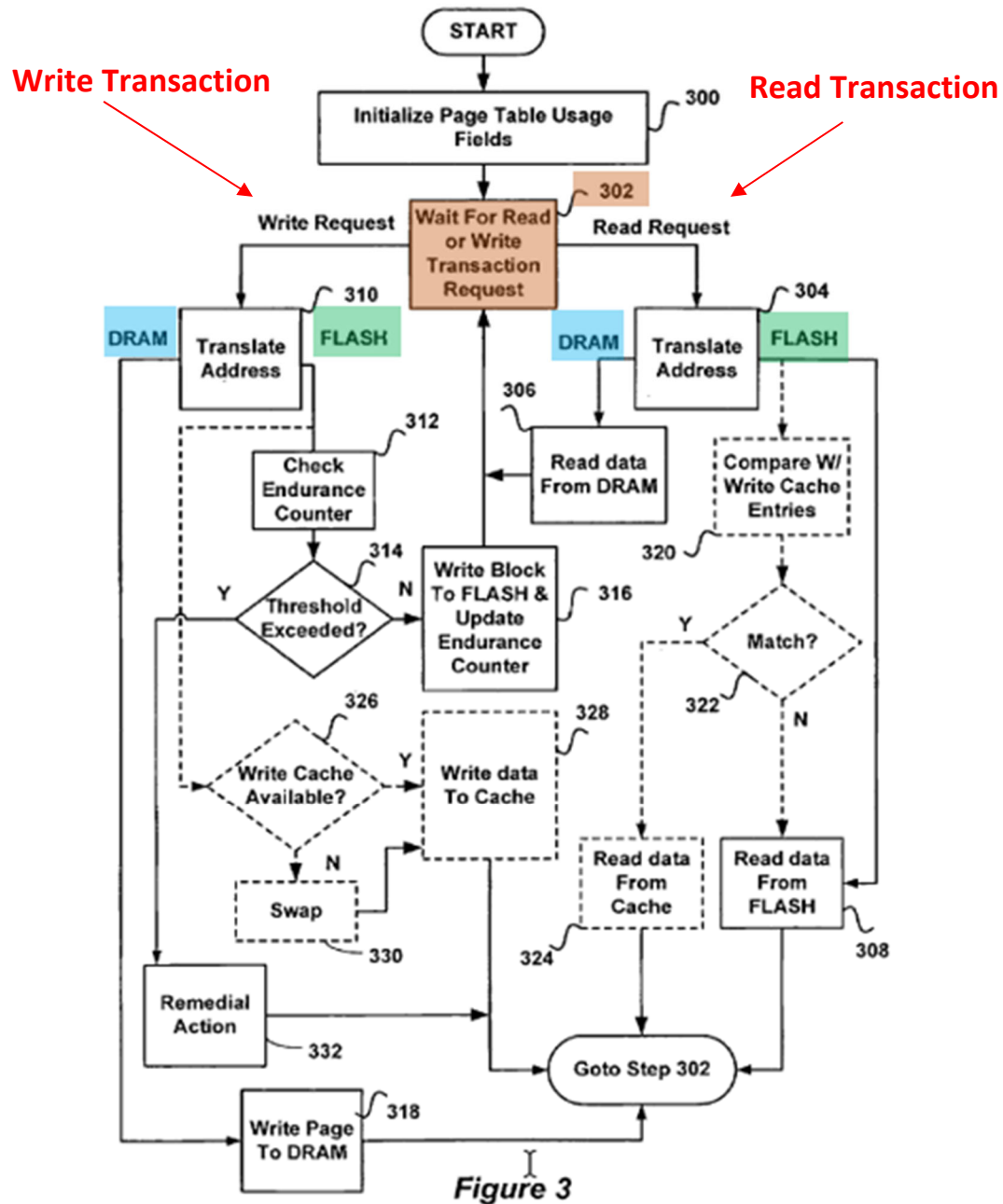
³⁴ Similarly, a POSITA would have understood that a second device profile for flash memory, particularly NAND flash, utilizes page read commands that result in sequentially accessing bytes of data located in a row location (page) of the memory. *See supra* note 6. As a result, flash memory would have a device that is optimal for a sequential data type.

virtual address is translated into a physical address at step 304.” *Id.* at 12:14-15. Moreover, “[o]ne aspect of step 304 is reading the usage field for the page. If the usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be read from DRAM, then the data (which may contain program code) is read at step 306 from DRAM and returned to the requestor.” *Id.* at 12:17-22. Likewise, “[i]f the transaction request is a write request, the virtual address is translated into a physical address at step 310, which once again includes reading the usage field for the page containing the specified address. If the usage field indicates that the write data should be written to DRAM (i.e., to a first portion of main memory), then the write data is written to DRAM at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45.

438. A POSITA would understand that command details would be required for executing these steps of read or write request. As such, Ware requires identifying command details for identifying a first memory location and causing a memory transaction to be performed on the first memory location.

439. Annotated Figure 3, below, further illustrates determining a read or write commence at step 302, which requires **identifying command details** for causing the memory transaction to be performed for the **first memory device**

(DRAM):



440. For at least these reasons, it is my opinion that Ware teaches “said performing the memory transaction further comprising identifying command details

for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- i. **[Claim 1h]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

441. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses that both the first memory location (first memory portion 104) and second memory location (second memory portion 106) have a set of attributes comprising a device profile. *See supra* § X.B.1.c. Ware further teaches that “[t]he memory devices in portion 104 have one or more ***attributes that differ*** from the attributes of the memory devices in portion 106.” Ex-1006 at 4:4-9 (emphasis added); *see also id.* at Abstract (“The first and second sets of attributes have ***at least one differing attribute***”); *id.* at 2:21-25 (“The memory devices . . . include two or more memory device types (e.g., DRAM, Flash) ***with different attributes.***”); *id.* at 4:19-29 (explaining that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “***differs from***” DRAM in which the read and write access times are less than 100 ns); *id.*, claim 15 (reciting a first memory type and second memory type having attributes, “wherein the first and second sets of attributes have ***at least one differing attribute***”) (all

emphasis added). The differences in these attributes demonstrate that a difference exists in the device profiles for the first and second memory locations.

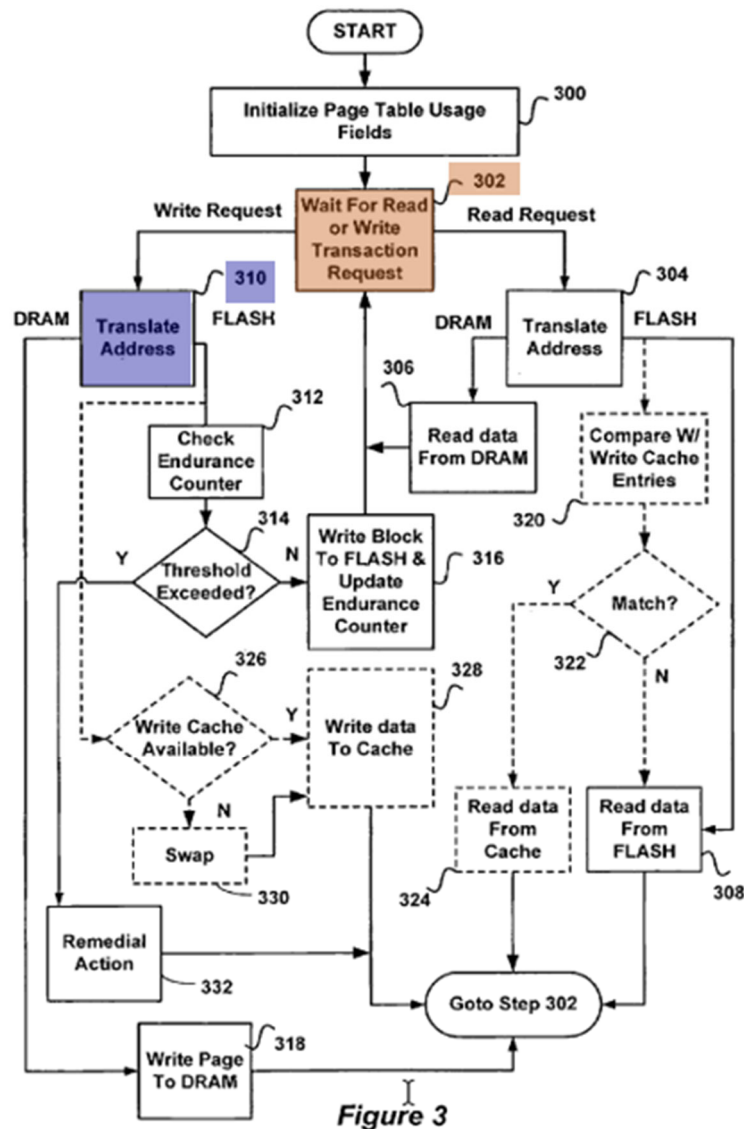
442. For at least these reasons, it is my opinion that Ware discloses “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

j. [Claim 1i]: “after identifying the command details, obtaining the first set of attributes;”

443. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches identifying command details for causing a memory transaction to be performed. *Supra* § X.B.1.h; *see also supra* § X.B.1.i. Ware further teaches that the first set of attributes are obtained after the command details are identified. For example, in relation to Figure 3, reproduced below, Ware teaches that once it has been determined that a transaction request is a write request at step 302—i.e., once the command details for a write transaction have been identified—***then*** a virtual address is “translated into a physical address at step 310, which once again ***includes reading the usage field*** for the page containing the specified address.” Ex-1006 at 12:38-41; *see also id.* at 12:41-45 (explaining that the usage field then indicates whether the write data should be written to DRAM). The usage field

“stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212.*” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches that, after identifying the command details for a read or write (as shown in step 302 of annotated Figure 3 below), a first set of attributes in the usage field, which describe the corresponding usage model for the first memory location in memory device 104, are obtained (as shown in step 310 of annotated Figure 3 below).

444. Annotated Figure 3, below, illustrates that—after the **command details are identified** in step 302—the address is translated in step 310, which requires **obtaining the set of attributes**:



445. For at least these reasons, it is my opinion that Ware discloses “after identifying the command details, obtaining the first set of attributes.”

- k. **[Claim 1j]: “wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory location;”**

446. Ware renders this claim element obvious in view of the knowledge of

a POSITA. Ware discloses that “[t]he memory devices in portion 104 have one or more attributes that differ from the attributes of the memory devices in portion 106. Examples of such attributes are: ***the ratio of write access time to read access time, volatility, and endurance.***” Ex-1006 at 4:6-11 (emphasis added). Ware further teaches that “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1.” *Id.* at 4:19-24 (emphasis added). This attribute differs from DRAM, in which the ratio of write access time to read access time “is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:29-31. These ratios are qualifiers for the attribute of the ratio of write access time to read access time.

447. A POSITA would have further understood that the attribute qualifiers for the attribute “volatility” would be yes/no (1/0 indicating the memory is either volatile or non-volatile) while the attribute qualifier for endurance would be a number indicating the number of program/erase cycles the memory could endure before the memory becomes unreliable.

448. Accordingly, it is my opinion “attribute qualifiers”—such as values representing the ratio of write time to speed time, volatility, and endurance—would be well known and readily supplied based on the common sense of knowledge of a

POSITA. Indeed, a POSITA would know that memory attributes, including the attributes described in Ware, comprise a “qualifier” to describe the attribute, such as the value of the attribute or the type of the attribute.

449. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious “wherein each attribute in the first set of attributes and second set of attributes is associated with a respective attribute qualifier that qualifies a respective memory location.”

- I. **[Claim 1k]: “wherein the memory store is directly coupled to at least one memory bus and wherein the at least one memory bus is directly coupled to a controller, and wherein the controller performs the memory read transaction and memory write transaction on the memory store;”**

450. It is my opinion that Ware discloses this claim element. For example, Ware teaches that “[t]he memory controller 102 includes memory interfaces 112, 114 for *coupling the memory controller 102 to the memory devices in two or more distinct portions (104, 106) of main memory 101*, and processor interface logic 122.” Ex-1006 at 7:21-25; *see also id.*, Abstract (“A memory controller includes a first memory interface adapted to be coupled to one or more first memory devices . . . and a second memory interface adapted to be coupled to one or more second memory devices.”); *id.*, claim 1 (same). Ware also teaches that “[t]he memory interfaces 112 and 114 may include signal conditioning circuitry and other devices for transmitting

addresses, data, and control signals to and from the memory devices in main memory portions 104 and 106.” *Id.* at 7:25-29 (emphasis added).

451. Annotated Figure 1, shown below, illustrates that the **memory store** (main memory 101) is directly coupled to the **memory controller** (memory controller 102) via **communication paths**, which a POSITA would understand as encompassing a memory bus:

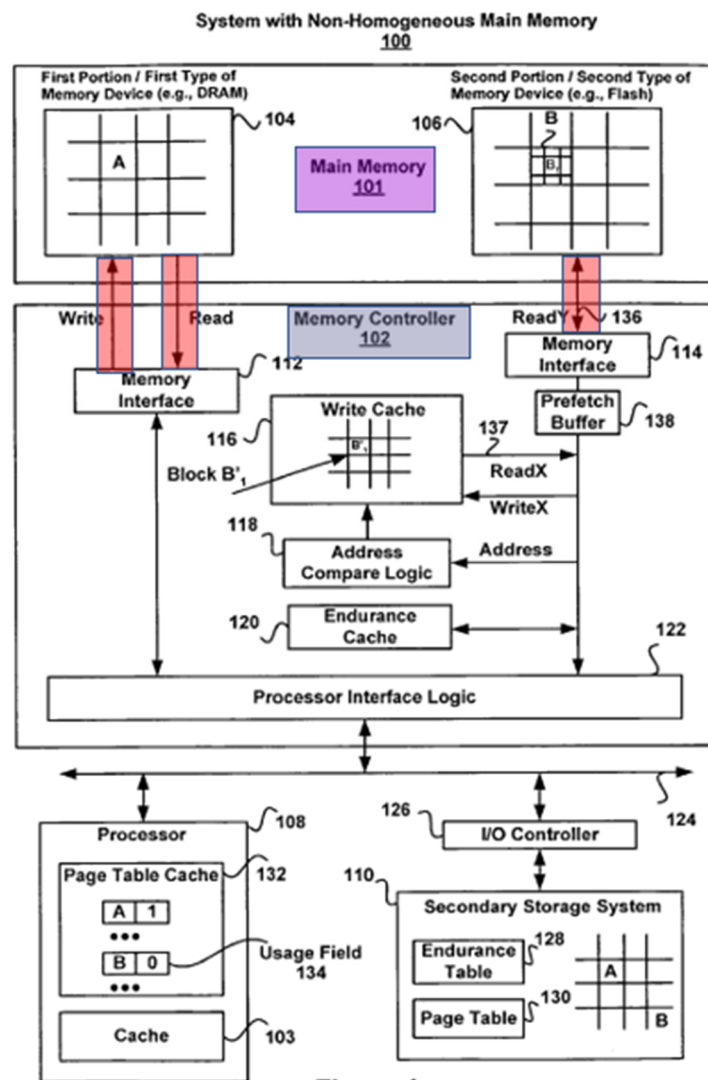


Figure 1

452. Ware also teaches that the controller performs read and write transactions on the memory store by “transmitting addresses, data, and control signals to and from the memory devices in main memory portions 104 and 106.” *Id.* at 7:21-29; *see also id.*, Abstract (“The controller also includes interface logic configured to direct memory transactions.”); *id.* at 12:12-64 (explaining the process whereby the memory controller performs a read or write transaction); *id.*, claim 1 (reciting “a controller comprising . . . interface logic . . . configured to direct memory transactions”).

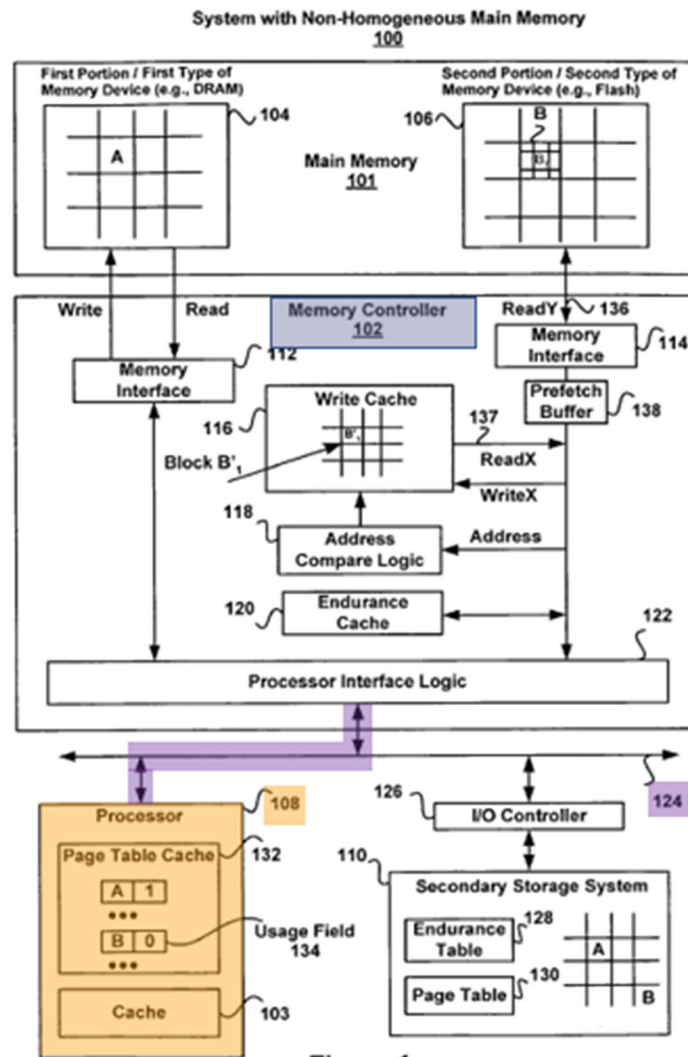
453. For at least these reasons, it is my opinion that Ware discloses “wherein the memory store is directly coupled to at least one memory bus and wherein the at least one memory bus is directly coupled to a controller, and wherein the controller performs the memory read transaction and memory write transaction on the memory store.”

- m. **[Claim 1]: “wherein the host is directly coupled to the controller by a communication path as the controller performs the memory read transaction or memory write transaction;”**

454. It is my opinion that Ware discloses this claim element. Ware teaches that a host “*processor 108 is coupled to the memory controller 102 via bus 124* and processor interface logic 122.” Ex-1006 at 7:9-10. Ware further discloses that the host processor is coupled to the controller as it performs memory transactions,

explaining, for example, that the “main memory 101 stores programs and data used by processor 108 during execution of those programs.” *Id.* at 3:61-63; *see also id.* at 7:46-49 (teaching that the controller comprises “processor interface logic 122 [that] may include signal conditioning circuitry and other devices for transmitting data, addresses and control signals to and from the bus 124”); *supra* § X.B.1.1.

455. Annotated Figure 1, shown below, further illustrates that the **host processor** (processor 108) is directly coupled to the **memory controller** (memory controller 102) via **bus 124** as the controller performs memory read and write transactions:



456. For at least these reasons, it is my opinion that Ware discloses “wherein the host is directly coupled to the controller by a communication path as the controller performs the memory read transaction or memory write transaction.”

- n. [Claim 1m]: “wherein the first device profile and second device profile are each stored in the memory store;”

457. It is my opinion that Ware discloses this claim element. As explained

above, Ware teaches device profiles in the form of “usage fields.” *Supra* § X.B.1.c. Ware also teaches that these usage fields may be stored in the memory store. *See, e.g.*, Ex-1006 at 11:50-55 (noting that usage data may be kept “in the first portion 104 of main memory” and that the usage field “of each page in a main memory 101 can be set based on the accumulated usage data.”). In this manner, Ware teaches that the usage fields (e.g., first device profile and second device profile) in step 300 in Figure 3 of Ware are transferred from secondary storage to main memory 101 during initialization.

o. [Claim 1n]: “and using attributes from said first and second device profiles; and”

458. It is my opinion that Ware, in view of a POSITA’s knowledge, renders this claim element obvious. As explained above, Ware teaches that the first and second device profiles (i.e., usage fields) include a variety of attributes. *See supra* § X.B.1.i. Ware also discloses using the attributes from each of these device profiles. For example, Ware teaches use of attributes such as the ratio of write access time to read access time in each device profile. As Ware explains “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1. This attribute differs from the write access time of, for example, DRAM, which is

typically about the same as the read access time of DRAM devices. For example, the read access time and write access time of a DRAM device are typically both less than 100 ns, and often both are between 20 ns and 40 ns. The ratio of write access time to read access time of DRAM devices is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:19-31.

459. Ware also discloses using the endurance attribute. As Ware explains, endurance is the “maximum number of erase-write cycles that the memory device can support.” *Id.* at 6:28-33. These erase-write cycle count “values” are “updated, maintained and checked to ensure that the endurance limit of the device is not exceeded (e.g., 10 K cycles).” *Id.* at 6:33-35. For example, “prior to performing a write operation,” the erase-write cycled count data is checked “to determine whether to complete the write operation, or perform a contingency operation (e.g., write to a different block of memory or to a different portion of main memory).” *Id.* at 6:36-41. Other measures may also be employed “to ensure that the endurance limitation of the memory devices in a portion 106 of main memory 101 are not exceeded.” *Id.* at 6:60-63.

460. Ware also teaches using a granularity attribute. For example, Ware discloses that “NOR Flash memory devices typically have different granularity for read and write operations. In particular, the minimum unit for reading data from

Flash memory may be a word (e.g., 16 bits) while the minimum unit for writing data to Flash memory is typically significantly larger than the minimum unit for reading, and is called either a block or a group of memory cells. In some Flash memory devices, each block includes between 1024 and 65,536 words, with blocks typically containing between 4 k (4096) and 32 k (32,768) Words.” Ex-1006 at 5:65-6:7.

461. Ware thus teaches that different memory devices have different granularity—i.e., the minimum unit for read and write operations—resulting in the need to select a transfer size for the memory transaction. For example, a POSITA would have known that flash memory can be written and read in pages. *Id.* at 6:9-12 (noting that a page “is the basic unit of data or storage in a virtual memory system”). A POSITA would have also known that a page of data is the number of bits in a row of data in a flash memory that may be transferred during a memory transaction. This page size (transfer size) is an attribute that would have been known to be associated with a device profile. A POSITA would have also known that a group of pages is a block of flash memory. *See e.g., id.* at 6:7-8 (teaching that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes.”). In this manner, Ware teaches that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes.

462. For at least these reasons, it is my opinion that Ware, in view of a POSITA's knowledge, renders obvious "using attributes from said first and second device profiles."

- p. **[Claim 1o]: "selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes."**

463. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches the use of a variety of different attributes, including different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. *Supra* § X.B.1.i, X.B.1.o. Ware further teaches that "[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes," thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. Ex-1006 at 6:7-8.

464. Ware also discloses that "it may be noted that a page, which is the basic unit of data or storage in a virtual memory system is typically, but not necessarily, of a different size than a block in a Flash memory device. Thus, a page may be stored *in a plurality of blocks*, or *one block could store multiple pages*, or *each page could be stored in a single block*." *Id.* at 6:9-14 (emphasis added). In view of these size differences between pages and blocks, which can result in pages being

stored either within a block or across blocks, the transfer size would be a function of the data size and the first set of attributes. *See also supra* § X.B.1.i, X.B.1.o.

465. For at least these reasons, it is my opinion that Ware discloses “selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

2. Claim 5 Is Rendered Obvious by Ware in View of a POSITA’s Knowledge

466. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious claim 5 of the ’190 patent.

a. [Claim 5]: “The method of claim 1, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

467. It is my opinion that Ware renders obvious the method of claim 1 for the reasons discussed above. *Supra* § X.B.1. It is also my opinion that Ware renders obvious the method recited in claim 5.

468. As explained above, Ware teaches shared attributes for the first and second device profiles (i.e., “usage fields”). *Supra* § X.B.1.k. For example, Ware discloses that “[t]he memory devices in portion 104 have one or more attributes that differ from the attributes of the memory devices in portion 106. Examples of such attributes are: the ratio of write access time to read access time, volatility, and

endurance.” Ex-1006 at 4:6-11; *see also id.* at 3:19-30 (describing attributes of Flash and DRAM memory). Accordingly, shared attributes are, for example, the volatilities of memory 104 (DRAM) and memory 106 (Flash). *See id.* at 3:56-61.

469. Moreover, as explained above, a POSITA would have understood that each attribute would have an attribute qualifier, which could be different for each device profile. *Supra* § X.B.1.k. For example, Ware teaches that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns.” *Id.* at 4:19-29. As result, the ratio of write access time to read access time would be different for the two devices. A POSITA would have also understood that the attribute qualifiers for the shared attribute “volatility” for DRAM and Flash are different. More specifically, the DRAM attribute qualifier for the volatility attribute is “volatile,” while the Flash attribute qualifier for the volatility attribute is “non-volatile.” *Id.* at 3:10-12, 30-33; *see also id.* at 3:18:30 (explaining that Flash has “different attributes than DRAM,” such as being “non-volatile (i.e., Flash memory retains its contents without requiring periodic refreshing”). Likewise, a POSITA would understand the attribute “endurance” would have an attribute qualifier of “unlimited” for DRAM and “limited” for Flash. *Id.* at 3:18-30; *supra* § X.B.1.o. Accordingly, as explained above, it is my opinion that use of attribute

qualifiers would have been obvious to a POSITA. *Supra* § X.B.1.k.

470. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “method of claim 1, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

3. Claim 6 Is Rendered Obvious by Ware in View of a POSITA’s Knowledge

471. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious claim 6 of the ’190 patent.

a. [Claim 6]: “The method of claim 5, wherein said different attribute qualifiers include different attribute values.”

472. It is my opinion that Ware renders obvious the method of claim 5 for the reasons discussed above. *Supra* § X.B.2.a. It is also my opinion that Ware renders obvious the method recited in claim 6.

473. As explained above, the use of different attribute qualifiers for shared attributes would have been obvious to a POSITA in view of Ware. For example, it would have been obvious to a POSITA that the “volatility” attribute would have different qualifiers such as “volatile” and “non-volatile,” which are different values. *Supra* § X.B.2.a.; *see also supra* § X.B.1.k (explaining that volatility would be represented by either 1 or 0, indicating that the memory is either volatile (“yes”)

or non-volatile (“no”)). Likewise, it would have been obvious that the “endurance” attribute would have different attribute values, such as “unlimited” and “limited.”

Supra § X.B.2.a.

474. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “method of claim 5, wherein said different attribute qualifiers include different attribute values.”

4. Claim 29 Is Rendered Obvious by Ware in View of a POSITA’s Knowledge

475. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious claim 29 of the ’190 patent.

- a. [Claim 29 pre]: “A non-transitory computer readable medium comprising computer executable instructions adapted to cause a method of performing memory transactions on a first memory location and a second memory location, said method comprising:”**

476. To the extent the preamble is limiting, it is my opinion that Ware discloses this claim element. For example, Ware discloses “memory systems and methods” utilizing “a memory controller for a non-homogeneous memory system.” Ex-1006 at 1:6-8; *see also id.*, claims 23-31 (reciting a “method of managing memory in a non-homogenous memory system”). A method of performing read and write transactions on set of memory locations is also illustrated in Figure 3, reproduced below, which provides a flow diagram of “a non-homogeneous memory

management process.” *Id.* at 11:62-63 (teaching a method for performing read and write transactions on a set of memory locations that includes different types of memory).

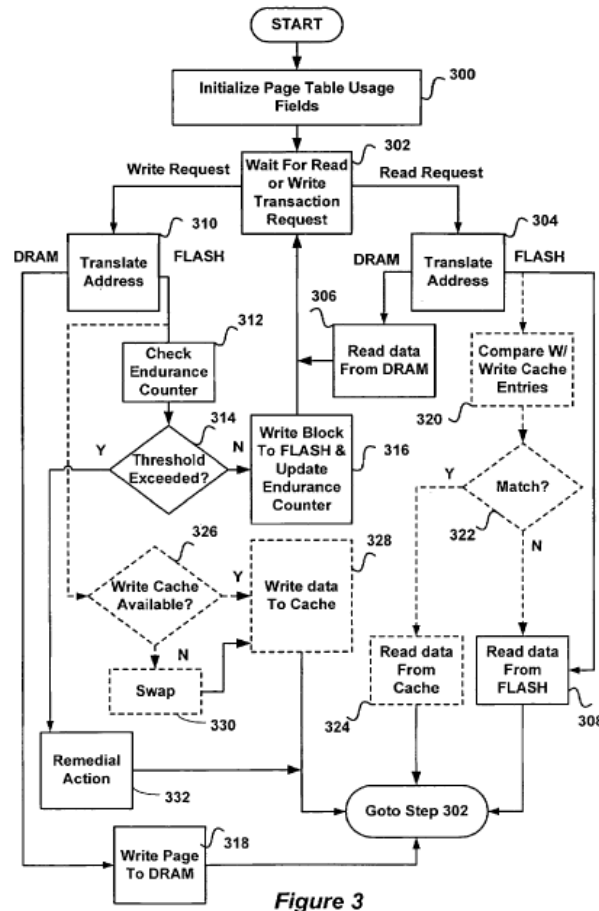
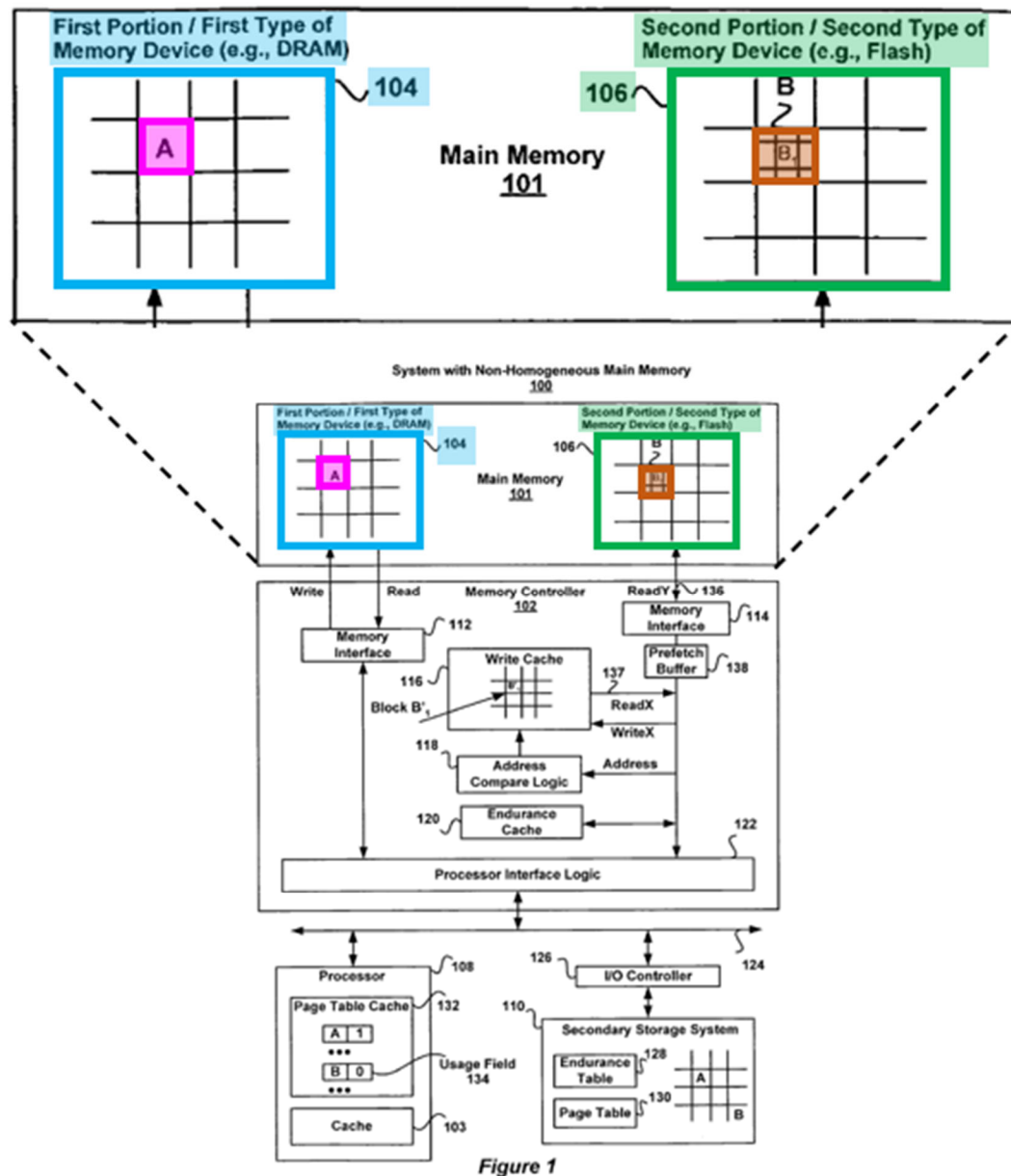


Figure 3

477. Ware also discloses performing memory transactions on a set of memory locations that includes a first memory location and a second memory location. For example, Figure 1 discloses a “main memory 101 [that] includes *a first portion 104* implemented with one or more memory devices of a first type (*e.g., DRAM*), and a *second portion 106* implemented with one or more memory devices

of a second type (*e.g., NOR Flash, often simply called Flash memory*).” *Id.* at 3:56-61 (emphasis added). Ware further teaches that “a usage field value of ‘A’ represents *a page mapped to the first portion 104 of main memory 101*, while a value of ‘B’ represents *a page mapped to the second portion 106 of main memory 101*.” *Id.* at 5:57-61(emphasis added).

478. Annotated Figure 1, shown below, illustrates the **first memory location** (field value A in a **first memory portion 104**) and a **second memory location** (field value B in a **second memory portion 106**):



479. Ware also discloses a non-transitory computer readable medium comprising computer executable instructions adapted to cause this method to be performed. For example, Ware discloses that computer system 100, illustrated above, comprises a secondary storage system that can be “any type of file storage

device” such as “hard disk units, optical disks, Universal Serial Bus (USB) Flash, storage area networks (SANs), a wireless connection to a file server system or wireless local area network (WLAN), and any other file storage device.” *Id.* at 4:61-5:1. A POSITA would readily understand that these devices comprise a non-transitory computer readable medium.

480. Ware further discloses that this non-transitory computer readable medium comprises computer executable instructions adapted to cause the method to be performed. For example, Ware teaches that the computer readable medium is used for “storing various forms of *programs (e.g., source, object, executable)* and temporary storage of virtual pages (e.g., swap space).” *Id.* at 5:3-6; *see also id.* at 1:25-26 (explaining that “[p]ages of data and *program code* are transferred from secondary storage to main memory”); *id.* at 5:9-12 (explaining that the secondary storage system includes a file management system with computer readable instructions “for providing mapping between the logical and physical views of a file via one or more services and the I/O controller 126”) (all emphasis added). The secondary storage can also “store[] computer programs and data, including *an operating system 410*, which includes a virtual memory management module or procedures 412, as well as application programs 432.” *Id.* at 10:33-37. A POSITA would readily understand that the operating system would comprise computer

readable instruction adapted to cause a method of performing memory transactions on a first and second memory location.

481. Annotated Figure 4, shown below, illustrates the **computer readable medium** (secondary storage 110), which stores computer executable instructions—including an operating system—adapted to cause read and write transactions to be performed:

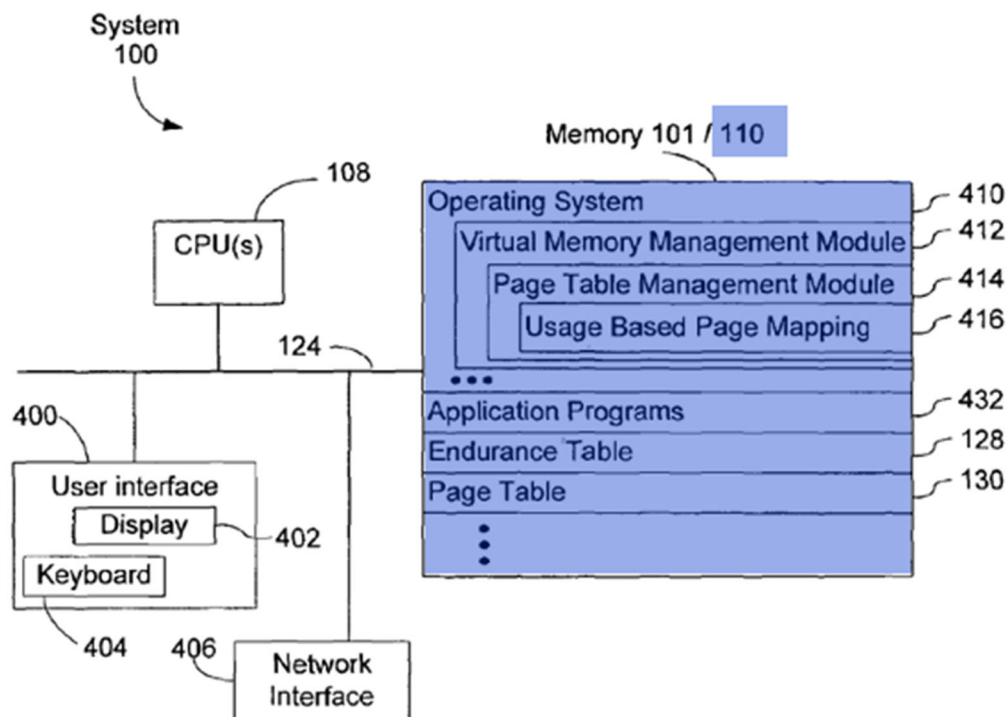


Figure 4

482. For at least these reasons, it is my opinion that Ware discloses “[a] non-transitory computer readable medium comprising computer executable instructions adapted to cause a method of performing memory transactions on a first memory

location and a second memory location.”

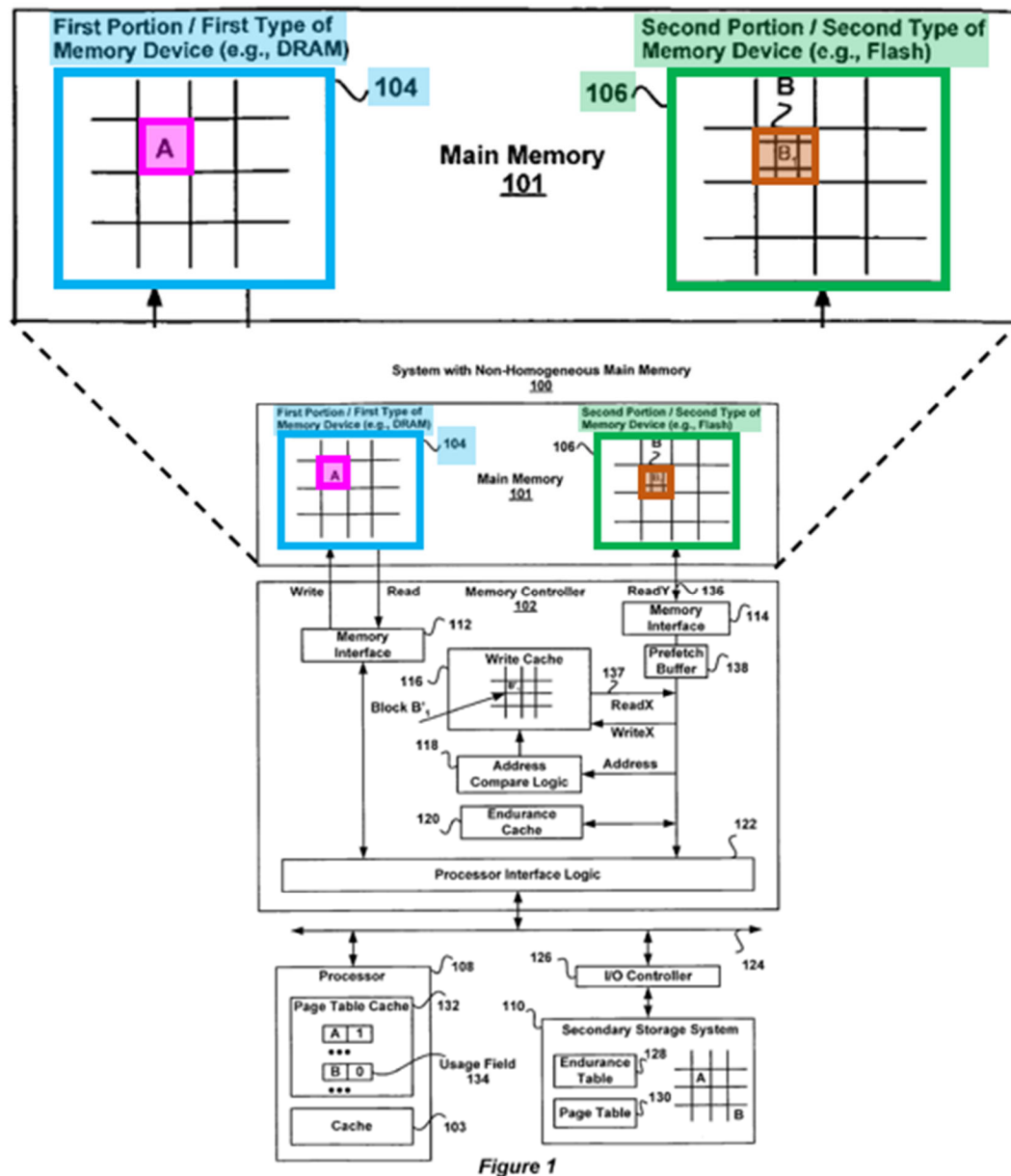
- b. [Claim 29a]: “performing a memory transaction comprising addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile;”**

483. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a memory store comprising a first memory location (first memory portion 104) and a second memory location (second memory portion 106). Ex-1006 at 3:56-61; *supra* § X.B.1.a. The first memory portion may be “implemented using one or more DRAM memory devices,” while the second memory portion may be “implemented using one or more Flash memory devices (e.g., NOR Flash memory devices).” Ex-1006 at 4:38-42; *see also id.* at 7:22-25 (noting that a memory controller includes interfaces for coupling memory “in two or more distinct portions (104, 106)”; *supra* § X.B.1.a.

484. Ware also teaches performing a memory transaction comprising addressing a first memory location in the first memory device. For example, Ware discloses that “[d]ata can be distributed among the first and second portions 104, 106 of main memory 101.” *Id.* at 3:64-67. Ware also teaches that “a usage field value of ‘A’ represents a page mapped to the first portion 104 of main memory”—i.e., the first memory location. Ex-1006 at 5:57-61. A POSITA would readily

understand that performing read and write transactions to the first memory location in the first memory device, as described by Ware, would require addressing that location. *See id.* at 7:22-25 (explaining that the memory device interfaces “include signal conditioning circuitry and other devices for transmitting addresses, data, and control signals to and from the memory devices in main memory portions 104 and 106”); *id.* at 7:46-49 (noting that the processor interface logic may include circuitry “for transmitting data, address, and control signals”).

485. Annotated Figure 1, below, illustrates the **first memory location** (field value A in a **first memory portion 104**) and a **second memory location** (field value B in a **second memory portion 106**):



486. Ware also teaches that the first memory location and the second memory location are respectively associated with a first device profile and a second device profile. For example, Ware teaches “one or more first memory devices of a first memory type having a *first set of attributes*, and a second memory interface

adapted to be coupled to one or more second memory devices of a second memory type having a *second set of attributes*.” Ex-1006, Abstract (emphasis added); claim 15 (reciting “one or more first memory devices . . . having a first set of attributes” and “one or more second memory devices . . . having a second set of attributes”). More specifically, Ware discloses a “usage field 208” that “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212*.” *Id.* at 11:1-4 (emphasis added). In one embodiment of Ware, such as Figure 2, the usage model is a one-bit flag that indicates whether the requested page is expected to be read-only or read-write. However, Ware discloses other embodiments: “In the above example, the usage field 208 is a one-bit flag that indicates whether the requested page is expected to be read-only or whether it is to be read-Write. It should be apparent, however, that the usage field 208 can represent other page usage models, and include more or fewer bits.” *Id.* at 11:19-23. In this manner, Ware teaches a first device profile, or set of attributes, associated with a first memory location (the first memory portion comprising DRAM) and a second device profile, or set of attributes, associated with a second memory location (the second memory portion comprising Flash).

487. Annotated Figure 2, shown below, illustrates the **device profile** (i.e.,

the usage field used to store a first and second sets of attributes), which is associated with both the **first memory location** (in the first memory device) and the **second memory location** (in the second memory device):

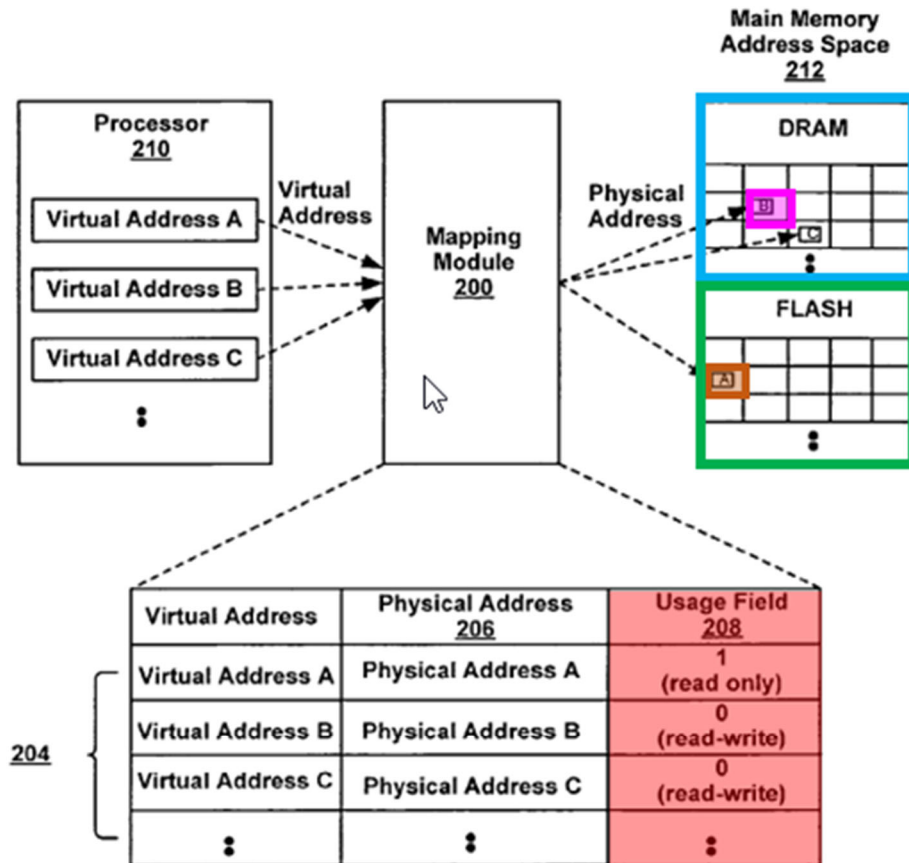


Figure 2

488. For at least these reasons, it is my opinion that Ware discloses “performing a memory transaction comprising addressing a first memory location, said first memory location and a second memory location are respectively associated

with a first device profile and a second device profile.”

- c. **[Claim 29b]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”**

489. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a first memory location that comprises DRAM having a device profile. A POSITA would have understood that a random read command results in accessing a byte of data at some random location in the memory. Ware also teaches that “[m]ain memory is typically implemented with Dynamic Random [data type] Access Memory (DRAM),” which “has a near-*optimal combination of operational attributes*”—comprising a device profile—“for implementing main memory.” Ex-1006, 1:65-2:6; *see also id.* at 11:1-4 (disclosing a “usage field 208” or profile that “stores data indicative of the usage model of the page, which can be used by processor 210 to *exploit the unique attributes of a physical memory device*,” such as DRAM’s suitability for random data”) (all emphasis added). Thus, a POSITA would understand that the device profile for DRAM would be optimal for a random data type.³⁵

³⁵ Similarly, a POSITA would have understood that a second device profile for flash memory, particularly NAND flash, utilizes page read commands that result in

490. For at least these reasons, it is my opinion that Ware discloses “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- d. **[Claim 29c]: “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

491. It is my opinion that Ware discloses this claim element. Ware teaches that, “[a]fter the initialization phase is complete, the memory controller waits [at] 302 for a read or write transaction request.” Ex-1006 at 12:12-14. A person of ordinary skill would readily understand that performing the read or write transaction request requires identifying command details for causing the memory transaction to be performed.

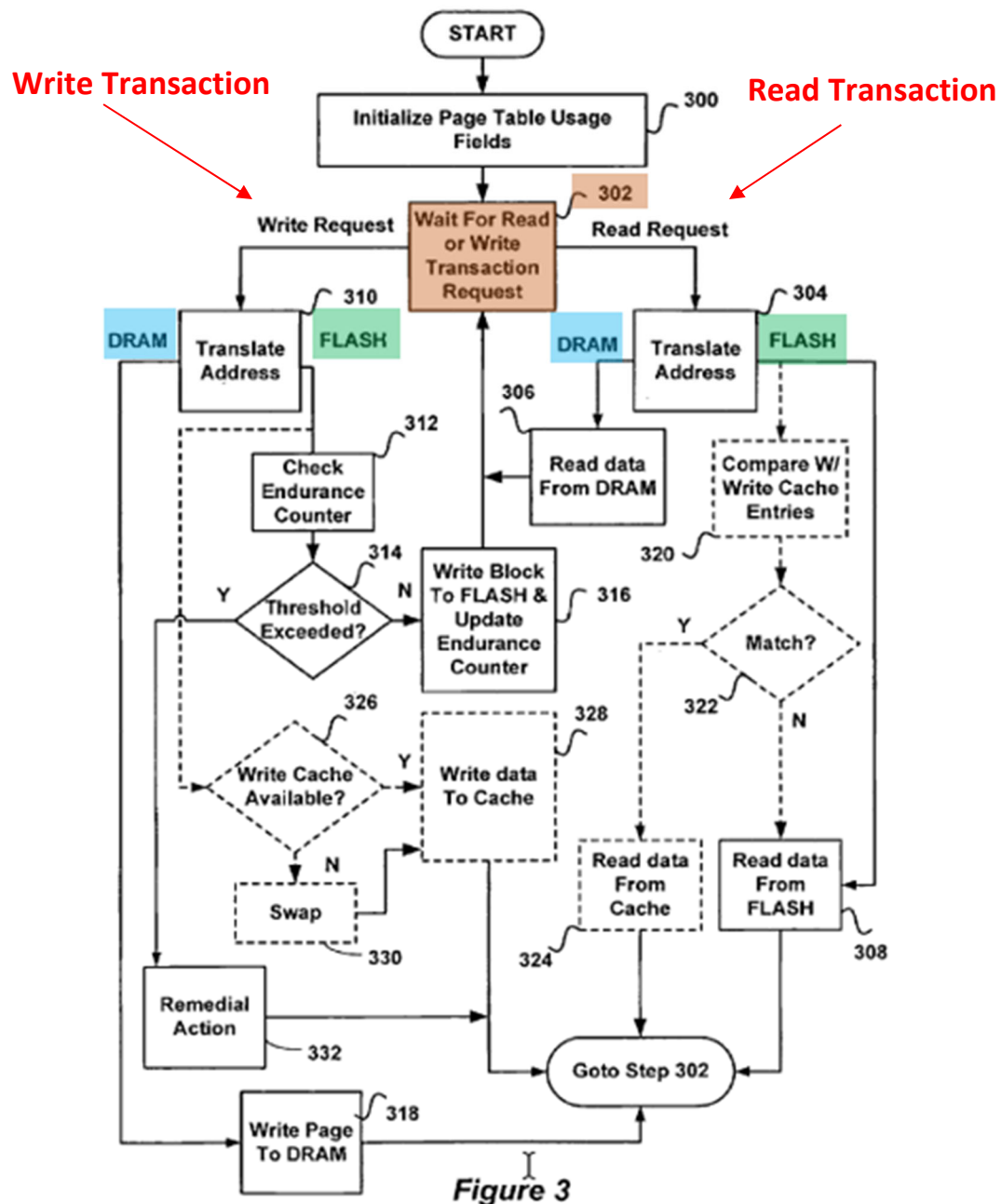
492. For example, Ware teaches that “[i]f a read request is received, the virtual address is translated into a physical address at step 304.” *Id.* at 12:14-15. Moreover, “[o]ne aspect of step 304 is reading the usage field for the page. If the

sequentially accessing bytes of data located in a row location (page) of the memory. *See supra* note 6. As a result, flash memory would have a device that is optimal for a sequential data type.

usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be read from DRAM, then the data (which may contain program code) is read at step 306 from DRAM and returned to the requestor.” *Id.* at 12:17-22. Likewise, “[i]f the transaction request is a write request, the virtual address is translated into a physical address at step 310, which once again includes reading the usage field for the page containing the specified address. If the usage field indicates that the write data should be written to DRAM (i.e., to a first portion of main memory), then the write data is written to DRAM at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45.

493. A POSITA would understand that command details would be required for executing these steps of read or write request. As such, Ware requires identifying command details for identifying a first memory location and causing a memory transaction to be performed on the first memory location.

494. Annotated Figure 3, below, further illustrates determining a read or write commence at step 302, which requires **identifying command details** for causing the memory transaction to be performed for the **first memory device** (DRAM):



495. For at least these reasons, it is my opinion that Ware teaches “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details

comprising the first memory device.”

- e. **[Claim 29d]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

496. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses that both the first memory location (first memory portion 104) and second memory location (second memory portion 106) have a set of attributes comprising a device profile. *See supra* § X.B.1.c. Ware further teaches that “[t]he memory devices in portion 104 have one or more *attributes that differ* from the attributes of the memory devices in portion 106.” Ex-1006 at 4:4-9 (emphasis added); *see also id.* at Abstract (“The first and second sets of attributes have *at least one differing attribute.*”); *id.* at 2:21-25 (“The memory devices . . . include two or more memory device types (e.g., DRAM, Flash) *with different attributes.*”); *id.* at 4:19-29 (explaining that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns); *id.*, claim 15 (reciting a first memory type and second memory type having attributes, “wherein the first and second sets of attributes have *at least one differing attribute*”) (all emphasis added). The differences in these attributes demonstrate that a difference

exists in the device profiles for the first and second memory locations.

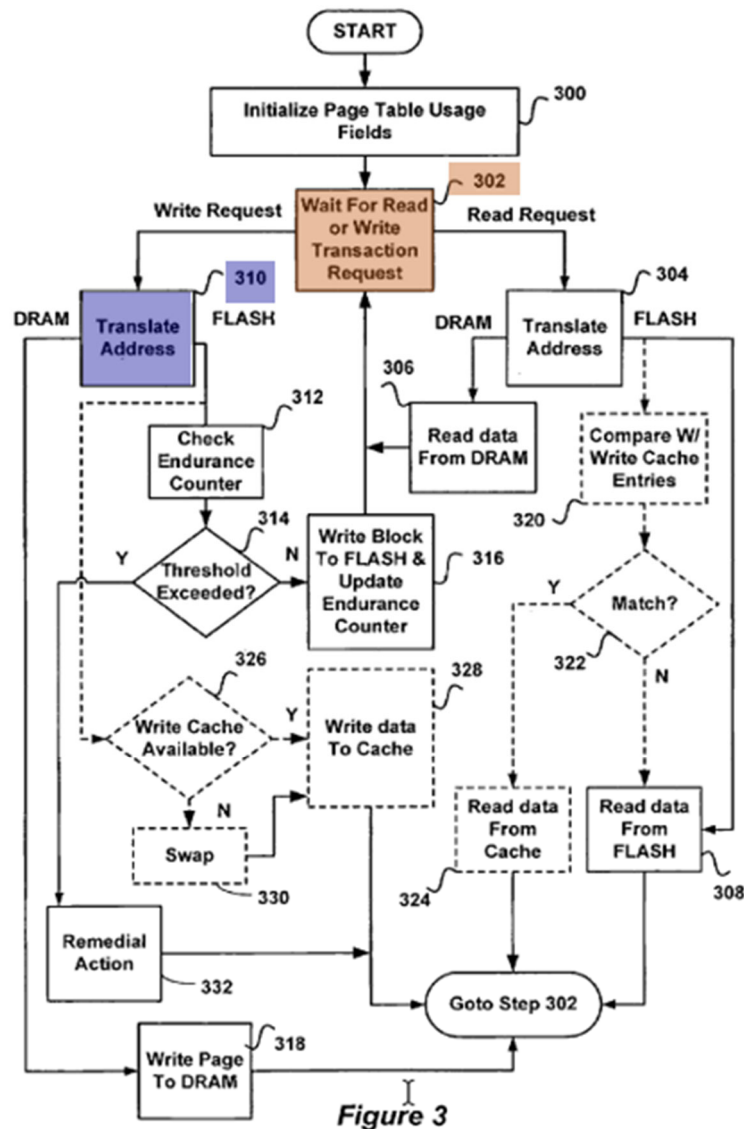
497. For at least these reasons, it is my opinion that Ware discloses “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

f. [Claim 29e]: “after identifying the command details, obtaining the first set of attributes;”

498. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches identifying command details for causing a memory transaction to be performed. *Supra* § X.B.1.h; *see also supra* § X.B.1.i. Ware further teaches that the first of attributes are obtained after the command details are identified. For example, in relation to Figure 3, reproduced below, Ware teaches that once it has been determined that a transaction request is a write request at step 302—i.e., once the command details for a write transaction have been identified—**then** a virtual address is “translated into a physical address at step 310, which once again **includes reading the usage field** for the page containing the specified address.” Ex-1006 at 12:38-41; *see also id.* at 12:41-45 (explaining that the usage field then indicates whether the write data should be written to DRAM). The usage field “stores data indicative of the usage model of the page, which can be used by processor 210 to

exploit the *unique attributes of a physical memory device associated with the main memory address space 212.*” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches that, after identifying the command details for a read or write (as shown in step 302 of annotated Figure 3 below), a first set of attributes in the usage field, which describe the corresponding usage model for the first memory location in memory device 104, are obtained (as shown in step 310 of annotated Figure 3 below).

499. Annotated Figure 3, below, illustrates that—after the **command details are identified** in step 302—the address is translated in step 310, which requires **obtaining the set of attributes**:



500. For at least these reasons, it is my opinion that Ware discloses “after identifying the command details, obtaining the first set of attributes.”

g. [Claim 29f]: “and using attributes from said first and second device profiles; and”

501. It is my opinion that Ware in view of a POSITA’s knowledge, renders this claim element obvious. As explained above, Ware teaches that the first and

second device profiles (i.e., usage fields) include a variety of attributes. *See supra* § X.B.1.i. Ware also discloses using the attributes from each of these device profiles. For example, Ware teaches that “NOR Flash memory devices typically have different granularity for read and write operations. In particular, the minimum unit for reading data from Flash memory may be a word (e.g., 16 bits) while the minimum unit for writing data to Flash memory is typically significantly larger than the minimum unit for reading, and is called either a block or a group of memory cells. In some Flash memory devices, each block includes between 1024 and 65,536 words, with blocks typically containing between 4 k (4096) and 32 k (32,768) Words.” Ex-1006 at 5:65-6:7.

502. Ware thus teaches that different memory devices have different granularity—i.e., the minimum unit for read and write operations—resulting in the need to select a transfer size for the memory transaction. For example, a POSITA would have known that flash memory can be written and read in pages. *Id.* at 6:9-12 (noting that a page “is the basic unit of data or storage in a virtual memory system”). A POSITA would have also known that a page of data is the number of bits in a row of data in a flash memory that may be transferred during a memory transaction. This page size (transfer size) is an attribute that would have been known to be associated with a device profile. A POSITA would have also known that a

group of pages is a block of flash memory. *See e.g., id.* at 6:7-8 (teaching that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes”). In this manner, Ware teaches that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes.

503. Similarly, Ware teaches use of attributes such as the ratio of write access time to read access time in each device profile. For example, “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1. This attribute differs from the write access time of, for example, DRAM, which is typically about the same as the read access time of DRAM devices. For example, the read access time and write access time of a DRAM device are typically both less than 100 ns, and often both are between 20 ns and 40 ns. The ratio of write access time to read access time of DRAM devices is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:19-31.

504. For at least these reasons, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious “using attributes from said first and second device profiles.”

h. [Claim 29g]: “selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

505. It is my opinion that Ware discloses this claim element. As explained Ware teaches the use of a variety of different attributes, including different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. *Supra* §§ X.B.1.i, X.B.1.o. Ware further teaches that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes,” thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. Ex-1006 at 6:7-8.

506. Ware also discloses that “it may be noted that a page, which is the basic unit of data or storage in a virtual memory system is typically, but not necessarily, of a different size than a block in a Flash memory device. Thus, a page may be stored *in a plurality of blocks*, or *one block could store multiple pages*, or *each page could be stored in a single block*.” *Id.* at 6:9-14 (emphasis added). In view of these size differences between pages and blocks, which can result in pages being stored either within a block or across blocks, the transfer size would be a function of the data size and the first set of attributes. *See also supra* §§ X.B.1.i, X.B.1.o.

507. For at least these reasons, it is my opinion that Ware discloses

“selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

5. Claim 36 Is Rendered Obvious by Ware in View of a POSITA’s Knowledge

508. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 36 of the ’190 patent.

- a. [Claim 36]: “The computer readable medium of claim 29, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

509. It is my opinion that Ware renders obvious the computer readable medium of claim 29 for the reasons discussed above. *Supra* § X.B.1. It is also my opinion that Ware renders obvious the computer readable medium of claim 36.

510. As explained above, Ware teaches shared attributes for the first and second device profiles (i.e., usage fields”). *Supra* § X.B.1.k. For example, Ware discloses that “[t]he memory devices in portion 104 have one or more attributes that differ from the attributes of the memory devices in portion 106. Examples of such attributes are: the ratio of write access time to read access time, volatility, and endurance.” Ex-1006 at 4:6-11; *see also id.* at 3:19-30 (describing attributes of Flash and DRAM memory). Accordingly, shared attributes are, for example, the volatilities of memory 104 (DRAM) and memory 106 (Flash). *See id.* at 3:56-61.

511. Moreover, as explained above, a POSITA would have understood that each attribute would have an attribute qualifier, which would be different for each device profile. *Supra* § X.B.1.k. For example, Ware teaches that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns.” *Id.* at 4:19-29. As result, the ratio of write access time to read access time would be different for the two devices. A POSITA would have also understood that the attribute qualifiers for the shared attribute “volatility” for DRAM and Flash are different. More specifically, the DRAM attribute qualifier for the volatility attribute is “volatile,” while the Flash attribute qualifier for the volatility attribute is “non-volatile.” *Id.* at 3:10-12, 30-33; *see also id.* at 3:18:30 (explaining that Flash has “different attributes than DRAM,” such as being “non-volatile (i.e., Flash memory retains its contents without requiring periodic refreshing”). Likewise, a POSITA would understand the attribute “endurance” would have an attribute qualifier of “unlimited” for DRAM and “limited” for Flash. *Id.* at 3:18-30; *supra* § X.B.1.o. Accordingly, as explained above, it is my opinion that use of attribute qualifiers would have been obvious to a POSITA. *Supra* § X.B.1.k.

512. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “the computer readable medium of

claim 29, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

6. Claim 37 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

513. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 37 of the ’190 patent.

a. [Claim 37]: “The computer readable medium of claim 36, wherein said different attribute qualifiers include different attribute values.”

514. It is my opinion that Ware renders obvious the method of claim 36 for the reasons discussed above. *Supra* § X.B.2.a. It is also my opinion that Ware renders obvious the method recited in claim 37.

515. As explained above, the use of different attribute qualifiers for shared attributes would have been obvious to a POSITA in view of Ware. For example, it would have been obvious to a POSITA that the “volatility” attribute would have different qualifiers such as “volatile” and “non-volatile,” which are different values *Supra* § X.B.2.a; *see also supra* § X.B.1.k (explaining that volatility would be represented by either 1 or 0, indicating that the memory is either volatile (“yes”) or non-volatile (“no”). Likewise, it would have been obvious that the “endurance” attribute would have different attribute values such as “unlimited” and “limited.” *Supra* § X.B.2.a.

516. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “computer readable medium of claim 36, wherein said different attribute qualifiers include different attribute values.”

7. Claim 45 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

517. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 45 of the ’190 patent.

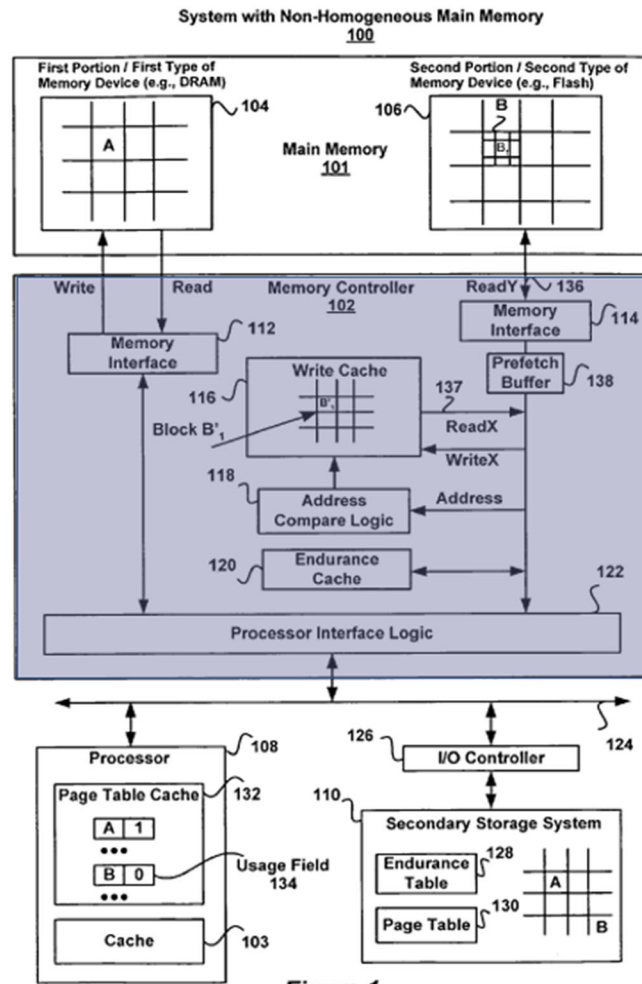
a. [Claim 45 pre]: “A memory controller for performing memory transactions, said memory controller comprising:”

518. To the extent the preamble of claim 45 is limiting, it is my opinion that Ware discloses this claim element. Ware is directed “to a memory controller for a non-homogenous memory.” Ex-1006 at 1:7-8; *see also id.* at 2:18-21 (describing a “memory controller for a non-homogenous memory system [that] is configurable to facilitate page operations between a virtual memory address space and physical pages of memory devices in the memory system”); claim 1 (reciting a controller with “a first memory interface adapted to be coupled to one or more first memory devices” and “a second memory interface adapted to be coupled to one or more second memory devices.”)

519. Ware further discloses that “[t]he memory controller 102 includes memory interfaces 112, 114 for coupling the memory controller 102 to the memory

devices in two or more distinct portions (104, 106) of main memory 101, and processor interface logic 122.” *Id.* at 7:21-25. The memory controller “can be configured to exploit the different attributes of the memory device types in the respective portions of main memory so as to improve one or more aspects of memory system performance.” *Id.* at 4:47-52; claim 1 (reciting that the controller comprises interface logic “configured to direct memory transactions”); claim 7 (reciting that the controller “is configurable to write data stored therein to the one or more second memory devices”). As such, Ware teaches that the memory controller performs memory transactions.

520. Annotated Figure 1, shown below, illustrates the **memory controller** (memory controller 102) for performing memory transactions:



521. For at least these reasons, it is my opinion that Ware discloses a “memory controller for performing memory transactions.”

- b. **[Claim 45a]: “means for performing a memory transaction comprising a means for addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile;”**

522. It is my opinion that Ware discloses this claim element. As explained above, this claim element contains two means plus function limitations. I address

each of these means plus function limitations in turn.

523. The first means-plus-function element recited in the claim is a “means for performing a memory transaction.” As explained above, this term has a recited function of “performing a memory transaction.” *Supra* § VIII.B. Pursuant to Petitioner’s proposed construction, the structure for performing this function is a “memory controller or similar structure configured to perform a memory transaction.”

Id. Ware discloses both the recited function and structure.

524. For example, Ware discloses “a memory controller for a non-homogenous memory,” which “direct[s] memory transactions” to first and second memory interfaces. Ex-1006, Abstract; 1:7-8; *see also id.* at 2:18-21 (“A memory controller for a non-homogenous memory system is configurable to facilitate page operations between a virtual memory address space and physical pages of memory devices in the memory system”); 7:22-8:6 (describing structure and function of memory controller).

525. Accordingly, it is my opinion that Ware discloses means for performing a memory transaction—i.e., a “memory controller or similar structure configured to perform a memory transaction.”

526. The second means-plus-function element recited in this claim is a “means for addressing a first memory location.” As explained above, this term has

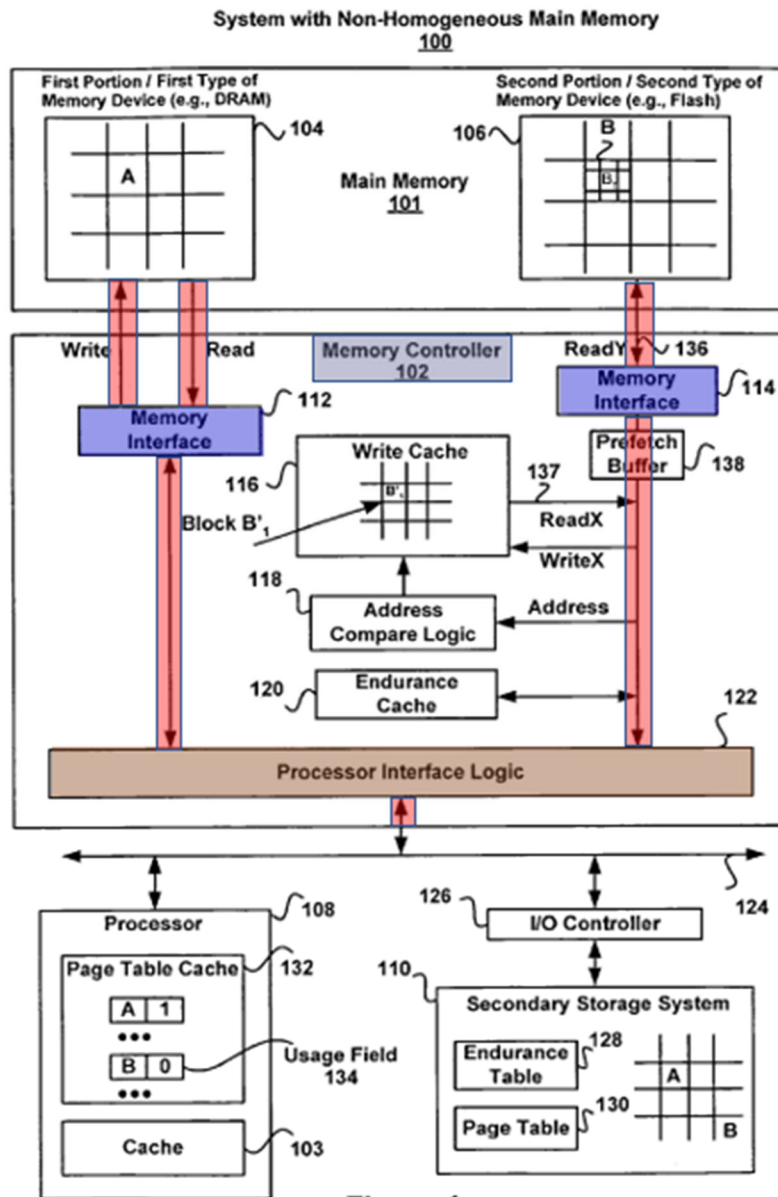
a recited function of “addressing a first memory location.” *Supra* § VIII.D. Pursuant to Petitioner’s construction, the structure for performing this function is “a memory device interface, a host interface, an interface controller or similar structure configured to process memory transaction requests, and associated communication pathways.” *Id.* Ware discloses both the recited function and structure.

527. Ware discloses a memory device interface that is associated with addressing a first memory location. In particular, Ware discloses “memory interfaces 112, 114 for coupling the memory controller 102 to the memory devices in two or more distinct portions.” *Id.* at 7:22-25. The memory interfaces “include signal conditioning circuitry and other devices for transmitting addresses, data, and control signals to and from memory devices.” *Id.* at 7:25-29

528. Ware also discloses a host interface and interface controller. Ware explains that the host processor “is coupled to the memory controller 102 via bus 124 and processor interface logic 122.” *Id.* at 7:9-10. The processor interface logic “may include signal conditioning circuitry and other devices for transmitting data, addresses and control signals to and from” the host on bus 124. *Id.* at 7:46-49. The processor interface logic may also “include decoding logic for decoding one or more bits of a physical address to determine whether the first or second portion of main memory will receive a memory transaction request, as described below with respect

to FIG. 2.” *Id.* at 7:49-54; *see also id.*, claim 1 (reciting a controller comprising “interface logic coupled to the first and second interfaces and configured to direct memory transactions”). Accordingly, Ware confirms that the processor interface logic includes a host interface and is an interface controller or similar structure configured to process memory transactions requests. Ware further discloses associated communication pathways for these components.

529. Annotated Figure 1, below, illustrates the **memory device interfaces**, a host interface that is part of an **interface controller** (processor interface logic, as explained below), and **associated communication pathways**.



530. Ware also teaches that the first and second memory locations are respectively associated with a first device profile and a second device profile as explained above for element [1b]. *Supra* § X.B.1.c.

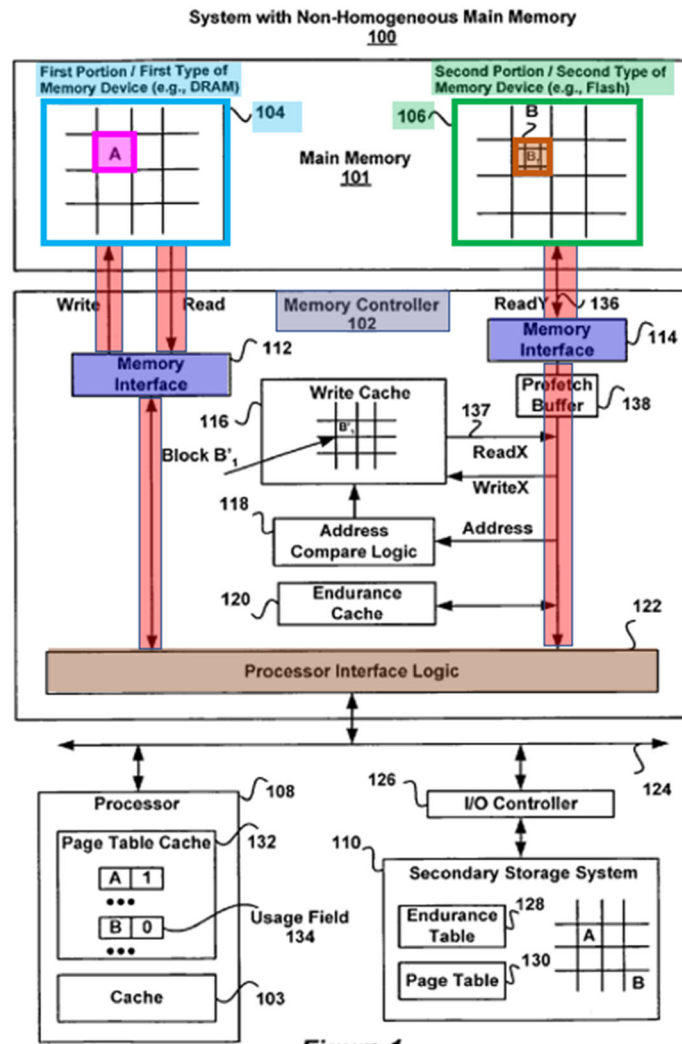
531. For at least these reasons, it is my opinion that Ware discloses “means

for performing a memory transaction comprising a means for addressing a first memory location, said first memory location and a second memory location are respectively associated with a first device profile and a second device profile” pursuant to Petitioners’ proposed construction.

c. **[Claim 45b]: “said means for addressing is directly coupled to the first memory location and second memory location;”**

532. It is my opinion that Ware discloses this claim limitation. Specifically, “[t]he memory devices in the first and second portions 104, 106 of *main memory* *are coupled* to the memory controller 102 *via memory interfaces 112 and 114*, respectively.” Ex-1006 at 4:4-6 (emphasis added); *see also id.* at 7:22-25 (explaining that the memory controller includes “memory interfaces 112, 114 for coupling the memory controller 102 to the memory devices.”).

533. Annotated Figure 1 illustrates that the **memory interfaces** (which are a component of the means for addressing structure) are **directly coupled** to the **first memory location** (in first memory device 104) and the **second memory location** (in second memory device 106):



534. For at least these reasons, it is my opinion that Ware discloses “said means for addressing is directly coupled to the first memory location and second memory location.”

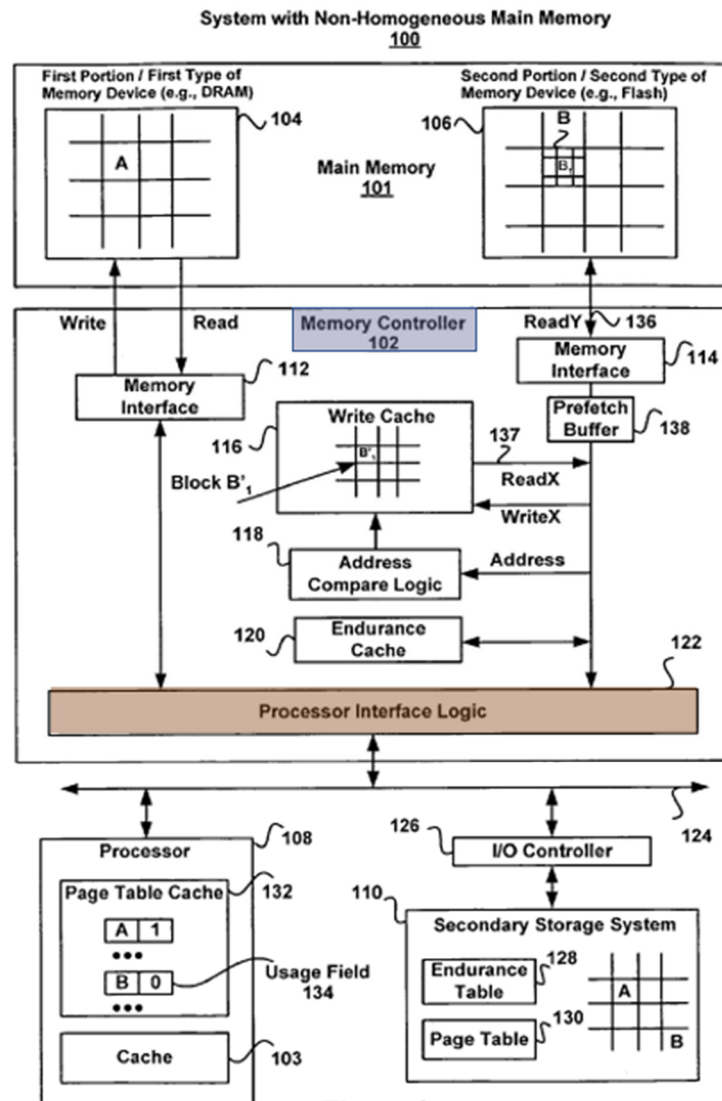
- d. [Claim 45c]: “wherein said means for addressing includes an interface controller means for processing memory transaction requests;”

535. It is my opinion that Ware discloses this claim limitation. As explained

above, “interface controller means for processing memory transaction requests” is a means-plus-function term with a recited function of “processing memory transaction requests.” *Supra* § VIII.C. Pursuant to Petitioner’s proposed construction, the structure for performing this function is an “interface controller or similar structure configured to process memory transaction requests.” *Id.* Ware discloses both the recited function and structure.

536. As previously explained, the processor interface logic “may include signal conditioning circuitry and other devices for transmitting data, addresses and control signals to and from” the host on bus 124. *Id.* at 7:46-49. The processor interface logic may also “include decoding logic for decoding one or more bits of a physical address to determine whether the first or second portion of main memory will receive a memory transaction request, as described below with respect to FIG. 2.” *Id.* at 7:49-54; *see also id.*, claim 1 (reciting a controller comprising “interface logic coupled to the first and second interfaces and configured to direct memory transactions”). A POSITA would readily understand that the processor interface logic includes a host interface and, based on Ware’s description, is an interface controller or similar structure configured to process memory transactions requests.

537. Annotated Figure 1, below, illustrates the **interface controller** (processor interface logic 122):



538. For at least these reasons, it is my opinion that Ware discloses “wherein said means for addressing includes an interface controller means for processing memory transaction requests.

- e. [Claim 45d]: “said means for addressing is directly coupled to a host as said means for addressing addresses the first memory location or second memory location;”

539. It is my opinion that Ware discloses this claim element. As explained above, the memory device interfaces—which are components of the means for addressing—are directly coupled to the host and the first and second memory locations are addressed. For example, Ware discloses that the memory interfaces “include signal conditioning circuitry and other devices for transmitting addresses, data, and control signals to and from memory devices.” Ex-1006 at 7:25-29.

540. Further, Ware’s disclosure of the processor interface logic demonstrates that it is coupled to the host, through the host interface, while the memory locations are addressed. For example, the processor interface logic includes “signal conditioning circuitry and other devices for transmitting data, addresses and control signals to and from” the host on bus 124. *Id.* at 7:46-49. The processor interface logic may also “include decoding logic for decoding one or more bits of a physical address to determine whether the first or second portion of main memory will receive a memory transaction request, as described below with respect to FIG. 2.” *Id.* at 7:49-54; *see also id.*, claim 1 (reciting a controller comprising “interface logic coupled to the first and second interfaces and configured to direct memory transactions”).

541. Annotated Figure 1, below, demonstrates the means for addressing—i.e., the **memory device interfaces**, a host interface that is part of an **interface**

controller (processor interface logic), and **associated communication pathways**—
is **directly coupled** (via bus 124) to the **host processor** as the **first** or **second**
memory location are addressed:

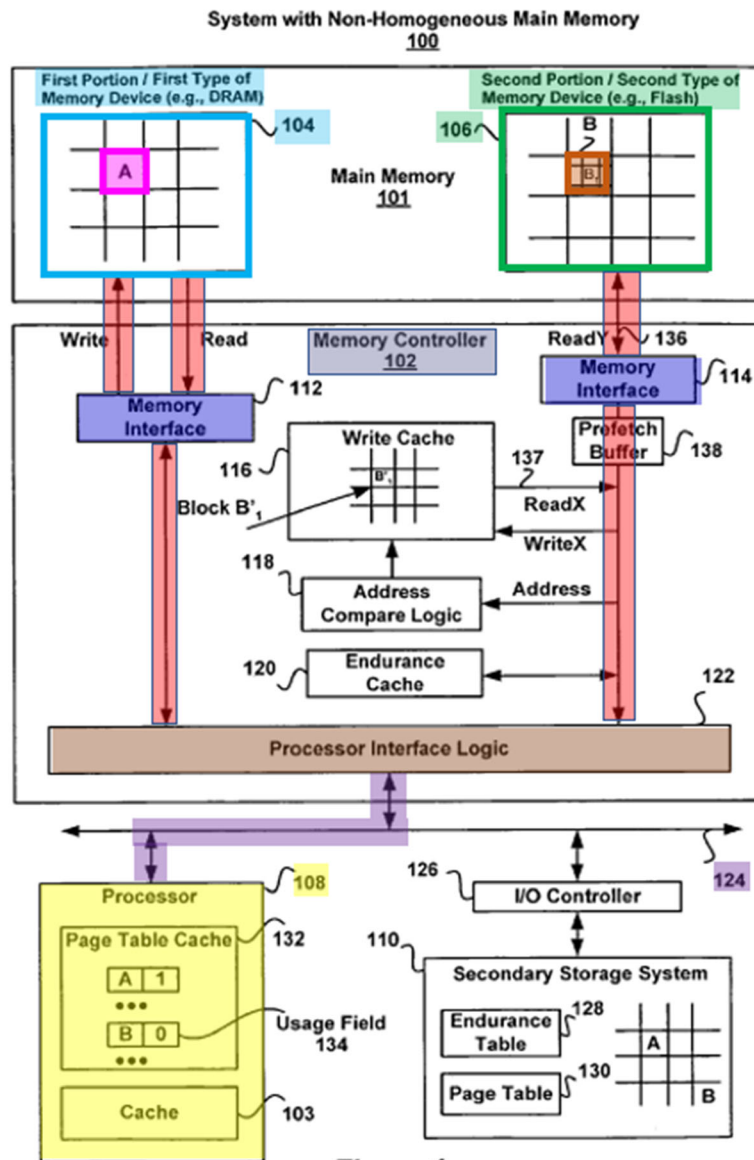


Figure 1

542. For at least these reasons, it is my opinion that Cornwell discloses “said means for addressing is directly coupled to a host as said means for addressing

addresses the first memory location or second memory location.”

- f. **[Claim 45e]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”**

543. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a first memory location comprises DRAM having a device profile. A POSITA would have understood that a random read command results in accessing a byte of data at some random location in the memory. Thus, a POSITA would understand that the device profile for DRAM would be optimal for a random data type. Indeed, Ware teaches that “[m]ain memory is typically implemented with Dynamic Random [data type] Access Memory (DRAM),” which “has a near-*optimal combination of operational attributes*”—comprising a device profile—“for implementing main memory.” Ex-1006, 1:65-2:6; *see also id.* at 11:1-4 (disclosing a “usage field 208” or profile that “stores data indicative of the usage model of the page, which can be used by processor 210 to *exploit the unique attributes of a physical memory device*,” such as DRAM’s suitability for random data.”) (all emphasis added).³⁶

³⁶ Similarly, a POSITA would have understood that a second device profile for flash memory, particularly NAND flash, utilizes page read commands that result in

544. For at least these reasons, it is my opinion that Ware discloses “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- g. **[Claim 45f]: “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

545. It is my opinion that Ware discloses this claim element. Ware teaches that, “[a]fter the initialization phase is complete, the memory controller waits [at] 302 for a read or write transaction request.” Ex-1006 at 12:12-14. A person of ordinary skill would readily understand that performing the read or write transaction request requires identifying command details for causing the memory transaction to be performed.

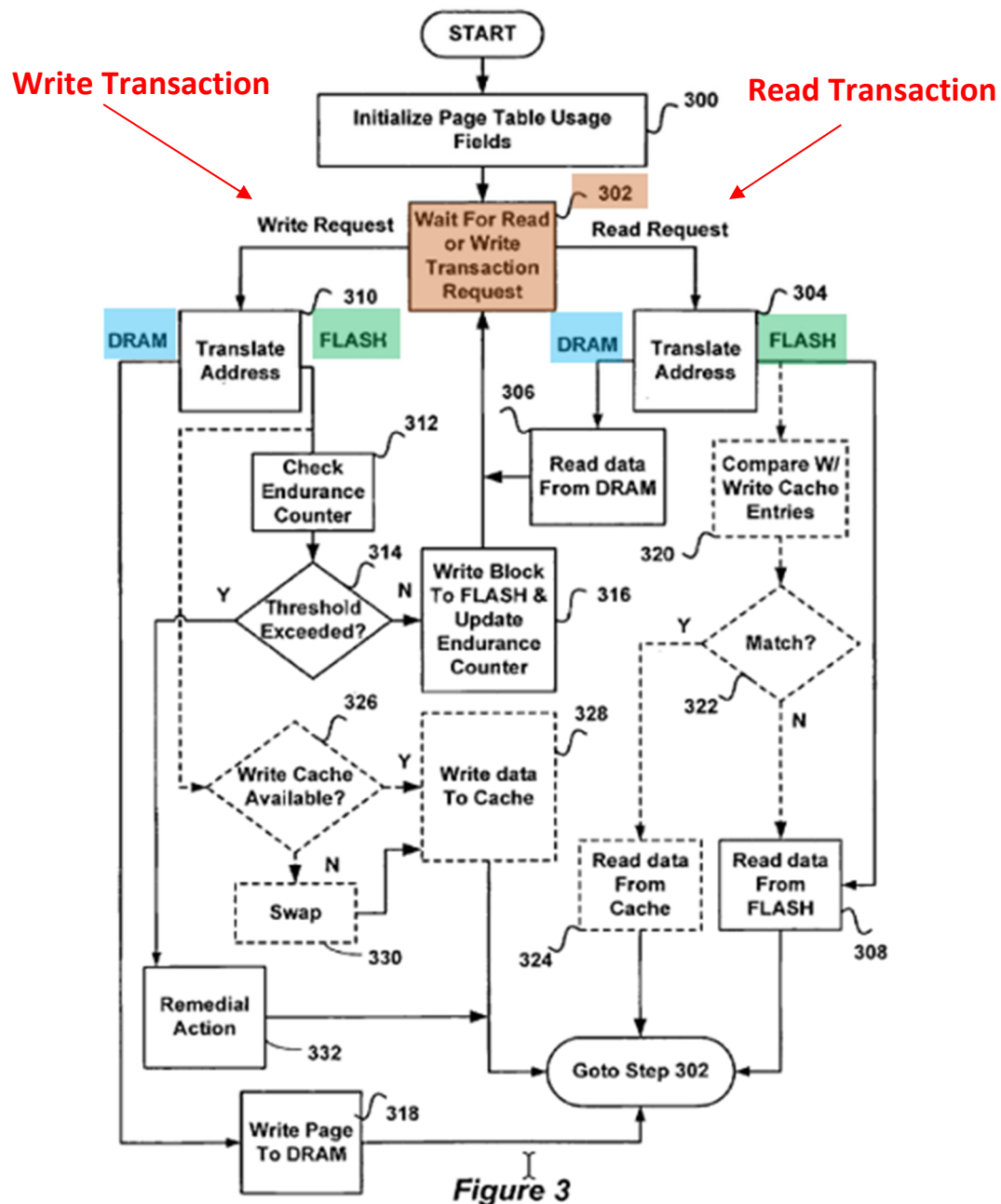
546. For example, Ware teaches that “[i]f a read request is received, the virtual address is translated into a physical address at step 304.” *Id.* at 12:14-15. Moreover, “[o]ne aspect of step 304 is reading the usage field for the page. If the

sequentially accessing bytes of data located in a row location (page) of the memory. *See supra* note 6. As a result, flash memory would have a device that is optimal for a sequential data type.

usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be read from DRAM, then the data (which may contain program code) is read at step 306 from DRAM and returned to the requestor.” *Id.* at 12:17-22. Likewise, “[i]f the transaction request is a write request, the virtual address is translated into a physical address at step 310, which once again includes reading the usage field for the page containing the specified address. If the usage field indicates that the write data should be written to DRAM (i.e., to a first portion of main memory), then the write data is written to DRAM at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45.

547. A POSITA would understand that command details would be required for executing these steps of read or write request. As such, Ware requires identifying command details for identifying a first memory location and causing a memory transaction to be performed on the first memory location.

548. Annotated Figure 3, below, further illustrates determining a read or write commence at step 302, which requires **identifying command details** for causing the memory transaction to be performed for the **first memory device** (DRAM):



549. For at least these reasons, it is my opinion that Ware teaches “said performing the memory transaction further comprising identifying command details for causing the memory transaction to be performed, wherein said command details

comprising the first memory device.”

- h. [Claim 45g]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”

550. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses that both the first memory location (first memory portion 104) and second memory location (second memory portion 106) have a set of attributes comprising a device profile. *See supra* § X.B.1.c. Ware further teaches that “[t]he memory devices in portion 104 have one or more *attributes that differ* from the attributes of the memory devices in portion 106.” Ex-1006 at 4:4-9 (emphasis added); *see also id.* at Abstract (“The first and second sets of attributes have *at least one differing attribute*”); 2:21-25 (“The memory devices . . . include two or more memory device types (e.g., DRAM, Flash) *with different attributes.*”); 4:19-29 (explaining that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns); claim 15 (reciting a first memory type and second memory type having attributes, “wherein the first and second sets of attributes have *at least one differing attribute*”) (all emphasis added). The differences in these attributes demonstrate that a difference exists in the device

profiles for the first and second memory locations.

551. For at least these reasons, it is my opinion that Ware discloses “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

- i. **[Claim 45h]: “said means for addressing obtains the first set of attributes after identifying the command details;”**

552. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches identifying command details for causing a memory transaction to be performed. *Supra* § X.B.1.h; *see also supra* § X.B.1.i. Ware further teaches that the first of attributes are obtained after the command details are identified. For example, in relation to Figure 3 reproduced below, Ware teaches that once it has been determined that a transaction request is a write request at step 302—i.e., once the command details for a write transaction have been identified—***then*** a virtual address is “translated into a physical address at step 310, which once again ***includes reading the usage field*** for the page containing the specified address.” Ex-1006 at 12:38-41; *see also id.* at 12:41-45 (explaining that the usage field then indicates whether the write data should be written to DRAM). The usage field “stores data

indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212.*” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches that, after identifying the command details for a read or write (as shown in step 302 of annotated Figure 3 below), a first set of attributes in the usage field, which describe the corresponding usage model for the first memory location in memory device 104, are obtained (as shown in step 310 of annotated Figure 3 below).

553. Annotated Figure 3, below, illustrates that—after the **command details are identified** in step 302—the address is translated in step 310, which requires **obtaining the set of attributes**:

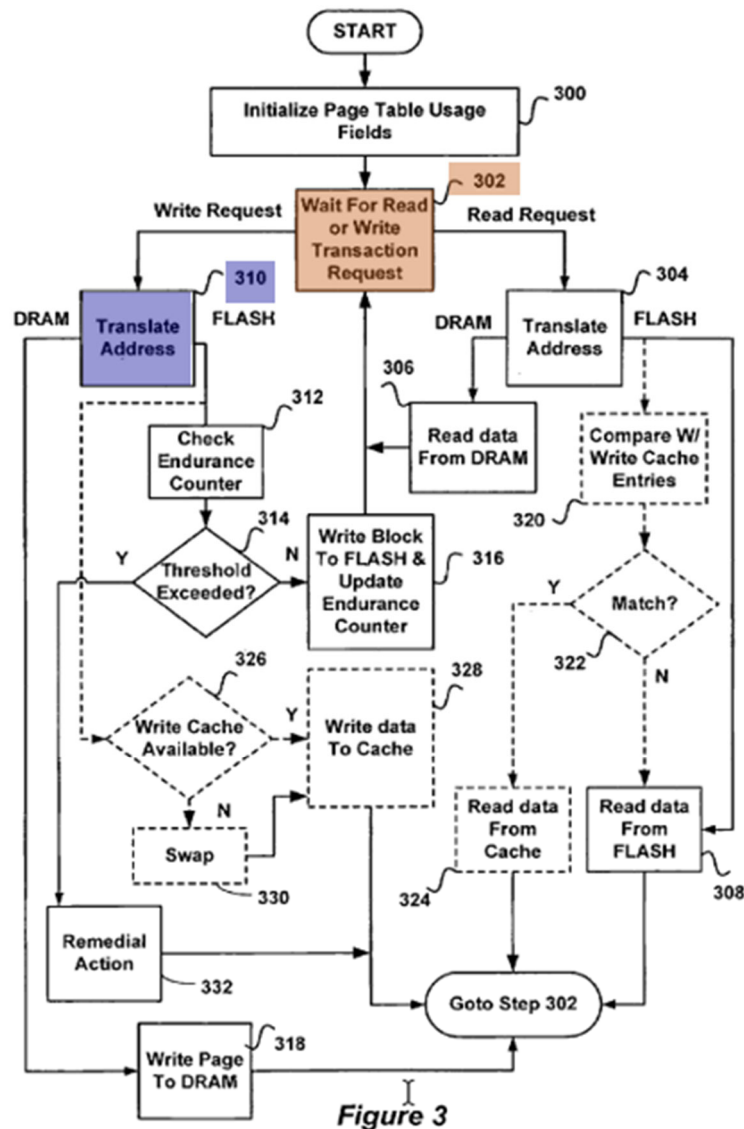


Figure 3

554. A POSITA would also understand that the means for addressing—i.e., the combination of memory interfaces 112 and 114, and processor interface logic 122—obtains the first set of attributes after identifying the command details. As illustrated above in annotated Figure 3, the host 108 sends the command details to the memory controller for identification (at step 302) via the processor interface

logic 122 (which is part of the means for addressing). In particular, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses *and control signals* to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction*, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added).

555. For at least these reasons, it is my opinion that Ware discloses “said means for addressing uses attributes from said first and second device profiles and selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

- j. **[Claim 45i]: “and said means for addressing uses attributes from said first and second device profiles and selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

556. It is my opinion that Ware in view of a POSITA’s knowledge renders obvious this claim element. Ware teaches using attributes from the first and second device profiles. As explained above, Ware teaches that the first and second device profiles (i.e., usage fields) include a variety of attributes. *See supra* § X.B.1.i. Ware also discloses using the attributes from each of these device profiles. For example,

Ware teaches that “NOR Flash memory devices typically have different granularity for read and write operations. In particular, the minimum unit for reading data from Flash memory may be a word (e.g., 16 bits) while the minimum unit for writing data to Flash memory is typically significantly larger than the minimum unit for reading, and is called either a block or a group of memory cells. In some Flash memory devices, each block includes between 1024 and 65,536 words, with blocks typically containing between 4 k (4096) and 32 k (32,768) Words.” Ex-1006 at 5:65-6:7.

557. Ware thus teaches that different memory devices have different granularity— i.e., the minimum unit for read and write operations—resulting in the need to select a transfer size for the memory transaction. For example, A POSITA would have known that flash memory can be written and read in pages. *Id.* at 6:9-12 (noting that a page “is the basic unit of data or storage in a virtual memory system”). A POSITA would have also known that a page of data is the number of bits in a row of data in a flash memory that may be transferred during a memory transaction. This page size (transfer size) is an attribute that would have been known to be associated with a device profile. A POSITA would have also known that a group of pages is a block of flash memory. *See e.g., id.* at 6:7-8 (teaching that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes.”). In this manner, Ware teaches that transfer size

is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes.

558. Similarly, Ware teaches use of attributes such as the ratio of write access time to read access time in each device profile. For example, “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1. This attribute differs from the write access time of, for example, DRAM, which is typically about the same as the read access time of DRAM devices. For example, the read access time and write access time of a DRAM device are typically both less than 100 ns, and often both are between 20 ns and 40 ns. The ratio of write access time to read access time of DRAM devices is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:19-31.

559. Ware also teaches selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes. As noted above, Ware discloses the use of a variety of different attributes, including different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. *Supra* §§ X.B.1.i, X.B.1.o. Ware further teaches that “[s]ome Flash memory

devices have blocks or groups of memory cells of two or more distinct sizes,” thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. Ex-1006 at 6:7-8.

560. Ware also discloses that “it may be noted that a page, which is the basic unit of data or storage in a virtual memory system is typically, but not necessarily, of a different size than a block in a Flash memory device. Thus, a page may be stored *in a plurality of blocks*, or *one block could store multiple pages*, or *each page could be stored in a single block*.” *Id.* at 6:9-14 (emphasis added). In view of these size differences between pages and blocks, which can result in pages being stored either within a block or across blocks, the transfer size would be a function of the data size and the first set of attributes. *See also supra* §§ X.B.1.i, X.B.1.o.

561. Ware also teaches that this functionality is performed by the means for addressing—i.e., the combination of memory interfaces 112 and 114, and processor interface logic 122. For example, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses *and control signals* to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory*

will receive a memory access transaction, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added). A POSITA would understand that, by decoding bits of a physical address to “determine whether the first or second portion of main memory will receive a memory access transaction,” the processor interface logic uses attributes from the first and second device profiles and selects a transfer size for the memory transaction as a function of the data size of the memory transaction and the first set of attributes.

562. For at least these reasons, it is my opinion that Ware in view of a POSITA’s knowledge renders obvious “said means for addressing uses attributes from said first and second device profiles and selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

8. Claim 51 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

563. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 51 of the ’190 patent.

- a. **[Claim 51]: “The memory controller of claim 45, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

564. It is my opinion that Ware renders obvious the memory controller of

claim 45 for the reasons discussed above. *Supra* § X.B.7. It is also my opinion that Ware renders obvious the memory controller recited in claim 51.

565. As explained above, Ware teaches shared attributes for the first and second device profiles (i.e., usage fields”). *Supra* § X.B.1.k. For example, Ware discloses that “[t]he memory devices in portion 104 have one or more attributes that differ from the attributes of the memory devices in portion 106. Examples of such attributes are: the ratio of write access time to read access time, volatility, and endurance.” Ex-1006 at 4:6-11; *see also id.* at 3:19-30 (describing attributes of Flash and DRAM memory). Accordingly, shared attributes are, for example, the volatilities of memory 104 (DRAM) and memory 106 (Flash). *See id.* at 3:56-61.

566. Moreover, as explained above, a POSITA would have understood that each attribute would have an attribute qualifier, which would be different for each device profile. *Supra* § X.B.1.k. For example, Ware teaches that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns.” *Id.* at 4:19-29. As result, the ratio of write access time to read access time would be different for the two devices. A POSITA would have also understood that the attribute qualifiers for the shared attribute “volatility” for DRAM and Flash are different. More specifically, the DRAM attribute qualifier for the

volatility attribute is “volatile,” while the Flash attribute qualifier for the volatility attribute is “non-volatile.” *Id.* at 3:10-12, 30-33; *see also id.* at 3:18:30 (explaining that Flash has “different attributes than DRAM,” such as being “non-volatile (i.e., Flash memory retains its contents without requiring periodic refreshing”). Likewise, a POSITA would understand the attribute “endurance” would have an attribute qualifier of “unlimited” for DRAM and “limited” for Flash. *Id.* at 3:18-30; *supra* § X.B.1.o. Accordingly, as explained above, it is my opinion that use of attribute qualifiers would have been obvious to a POSITA. *Supra* § X.B.1.k.

567. For at least these reasons, it is my opinion that Ware discloses the “memory controller of claim 45, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.

9. Claim 52 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

568. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 52 of the ’190 patent.

a. [Claim 52]: “The memory controller of claim 51, wherein said different attribute qualifiers include different attribute values.”

569. It is my opinion that Ware renders obvious the memory controller of claim 51 for the reasons discussed above. *Supra* § X.B.8.a. It is also my opinion

that Ware renders obvious the memory controller recited in claim 52.

570. As explained above, the use of different attribute qualifiers for shared attributes would have been obvious to a POSITA in view of Ware. For example, it would have been obvious to a POSITA that the “volatility” attribute would have different qualifiers such as “volatile” and “non-volatile,” which are different values *Supra* § X.B.2.a.; *see also supra* § X.B.1.k (explaining that volatility would be represented by either 1 or 0, indicating that that the memory is either volatile (“yes”) or non-volatile (“no”). Likewise, it would have been obvious that the “endurance” attribute would have different attribute values such as “unlimited” and “limited.” *Supra* § X.B.2.a.

571. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “memory controller of claim 51, wherein said different attribute qualifiers include different attribute values.”

10. Claim 59 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

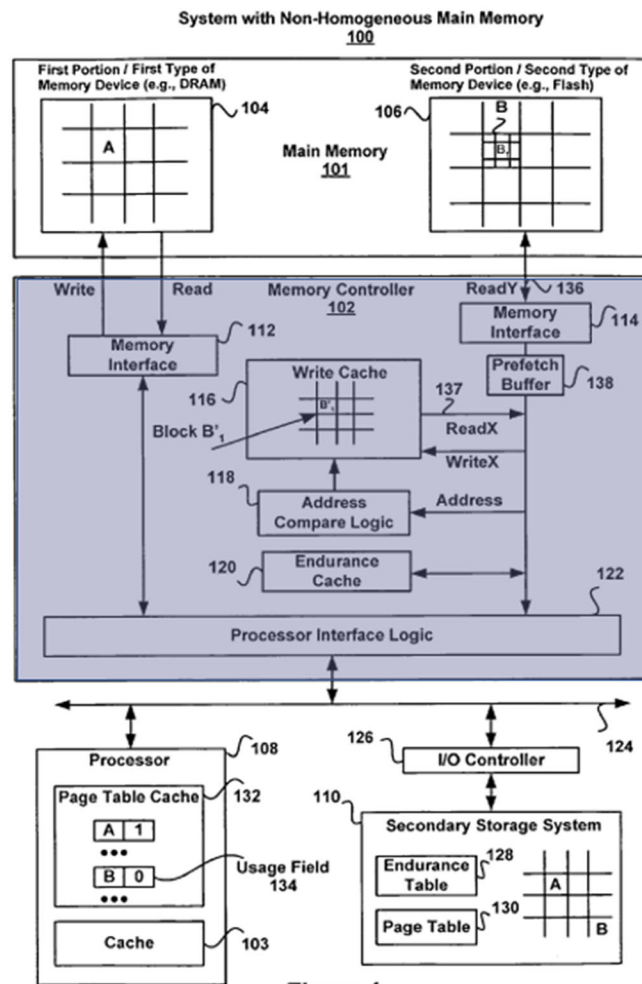
572. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 59 of the ’190 patent.

a. [Claim 59 pre]: “A memory controller comprising:”

573. To the extent the preamble of claim 59 is limiting, it is my opinion that Ware discloses this claim element. Ware is directed “to *a memory controller* for a

non-homogenous memory.” Ex-1006 at 1:7-8; *see also id.* at 2:18-21 (describing a “**memory controller** for a non-homogenous memory system [that] is configurable to facilitate page operations between a virtual memory address space and physical pages of memory devices in the memory system”); *id.* at 3:39-43 (describing a computer system that “includes a **memory controller 102**”); claim 1 (reciting a controller with “a first memory interface adapted to be coupled to one or more first memory devices” and “a second memory interface adapted to be coupled to one or more second memory devices.”) (all emphasis added).

574. Annotated Figure 1, shown below, illustrates the **memory controller** (memory controller 102):



575. For at least these reasons, it is my opinion that Ware discloses “[a] memory controller.”

- b. [Claim 59a]: “an interface controller coupled to a memory device interface and an input/output (IO) device interface;”

576. It is my opinion that Ware discloses this claim element. Ware discloses “processor interface logic 122 [that] may *include signal conditioning circuitry and other devices for transmitting* data, addresses and *control signals to and from the*

bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction*, as described below with respect to FIG. 2.” Ex-1006 at 7:46-54 (emphasis added). Ware further teaches that “[t]he memory controller 102 includes memory interfaces 112, 114 for coupling the memory controller 102 to the memory devices in two or more distinct portions (104, 106) of main memory 101, and processor interface logic 122.” *Id.* at 7:21-25.

577. A POSITA would thus understand that Ware’s “processor interface logic 122” includes both an interface controller and an input/output (IO) device interface consisting of signal conditioning circuitry and other components for transmitting signals to and from bus 124—to first or second portions of main memory—through memory device interfaces 112 and 114. This is shown in annotated Figure 1, below, which illustrates that the **interface controller** (processor interface logic 122, which would include a host interface) is coupled to the **memory device interfaces** and input/output device interface via **bus 124**:

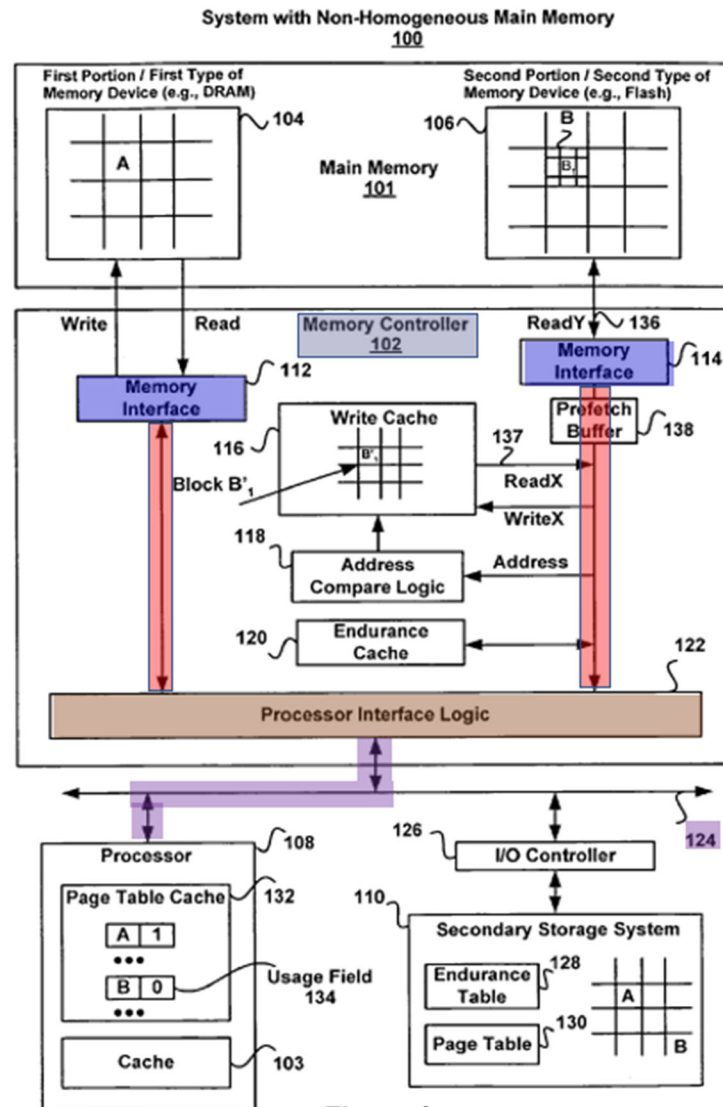


Figure 1

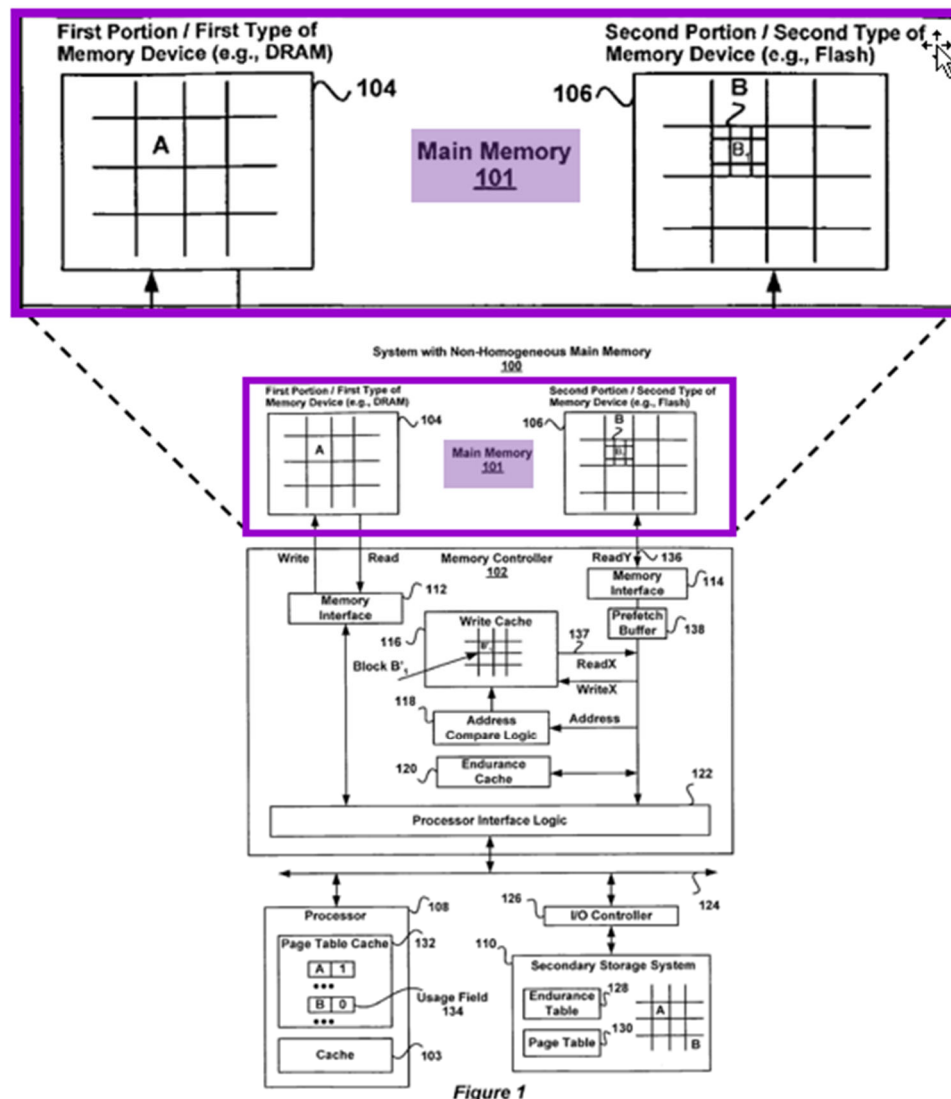
578. For at least these reasons, it is my opinion that Ware discloses “an interface controller coupled to a memory device interface and an input/output (IO) device interface.”

c. [Claim 59b]: “a memory store;”

579. It is my opinion that Ware discloses this claim element. For example,

Ware discloses “a computer system 100 having a non-homogeneous ***main memory 101***.” Ex-1006 at 3:39-40 (emphasis added). “The main memory 101 includes a first portion 104 implemented with one or more memory devices of a first type (e.g., DRAM), and a second portion 106 implemented with one or more memory devices of a second type (e.g., NOR Flash, often simply called Flash memory).” *Id.* at 3:56-61.

580. Annotated Figure 1, shown below, illustrates the memory store (main memory 101) within the system disclosed by Ware:



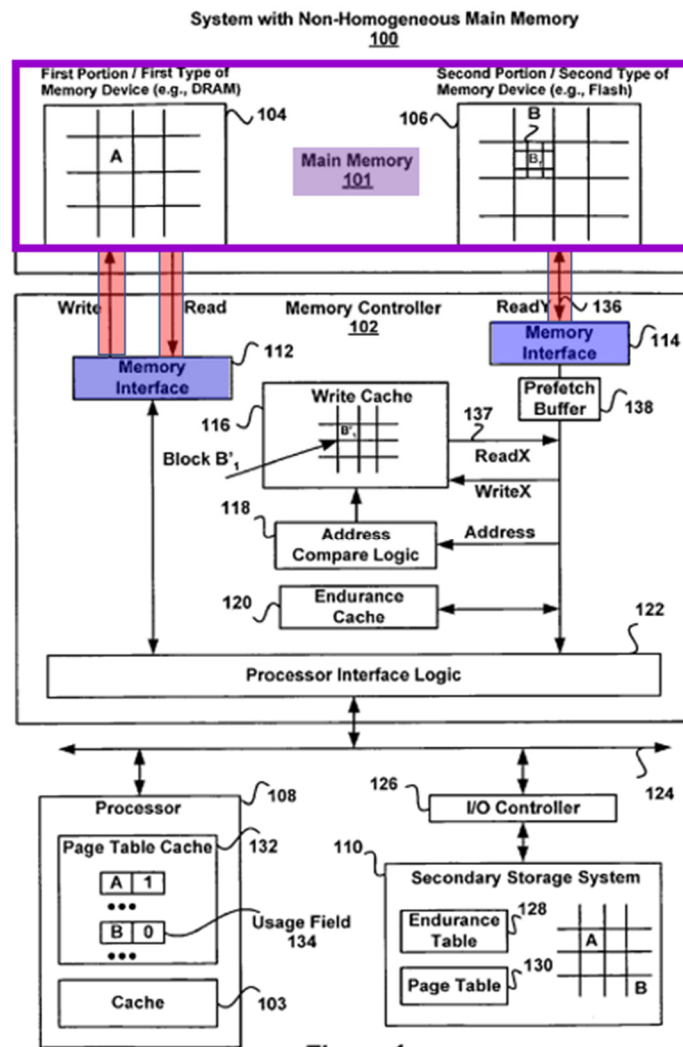
581. For at least these reasons, it is my opinion that Ware discloses “a memory store.”

- d. [Claim 59c]: wherein the memory device interface is directly coupled to the memory store;”

582. It is my opinion that Ware discloses this claim element. For example, Ware teaches that “memory controller 102 includes *memory interfaces 112, 114* for

coupling the memory controller 102 to the memory devices in two or more distinct portions (104, 106) of main memory 101, and processor interface logic 122.” Ex-1006 at 7:21-25 (emphasis added); *see also id.* at 4:4-6 (“The memory devices in the first and second portions 104, 106 of main memory are coupled to the memory controller 102 via memory interfaces 112 and 114, respectively.”).

583. Annotated Figure 1, shown below, illustrates that the **memory device interfaces** (interfaces 112 and 114) are directly coupled to the **memory store** (main memory 101) via **communication pathways**:



584. For at least these reasons, it is my opinion that Ware discloses “wherein the memory device interface is directly coupled to the memory store.”

- e. **[Claim 59d]: “said interface controller disposed to perform a memory transaction by addressing a first memory location in the memory store,”**

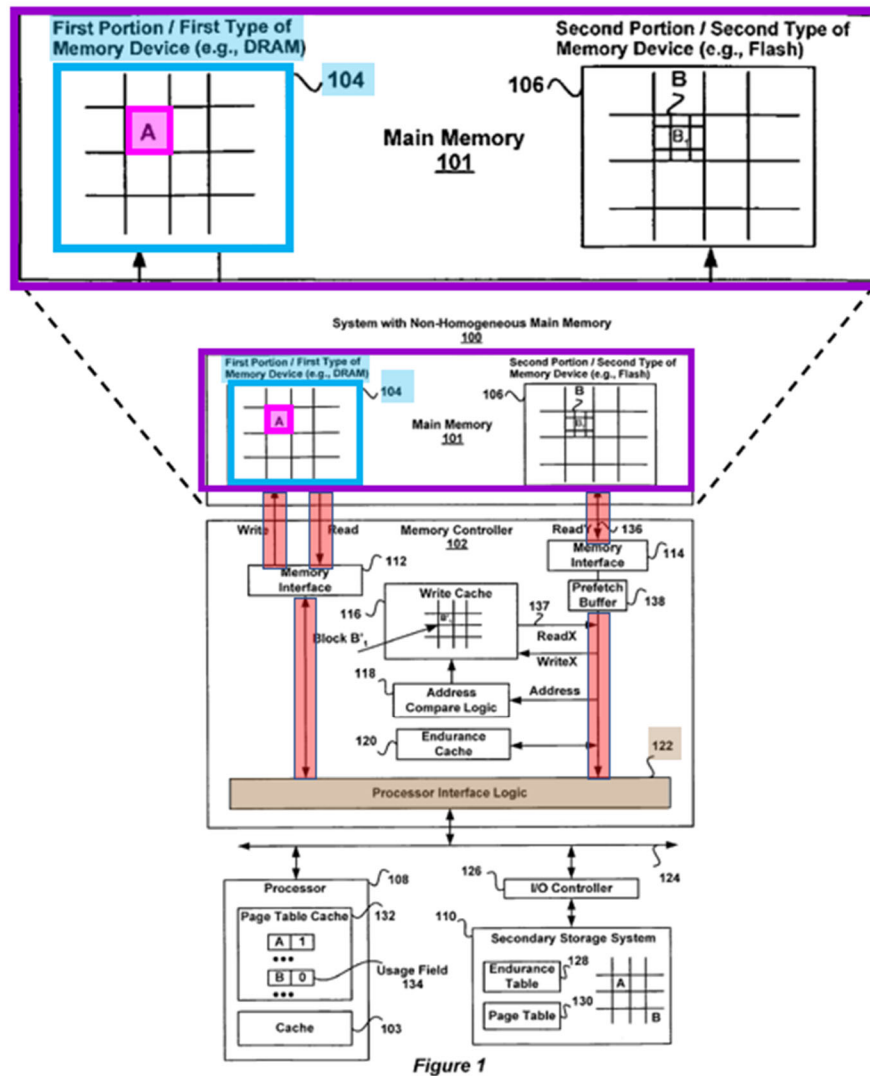
585. It is my opinion that Ware discloses this claim element. Ware discloses a memory store comprising a first memory location (first memory portion 104) and

a second memory location (second memory portion 106). Ex-1006 at 3:56-61. The first memory portion may be “implemented using one or more DRAM memory devices,” while the second memory portion may be “implemented using one or more Flash memory devices (e.g., NOR Flash memory devices).” Ex-1006 at 4:38-42; *see also id.* at 7:22-25 (noting that a memory controller includes interfaces for coupling memory “in two or more distinct portions (104, 106)”; *supra* § X.B.1.a.

586. Ware also teaches performing a memory transaction comprising addressing a first memory location in the first memory device. For example, Ware discloses that “[d]ata can be distributed among the first and second portions 104, 106 of main memory 101.” *Id.* at 3:64-67. Ware also teaches that “a usage field value of ‘A’ represents a page mapped to the first portion 104 of main memory”—i.e., the first memory location. Ex-1006 at 5:57-61. Moreover, A POSITA would readily understand that performing read and write transactions to the first memory location in the first memory device, as described by Ware, would require addressing that location. *See id.* at 7:22-25 (explaining that the memory device interfaces “include signal conditioning circuitry and other devices for transmitting addresses, data, and control signals to and from the memory devices in main memory portions 104 and 106.”); *id.* at 7:46-49 (noting that the processor interface logic may include circuitry “for transmitting data, address, and control signals.”).

587. Further, Ware teaches that the processor interface logic 122—i.e., the interface controller— “may include signal conditioning circuitry and other devices for *transmitting data, addresses and control signals* to and from the bus 124,” as well as “decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction.*” *Id.* at 7:46-54; see also *id.* at 7:21-25 (explaining that memory interfaces 112 and 114 enable “coupling the memory controller 102 to the memory devices in two or more distinct portions (104, 106) of main memory 101, and processor interface logic 122.”)

588. Annotated Figure 1, below, demonstrates that the **interface controller** (processor interface logic 122) is coupled to the **memory store** via **communication pathways**, thereby allowing it to perform a memory transaction by addressing a **first memory location** (field value A in a **first memory portion 104**):



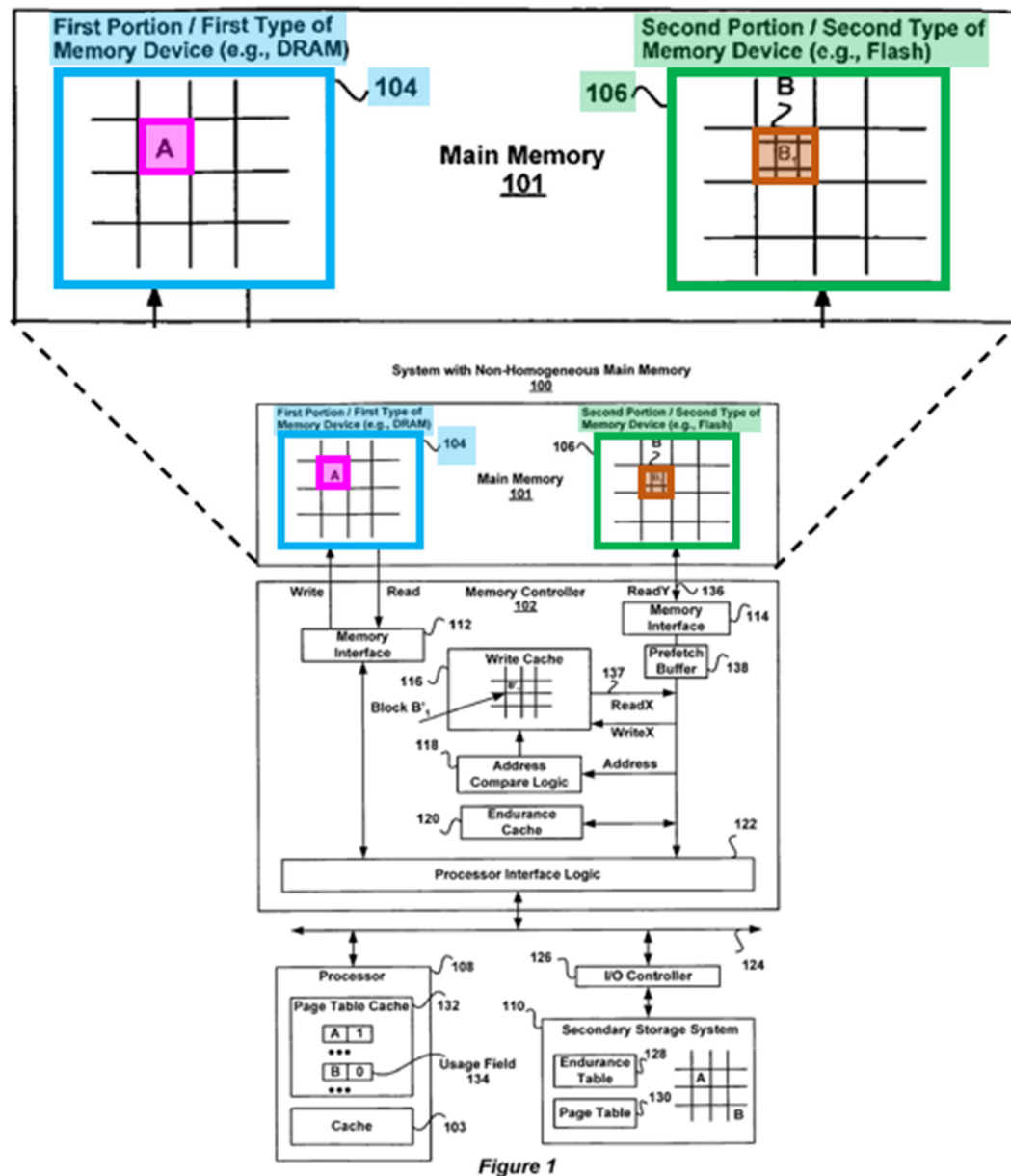
589. For at least these reasons, it is my opinion that Ware discloses “said interface controller disposed to perform a memory transaction by addressing a first memory location in the memory store.”

- f. [Claim 59e]: “said first memory location and a second memory location respectively associated with a first device profile and a second device profile;”

590. It is my opinion that Ware discloses this claim element. Ware teaches

a first memory location and a second memory location in a memory store. For example, Figure 1 of Ware discloses a “main memory 101 [that] includes *a first portion 104* implemented with one or more memory devices of a first type (*e.g., DRAM*), and a *second portion 106* implemented with one or more memory devices of a second type (*e.g., NOR Flash, often simply called Flash memory*).” *Id.* at 3:56-61 (emphasis added). Ware further teaches that “a usage field value of ‘A’ represents *a page mapped to the first portion 104 of main memory 101*, while a value of ‘B’ represents *a page mapped to the second portion 106 of main memory 101*.” *Id.* at 5:57-61 (emphasis added).

591. Annotated Figure 1, shown below, illustrates the **first memory location** (field value A in a **first memory portion 104**) and a **second memory location** (field value B in a **second memory portion 106**):



592. Ware further teaches that the first and second memory locations are associated with a device profile. For example, Ware discloses “one or more first memory devices of a first memory type having a *first set of attributes*, and a second memory interface adapted to be coupled to one or more second memory devices of

a second memory type having a *second set of attributes*.” Ex-1006, Abstract (emphasis added); claim 15 (reciting “one or more first memory devices . . . having a first set of attributes” and “one or more second memory devices . . . having a second set of attributes.”). More specifically, Ware discloses a “usage field 208” that “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212*.” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches a first device profile, or set of attributes, associated with a first memory location (the first memory portion comprising DRAM) and a second device profile, or set of attributes, associated with a second memory location (the second memory portion comprising Flash).

593. Annotated Figure 2, shown below, illustrates the **device profile** (i.e., the usage field used to store a first and second sets of attributes), which is associated with both the **first memory location** (in the first memory device) and the **second memory location** (in the second memory device):

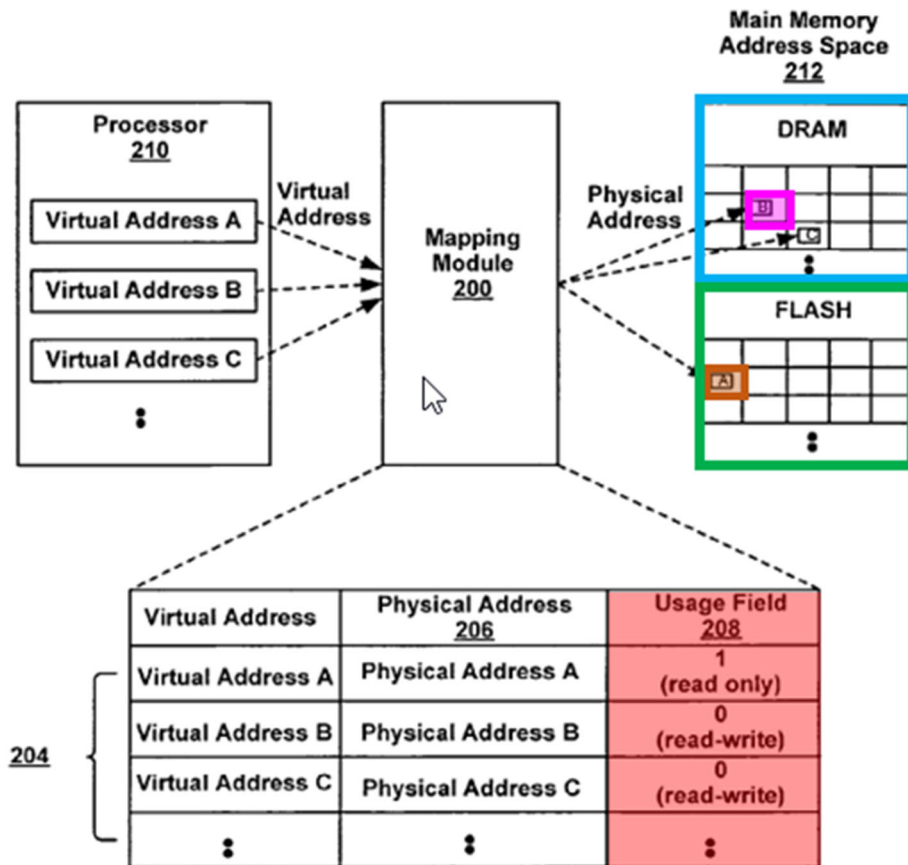


Figure 2

594. For at least these reasons, it is my opinion that Ware discloses “said first memory location and a second memory location respectively associated with a first device profile and a second device profile.”

- g. [Claim 59f]: “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”

595. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a first memory location comprising DRAM having a device profile. A POSITA would have understood that a random read command results in accessing a byte of data at some random location in the memory. Thus, a POSITA would understand that the device profile for DRAM would be optimal for a random data type. Indeed, Ware teaches that “[m]ain memory is typically implemented with Dynamic Random [data type] Access Memory (DRAM),” which “has a near-***optimal combination of operational attributes***”—comprising a device profile—“for implementing main memory.” Ex-1006, 1:65-2:6; *see also id.* at 11:1-4 (disclosing a “usage field 208” or profile that “stores data indicative of the usage model of the page, which can be used by processor 210 to ***exploit the unique attributes of a physical memory device***,” such as DRAM’s suitability for random data.”) (all emphasis added).³⁷

³⁷ Similarly, a POSITA would have understood that a second device profile for flash memory, particularly NAND flash, utilizes page read commands that result in sequentially accessing bytes of data located in a row location (page) of the memory. *See supra* note 6. As a result, flash memory would have a device that is optimal for a sequential data type.

596. For at least these reasons, it is my opinion that Ware discloses “wherein said first device profile is optimal for a data type subject to the memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

h. [Claim 59g]: “said interface controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”

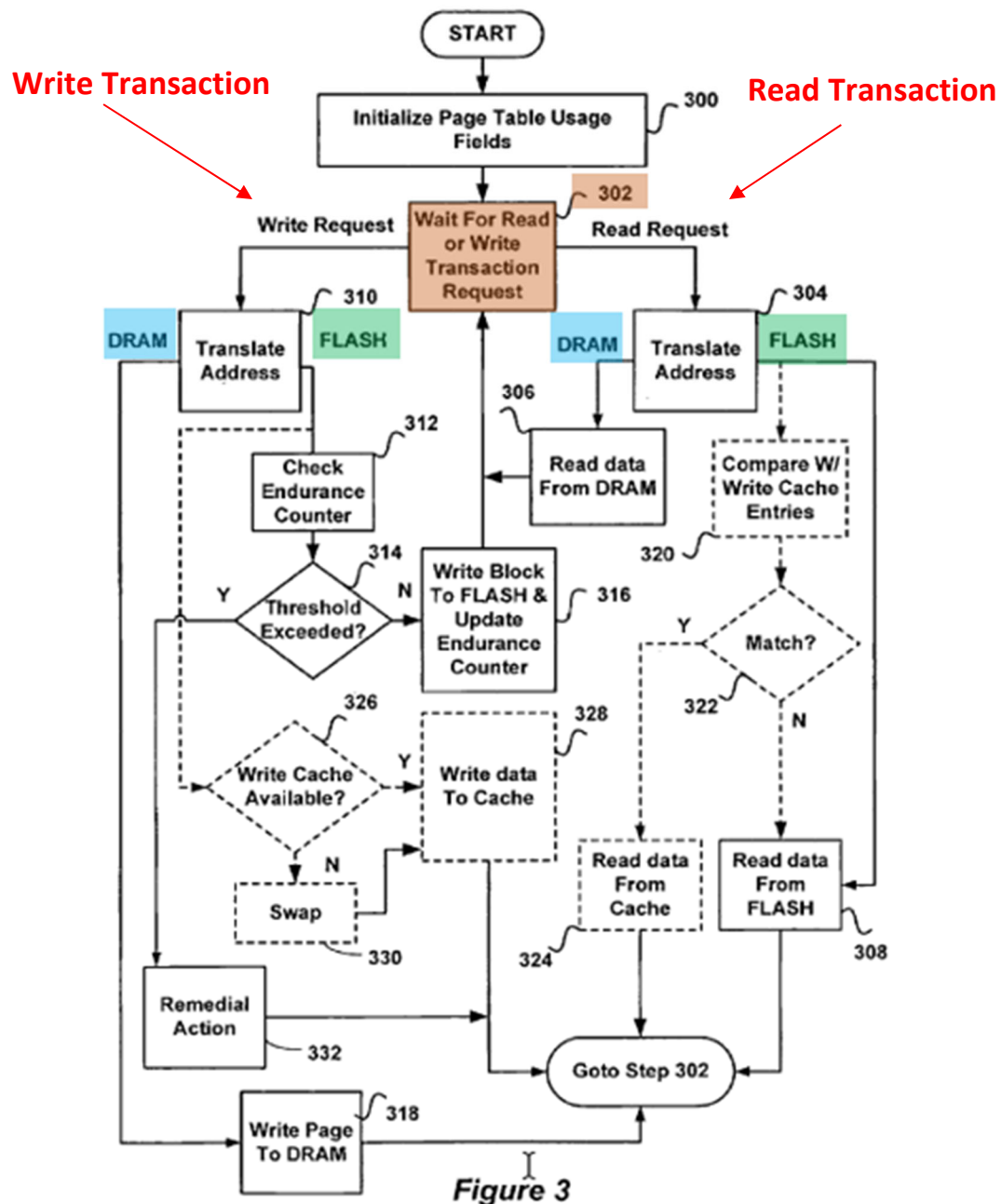
597. It is my opinion that Ware discloses this claim element. Ware teaches that, “[a]fter the initialization phase is complete, the memory controller waits [at] 302 for a read or write transaction request.” Ex-1006 at 12:12-14. A person of ordinary skill would readily understand that performing the read or write transaction request requires identifying command details for causing the memory transaction to be performed.

598. For example, Ware teaches that “[i]f a read request is received, the virtual address is translated into a physical address at step 304.” *Id.* at 12:14-15. Moreover, “[o]ne aspect of step 304 is reading the usage field for the page. If the usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be read from DRAM, then the data (which may contain program code) is read at step 306 from DRAM and returned to the requestor.” *Id.* at 12:17-22. Likewise, “[i]f the transaction request is a write request, the virtual address is translated into a physical address at step 310, which once again

includes reading the usage field for the page containing the specified address. If the usage field indicates that the write data should be written to DRAM (i.e., to a first portion of main memory), then the write data is written to DRAM at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45.

599. A POSITA would understand that command details would be required for executing these steps of read or write request. As such, Ware requires identifying command details for identifying a first memory location and causing a memory transaction to be performed on the first memory location.

600. Annotated Figure 3, below, further illustrates determining a read or write commence at step 302, which requires **identifying command details** for causing the memory transaction to be performed for the **first memory device** (DRAM):



601. Further, Ware teaches that the processor interface logic 122—i.e., the interface controller—identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

For example, Ware teaches that the processor interface logic 122 “may include signal conditioning circuitry and other devices for *transmitting data, addresses and control signals* to and from the bus 124,” as well as “decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction.*” *Id.* at 7:46-54.

602. For at least these reasons, it is my opinion that Ware teaches “said interface controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- i. **[Claim 59h]: “said device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”**

603. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses that both the first memory location (first memory portion 104) and second memory location (second memory portion 106) have a set of attributes comprising a device profile. *See supra* § X.B.1.c. Ware further teaches that “[t]he memory devices in portion 104 have one or more *attributes that differ* from the attributes of the memory devices in portion 106.” Ex-1006 at 4:4-9 (emphasis added); *see also id.* at Abstract (“The first and second sets of attributes have *at least one differing attribute*”); 2:21-25 (“The memory devices . . . include two or more

memory device types (e.g., DRAM, Flash) *with different attributes.*”); 4:19-29 (explaining that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns); claim 15 (reciting a first memory type and second memory type having attributes, “wherein the first and second sets of attributes have *at least one differing attribute*”) (all emphasis added). The differences in these attributes demonstrate that a difference exists in the device profiles for the first and second memory locations.

604. For at least these reasons, it is my opinion that Ware discloses “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

j. **[Claim 59i]: “said interface controller obtaining the first set of attributes after identifying the command details;”**

605. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches identifying command details for causing a memory transaction to be performed. *Supra* § X.B.1.h; *see also supra* § X.B.1.i. Ware further teaches that the first of attributes are obtained after the command details are identified. For

example, in relation to Figure 3 reproduced below, Ware teaches that once it has been determined that a transaction request is a write request at step 302—i.e., once the command details for a write transaction have been identified—**then** a virtual address is “translated into a physical address at step 310, which once again **includes reading the usage field** for the page containing the specified address.” Ex-1006 at 12:38-41; *see also id.* at 12:41-45 (explaining that the usage field then indicates whether the write data should be written to DRAM). The usage field “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the **unique attributes of a physical memory device associated with the main memory address space 212.**” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches that, after identifying the command details for a read or write (as shown in step 302 of annotated Figure 3 below), a first set of attributes in the usage field, which describe the corresponding usage model for the first memory location in memory device 104, are obtained (as shown in step 310 of annotated Figure 3 below).

606. Annotated Figure 3, below, illustrates that—after the **command details are identified** in step 302—the address is translated in step 310, which requires **obtaining the set of attributes:**

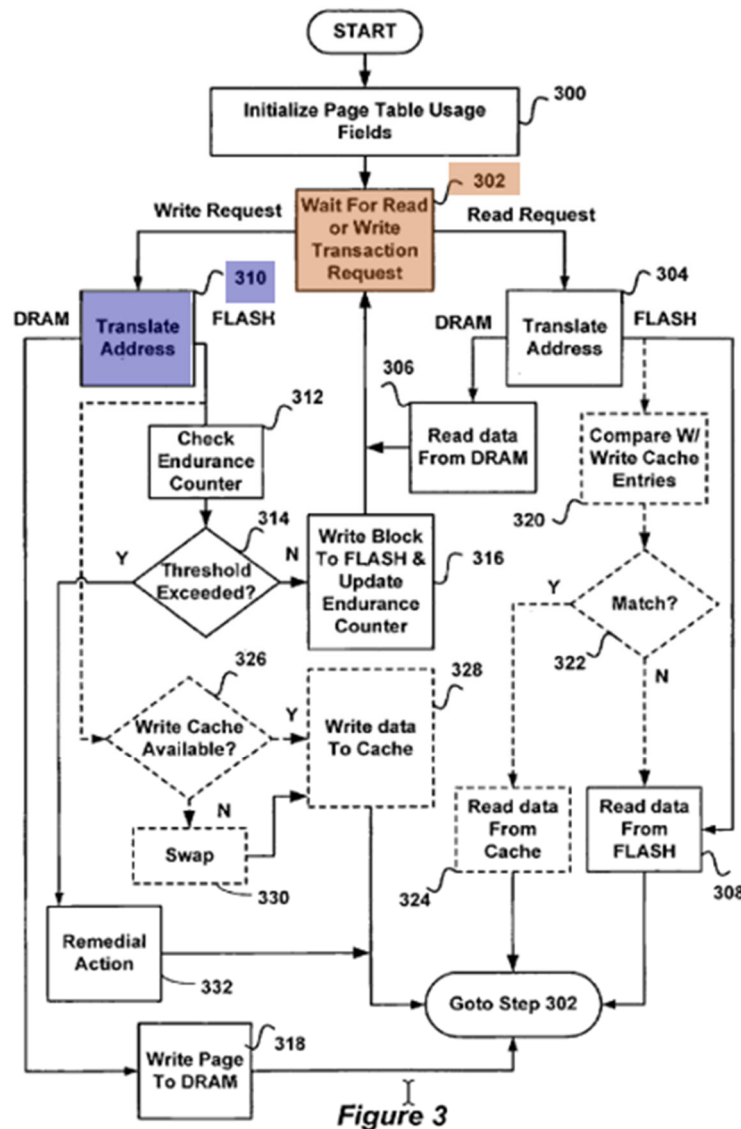


Figure 3

607. Ware also teaches that the processor interface logic 122—i.e., the interface controller—performs these functions.” Ware teaches that the processor interface logic 122 “may include signal conditioning circuitry and other devices for *transmitting data, addresses and control signals* to and from the bus 124,” as well as “decoding logic for decoding one or more bits of a physical address *to determine*

whether the first or second portion of main memory will receive a memory access transaction.” Id. at 7:46-54.

608. For at least these reasons, it is my opinion that Cornwell discloses “said interface controller obtaining the first set of attributes after identifying the command details.”

k. [Claim 59j]: “and said addressing of said first memory location includes using said attributes from said first device profile; and”

609. It is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious this claim element. As explained above, Ware teaches that the first and second device profiles (i.e., usage fields) include a variety of attributes. *See supra* § X.B.1.i. Ware also discloses using the attributes from each of these device profiles. For example, Ware teaches use of attributes such as the ratio of write access time to read access time in each device profile. As Ware explains “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1. This attribute differs from the write access time of, for example, DRAM, which is typically about the same as the read access time of DRAM devices. For example, the read access time and write access time of a DRAM device are typically both less

than 100 ns, and often both are between 20 ns and 40 ns. The ratio of write access time to read access time of DRAM devices is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:19-31.

610. Ware also discloses using the endurance attribute. As Ware explains, endurance is the “maximum number of erase-write cycles that the memory device can support.” *Id.* at 6:28-33. These erase-write cycle count “values” are “updated, maintained and checked to ensure that the endurance limit of the device is not exceeded (e.g., 10 K cycles).” *Id.* at 6:33-35. For example, “prior to performing a write operation,” the erase-write cycled count data is checked “to determine whether to complete the write operation, or perform a contingency operation (e.g., write to a different block of memory or to a different portion of main memory).” *Id.* at 6:36-41. Other measures may also be employed “to ensure that the endurance limitation of the memory devices in a portion 106 of main memory 101 are not exceeded.” *Id.* at 6:60-63.

611. Ware also teaches using a granularity attribute. For example, Ware discloses that “NOR Flash memory devices typically have different granularity for read and write operations. In particular, the minimum unit for reading data from Flash memory may be a word (e.g., 16 bits) while the minimum unit for writing data to Flash memory is typically significantly larger than the minimum unit for reading,

and is called either a block or a group of memory cells. In some Flash memory devices, each block includes between 1024 and 65,536 words, with blocks typically containing between 4 k (4096) and 32 k (32,768) Words.” Ex-1006 at 5:65-6:7.

612. Ware thus teaches that different memory devices have different granularity—i.e., the minimum unit for read and write operations—resulting in the need to select a transfer size for the memory transaction. For example, a POSITA would have known that flash memory can be written and read in pages. *Id.* at 6:9-12 (noting that a page “is the basic unit of data or storage in a virtual memory system”). A POSITA would have also known that a page of data is the number of bits in a row of data in a flash memory that may be transferred during a memory transaction. This page size (transfer size) is an attribute that would have been known to be associated with a device profile. A POSITA would have also known that a group of pages is a block of flash memory. *See e.g., id.* at 6:7-8 (teaching that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes.”). In this manner, Ware teaches that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes.

613. For at least these reasons, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious “using attributes from said first and second

device profiles.”

614. For at least these reasons, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious “said addressing of said first memory location includes using said attributes from said first device profile.”

- I. **[Claim 59k]: “said addressing of said first memory location includes selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

615. It is my opinion that Ware discloses this claim element. As explained Ware teaches the use of a variety of different attributes, including different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. *Supra* §§ X.B.1.i, X.B.1.o. Ware further teaches that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes,” thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. Ex-1006 at 6:7-8.

616. Ware also discloses that “it may be noted that a page, which is the basic unit of data or storage in a virtual memory system is typically, but not necessarily, of a different size than a block in a Flash memory device. Thus, a page may be stored *in a plurality of blocks*, or *one block could store multiple pages*, or *each page could*

be stored in a single block.” *Id.* at 6:9-14 (emphasis added). In view of these size differences between pages and blocks, which can result in pages being stored either within a block or across blocks, the transfer size would be a function of the data size and the first set of attributes. *See also supra* §§ X.B.1.i, X.B.1.o.

617. For at least these reasons, it is my opinion that Ware discloses “said addressing of said first memory location includes selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

11. Claim 71 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

618. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 71 of the ’190 patent.

- a. **[Claim 71]: “The memory controller of claim 59, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

619. It is my opinion that Ware renders obvious the memory controller of claim 59 for the reasons discussed above. *Supra* § X.B.10. It is also my opinion that Ware renders obvious the memory controller recited in claim 71.

620. As explained above, Ware teaches shared attributes for the first and second device profiles (i.e., usage fields”). *Supra* § X.B.1.k. For example, Ware

discloses that “[t]he memory devices in portion 104 have one or more attributes that differ from the attributes of the memory devices in portion 106. Examples of such attributes are: the ratio of write access time to read access time, volatility, and endurance.” Ex-1006 at 4:6-11; *see also id.* at 3:19-30 (describing attributes of Flash and DRAM memory). Accordingly, shared attributes are, for example, the volatilities of memory 104 (DRAM) and memory 106 (Flash). *See id.* at 3:56-61.

621. Moreover, as explained above, a POSITA would have understood that each attribute would have an attribute qualifier, which would be different for each device profile. *Supra* § X.B.1.k. For example, Ware teaches that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns.” *Id.* at 4:19-29. As result, the ratio of write access time to read access time would be different for the two devices. A POSITA would have also understood that the attribute qualifiers for the shared attribute “volatility” for DRAM and Flash are different. More specifically, the DRAM attribute qualifier for the volatility attribute is “volatile,” while the Flash attribute qualifier for the volatility attribute is “non-volatile.” *Id.* at 3:10-12, 30-33; *see also id.* at 3:18:30 (explaining that Flash has “different attributes than DRAM,” such as being “non-volatile (i.e., Flash memory retains its contents without requiring periodic refreshing”). Likewise,

a POSITA would understand the attribute “endurance” would have an attribute qualifier of “unlimited” for DRAM and “limited” for Flash. *Id.* at 3:18-30; *Supra* § X.B.1.o. Accordingly, as explained above, it is my opinion that use of attribute qualifiers would have been obvious to a POSITA. *Supra* § X.B.1.k.

622. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “memory controller of claim 59, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”

12. Claim 76 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

623. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 76 of the ’190 patent.

a. [Claim 76]: “The memory controller of claim 71, wherein said different attribute qualifiers include different attribute values.”

624. It is my opinion that Ware renders obvious the memory controller of claim 71 for the reasons discussed above. *Supra* § X.B.2.a X.B.11.a. It is also my opinion that Ware renders obvious the memory controller recited in claim 76.

625. As explained above, the use of different attribute qualifiers for shared attributes would have been obvious to a POSITA in view of Ware. For example, it would have been obvious to a POSITA that the “volatility” attribute would have

different qualifiers such as “volatile” and “non-volatile,” which are different values *Supra* § X.B.2.a.; *see also supra* § X.B.1.k (explaining that volatility would be represented by either 1 or 0, indicating that the memory is either volatile (“yes”) or non-volatile (“no”). Likewise, it would have been obvious that the “endurance” attribute would have different attribute values such as “unlimited” and “limited.” *Supra* § X.B.2.a.

626. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “memory controller of claim 71, wherein said different attribute qualifiers include different attribute values.”

13. Claim 80 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

627. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 80 of the ’190 patent.

a. [Claim 80 pre]: “An electronic storage device, comprising:”

628. To the extent the preamble is limiting, it is my opinion that Ware discloses this claim element. For example, Ware discloses “*a computer system 100* having a non-homogeneous main memory 101. *The computer system 100* is [sic] includes a *memory controller 102, main memory 101*, one or more processors 108 and *secondary storage 100* (e.g., a hard disk unit).” Ex-1006 at 3:38-43 (emphasis

added); claim 15 (reciting a system comprising first and second memory devices, interface logic, and a processor”). As such, Ware’s “computer system 100” is an electronic storage device comprising memory and a memory controller.

629. Figures 1, below, illustrates the electronic storage device (computer system 100):

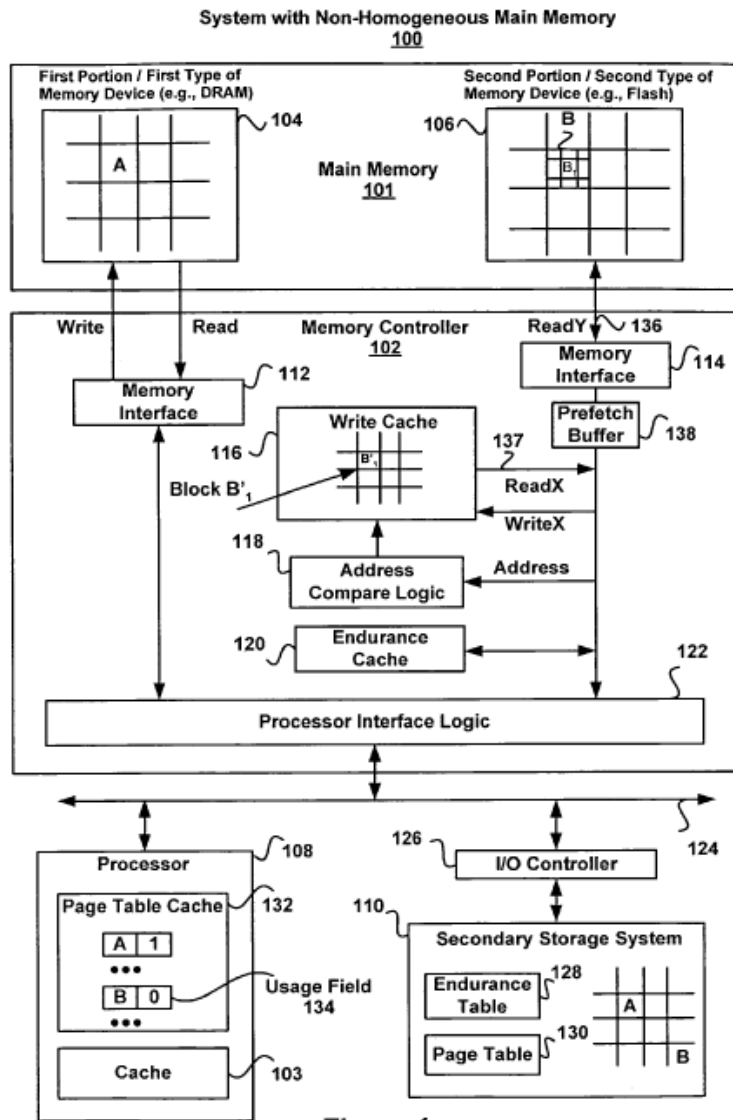


Figure 1

Electronic Storage
Device

630. Figure 4, below, further illustrates the electronic storage device (computer system 100):

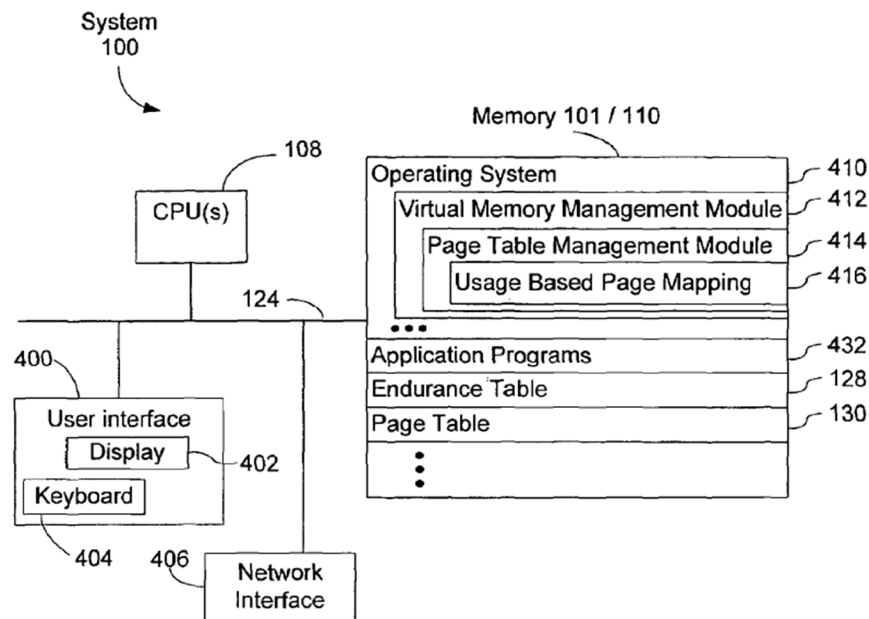


Figure 4

Electronic Storage Device

631. For at least these reasons, it is my opinion that Ware discloses “[a]n electronic storage device.”

- b. [Claim 80a]: “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile;”

632. It is my opinion that Ware discloses this claim element. As noted above, Ware discloses a memory controller. *Supra* § X.B.7.a. Ware teaches that the “main memory 101 [that] includes *a first portion 104* implemented with one or more

memory devices of a first type (*e.g., DRAM*), and a *second portion 106* implemented with one or more memory devices of a second type (*e.g., NOR Flash, often simply called Flash memory*).” Ex-1006 at 3:56-61 (emphasis added). Ware further teaches that “a usage field value of ‘A’ represents *a page mapped to the first portion 104 of main memory 101*, while a value of ‘B’ represents *a page mapped to the second portion 106 of main memory 101*.” *Id.* at 5:57-61(emphasis added). In this manner, Ware teaches first and second memory locations in the electronic device.

633. Ware also teaches that the first and second memory are directly coupled to the memory controller. For example, Ware explains that the first and second memory devices “are *coupled to the memory controller 102* via memory interfaces 112 and 114, respectively.” *Id.* at 4:4-6; *see also id.* at 7:22-25 (“The memory controller 102 includes memory interfaces 112, 114 for *coupling the memory controller 102 to the memory devices* in two or more distinct portions (104, 106) of main memory 101.”); claim 1 (reciting a controller comprising “a first memory interface adapted to be coupled to or more first memory devices” and “a second memory interface adapted to be coupled to one or more second memory devices”) (all emphasis added).

634. Annotated Figure 1, shown below, illustrates that the **memory controller** (memory controller 102) is directly coupled via **communication**

memory interface adapted to be coupled to one or more second memory devices of a second memory type having a *second set of attributes*.” Ex-1006, Abstract (emphasis added); claim 15 (reciting “one or more first memory devices . . . having a first set of attributes” and “one or more second memory devices . . . having a second set of attributes.”). More specifically, Ware discloses a “usage field 208” that “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212*.” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches a first device profile, or set of attributes, associated with a first memory location (the first memory portion comprising DRAM) and a second device profile, or set of attributes, associated with a second memory location (the second memory portion comprising Flash).

636. Annotated Figure 2, shown below, illustrates the **device profile** (i.e., the usage field used to store a first and second sets of attributes), which is associated with both the **first memory location** (in the first memory device) and the **second memory location** (in the second memory device):

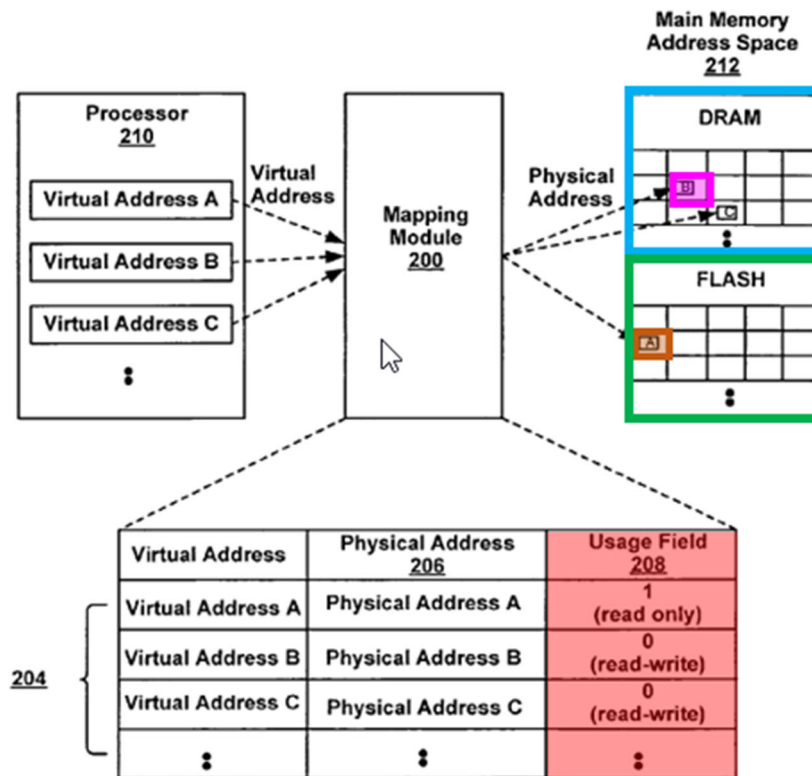


Figure 2

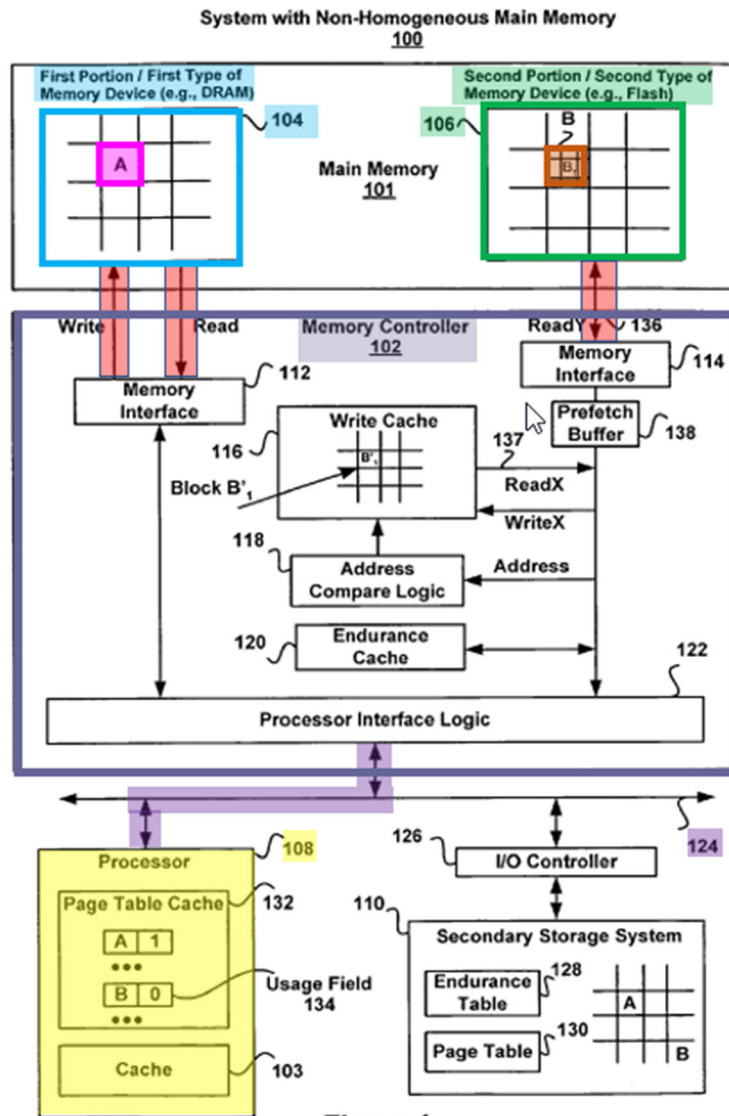
637. For at least these reasons, Ware discloses “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile.”

- c. [Claim 80b]: “said memory controller directly coupled to a host as said memory controller addresses the first memory location or second memory location;”

638. It is my opinion that Ware discloses this claim element. Ware teaches

that the host “processor 108 is coupled to the memory controller 102 via bus 124 and processor interface logic 122.” Ex-1006 at 7:9-10. Ware also teaches that the first and second memory devices “are coupled to the memory controller 102 via memory interfaces 112 and 114, respectively.” *Id.* at 4:4-6; *see also id.* at 7:22-25 (“The memory controller 102 includes memory interfaces 112, 114 for coupling the memory controller 102 to the memory devices in two or more distinct portions (104, 106) of main memory 101.”).

639. Annotated Figure 1, shown below, further illustrates that the **memory controller** (memory controller 102) is coupled to the **host processor** (processor 108) via **bus 124** and **communication paths** as it addresses the **first memory location** or **second memory location**:



640. For at least these reasons, it is my opinion that Ware discloses “said memory controller directly coupled to a host as said memory controller addresses the first memory location or second memory location.”

- d. [Claim 80c]: “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises one of a random data type or a sequential data type;”

641. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a first memory location comprising DRAM having a device profile. A POSITA would have understood that a random read command results in accessing a byte of data at some random location in the memory. Thus, a POSITA would understand that the device profile for DRAM would be optimal for a random data type. Indeed, Ware teaches that “[m]ain memory is typically implemented with Dynamic Random [data type] Access Memory (DRAM),” which “has a near-***optimal combination of operational attributes***”—comprising a device profile—“for implementing main memory.” Ex-1006, 1:65-2:6; *see also id.* at 11:1-4 (disclosing a “usage field 208” or profile that “stores data indicative of the usage model of the page, which can be used by processor 210 to ***exploit the unique attributes of a physical memory device***,” such as DRAM’s suitability for random data.”) (all emphasis added).³⁸

³⁸ Similarly, a POSITA would have understood that a second device profile for flash memory, particularly NAND flash, utilizes page read commands that result in sequentially accessing bytes of data located in a row location (page) of the memory. *See supra* note 6. As a result, flash memory would have a device that is optimal for a sequential data type.

642. For at least these reasons, it is my opinion that Ware discloses “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- e. **[Claim 80d]: “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

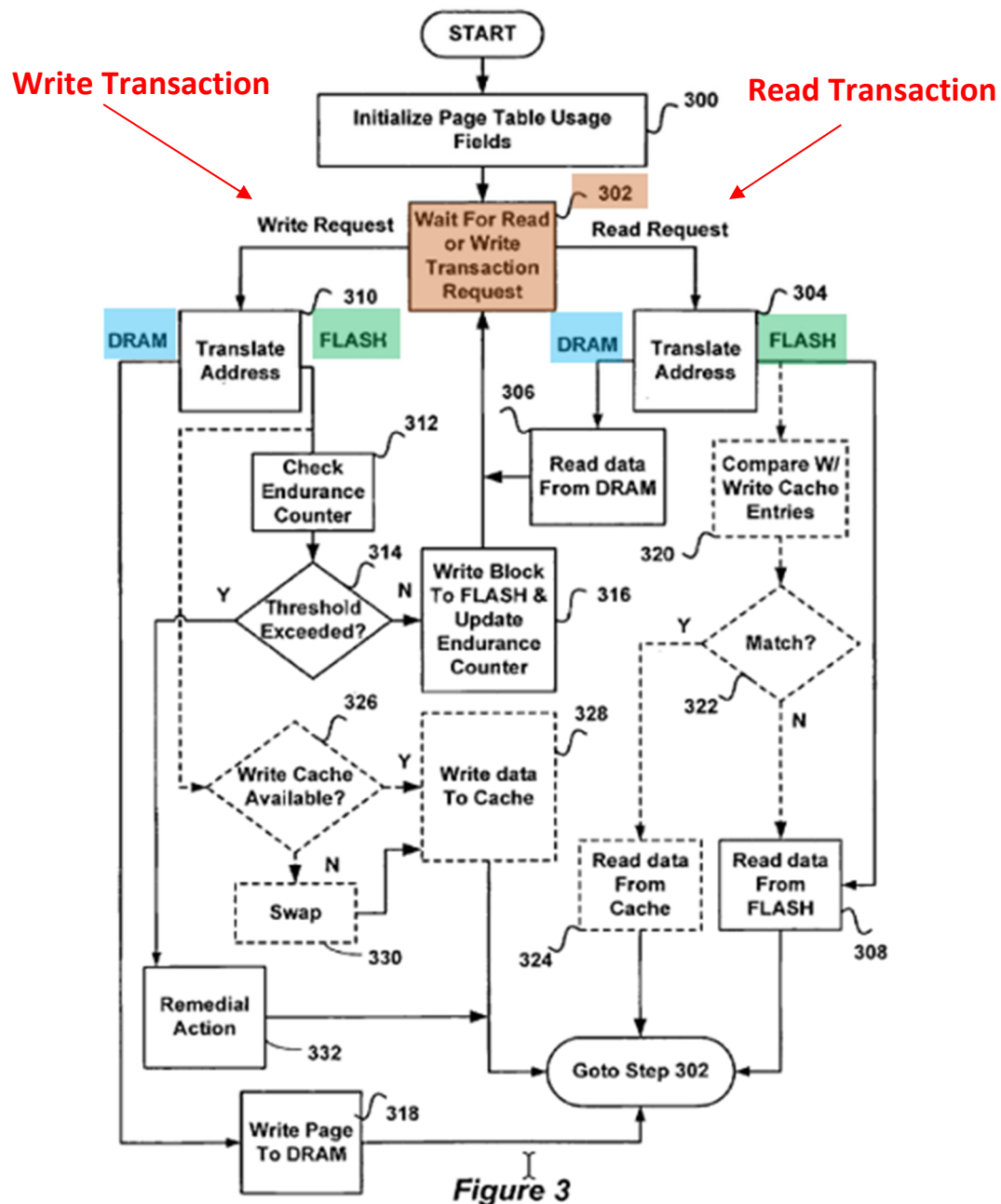
643. It is my opinion that Ware discloses this claim element. Ware teaches that, “[a]fter the initialization phase is complete, the memory controller waits [at] 302 for a read or write transaction request.” Ex-1006 at 12:12-14. POSITA would readily understand that performing the read or write transaction request requires the memory controller to identify command details for causing the memory transaction to be performed.

644. For example, Ware teaches that “[i]f a read request is received, the virtual address is translated into a physical address at step 304.” *Id.* at 12:14-15. Moreover, “[o]ne aspect of step 304 is reading the usage field for the page. If the usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be read from DRAM, then the data (which may contain program code) is read at step 306 from DRAM and returned to the requestor.” *Id.* at 12:17-22. Likewise, “[i]f the transaction request is a write request, the virtual address is translated into a physical address at step 310, which once again

includes reading the usage field for the page containing the specified address. If the usage field indicates that the write data should be written to DRAM (i.e., to a first portion of main memory), then the write data is written to DRAM at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45.

645. A POSITA would understand that command details would be required for executing these steps of read or write request. As such, Ware requires identifying command details for identifying a first memory location and causing a memory transaction to be performed on the first memory location.

646. Annotated Figure 3, below, further illustrates determining a read or write commence at step 302, which requires **identifying command details** for causing the memory transaction to be performed for the **first memory device** (DRAM):



647. For at least these reasons, it is my opinion that Ware teaches “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- f. [Claim 80e]: “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles;”

648. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses that both the first memory location (first memory portion 104) and second memory location (second memory portion 106) have a set of attributes comprising a device profile. *See supra* § X.B.1.c. Ware further teaches that “[t]he memory devices in portion 104 have one or more *attributes that differ* from the attributes of the memory devices in portion 106.” Ex-1006 at 4:4-9 (emphasis added); *see also id.* at Abstract (“The first and second sets of attributes have *at least one differing attribute*”); 2:21-25 (“The memory devices . . . include two or more memory device types (e.g., DRAM, Flash) *with different attributes.*”); 4:19-29 (explaining that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns); claim 15 (reciting a first memory type and second memory type having attributes, “wherein the first and second sets of attributes have *at least one differing attribute*”) (all emphasis added). The differences in these attributes demonstrate that a difference exists in the device profiles for the first and second memory locations.

649. For at least these reasons, it is my opinion that Ware discloses “said first device profile representing a first set of attributes of said first memory location, and said second device profile representing a second set of attributes of said second memory location, and a difference exists between said first and second device profiles.”

g. **[Claim 80f]: “said memory controller obtaining the first set of attributes after identifying the command details;”**

650. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches identifying command details for causing a memory transaction to be performed. *Supra* § X.B.1.h; *see also supra* § X.B.1.i. Ware further teaches that the first of attributes are obtained after the command details are identified. For example, in relation to Figure 3 reproduced below, Ware teaches that once it has been determined that a transaction request is a write request at step 302—i.e., once the command details for a write transaction have been identified—***then*** a virtual address is “translated into a physical address at step 310, which once again ***includes reading the usage field*** for the page containing the specified address.” Ex-1006 at 12:38-41; *see also id.* at 12:41-45 (explaining that the usage field then indicates whether the write data should be written to DRAM). The usage field “stores data indicative of the usage model of the page, which can be used by processor 210 to

exploit the *unique attributes of a physical memory device associated with the main memory address space 212.*” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches that, after identifying the command details for a read or write (as shown in step 302 of annotated Figure 3 below), a first set of attributes in the usage field, which describe the corresponding usage model for the first memory location in memory device 104, are obtained (as shown in step 310 of annotated Figure 3 below).

651. Annotated Figure 3, below, illustrates that—after the **command details are identified** in step 302—the address is translated in step 310, which requires **obtaining the set of attributes**:

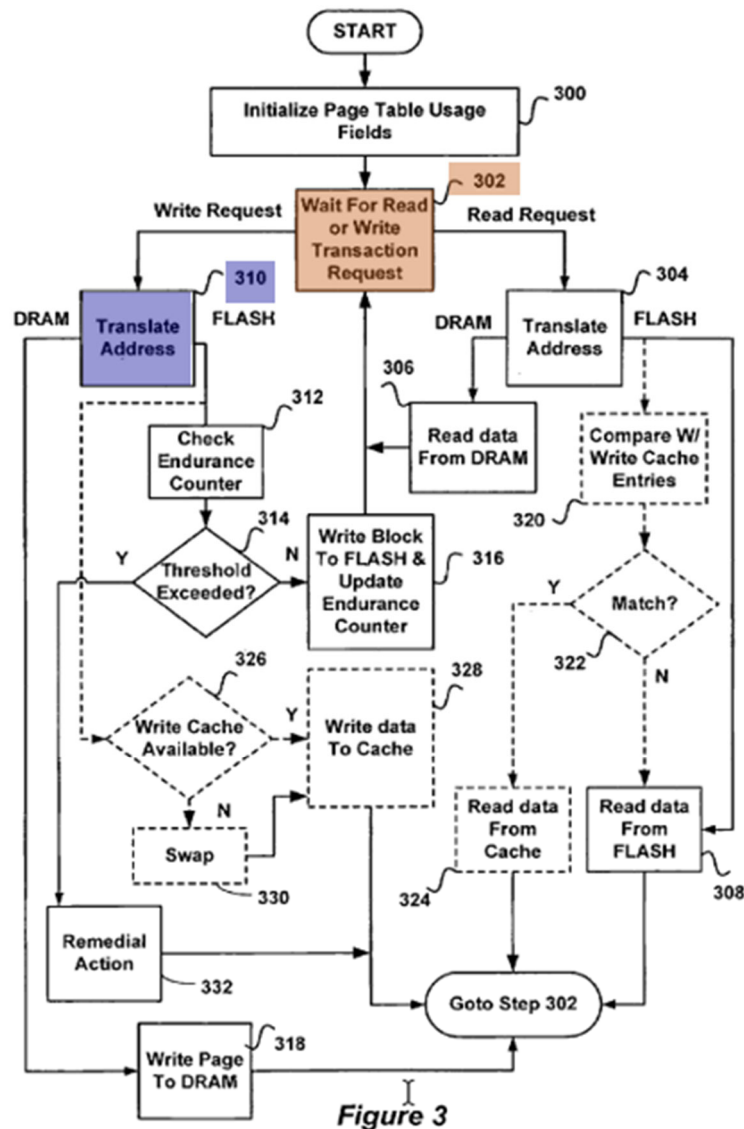


Figure 3

652. A POSITA would also understand that the memory controller obtains the first set of attributes after identifying the command details. As illustrated above in annotated Figure 3, the host 108 sends the command details to the memory controller for identification (at step 302) via the processor interface logic 122 (which is part of the means for addressing). In particular, Ware teaches that “[t]he processor

interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses *and control signals* to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction*, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added).

653. For at least these reasons, it is my opinion that Ware discloses “said memory controller obtaining the first set of attributes after identifying the command details.”

- h. [Claim 80g]: “and said memory controller using attributes from said first and second device profiles; and”**

654. It is my opinion that Ware, in view of a POSITA’s knowledge, renders this claim element obvious. As explained above, Ware teaches that the first and second device profiles (i.e., usage fields) include a variety of attributes. *See supra* § X.B.1.i. Ware also discloses using the attributes from each of these device profiles. For example, Ware teaches that “NOR Flash memory devices typically have different granularity for read and write operations. In particular, the minimum unit for reading data from Flash memory may be a word (e.g., 16 bits) while the minimum unit for writing data to Flash memory is typically significantly larger than the

minimum unit for reading, and is called either a block or a group of memory cells. In some Flash memory devices, each block includes between 1024 and 65,536 words, with blocks typically containing between 4 k (4096) and 32 k (32,768) Words.” Ex-1006 at 5:65-6:7.

655. Ware thus teaches that different memory devices have different granularity—i.e., the minimum unit for read and write operations—resulting in the need to select a transfer size (granularity) for the memory transaction. For example, a POSITA would have known that flash memory can be written and read in pages. *Id.* at 6:9-12 (noting that a page “is the basic unit of data or storage in a virtual memory system”). A POSITA would have also known that a page of data is the number of bits in a row of data in a flash memory that may be transferred during a memory transaction. This page size (transfer size) is an attribute that would have been known to be associated with a device profile. A POSITA would have also known that a group of pages is a block of flash memory. *See e.g., id.* at 6:7-8 (teaching that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes.”). In this manner, Ware teaches that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes.

656. Similarly, Ware teaches use of attributes such as the ratio of write

access time to read access time in each device profile. For example, “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1. This attribute differs from the write access time of, for example, DRAM, which is typically about the same as the read access time of DRAM devices. For example, the read access time and write access time of a DRAM device are typically both less than 100 ns, and often both are between 20 ns and 40 ns. The ratio of write access time to read access time of DRAM devices is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:19-31.

657. Ware in view of a POSITA’s knowledge also renders the use of these attributes by a memory controller obvious. For example, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses *and control signals* to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction*, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added). A POSITA would understand that, by decoding bits of a physical address to “determine

whether the first or second portion of main memory will receive a memory access transaction,” the processor interface logic in the memory controller uses attributes from the first and second device profiles.

658. For at least these reasons, it is my opinion that Ware, in view of a POSITA’s knowledge renders obvious “said memory controller using attributes from said first and second device profiles.”

- i. **[Claim 80h]: “said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

659. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches the use of a variety of different attributes, including different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. *Supra* §§ X.B.1.i, X.B.1.o. Ware further teaches that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes,” thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. Ex-1006 at 6:7-8.

660. Ware also discloses that “it may be noted that a page, which is the basic unit of data or storage in a virtual memory system is typically, but not necessarily, of a different size than a block in a Flash memory device. Thus, a page may be stored

in a plurality of blocks, or one block could store multiple pages, or each page could be stored in a single block.” *Id.* at 6:9-14 (emphasis added). In view of these size differences between pages and blocks, which can result in pages being stored either within a block or across blocks, the transfer size would be a function of the data size and the first set of attributes. *See also supra* §§ X.B.1.i, X.B.1.o.

661. Ware in view of a POSITA’s knowledge also renders using the memory controller selecting a transfer size for the memory transaction obvious. For example, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses **and control signals** to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address **to determine whether the first or second portion of main memory will receive a memory access transaction**, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added). A POSITA would understand that, by decoding bits of a physical address to “determine whether the first or second portion of main memory will receive a memory access transaction,” the memory controller selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.

662. For at least these reasons, it is my opinion that Ware discloses “said

memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

14. Claim 84 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

663. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 84 of the ’190 patent.

- a. **[Claim 84]: “The electronic storage device of claim 80, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles.”**

664. It is my opinion that Ware renders obvious the electronic storage device of claim 80 for the reasons discussed above. *Supra* § X.B.13. It is also my opinion that Ware renders obvious the electronic storage device recited in claim 84.

665. As explained above, Ware teaches shared attributes for the first and second device profiles (i.e., usage fields”). *Supra* § X.B.1.k. For example, Ware discloses that “[t]he memory devices in portion 104 have one or more attributes that differ from the attributes of the memory devices in portion 106. Examples of such attributes are: the ratio of write access time to read access time, volatility, and endurance.” Ex-1006 at 4:6-11; *see also id.* at 3:19-30 (describing attributes of Flash and DRAM memory). Accordingly, shared attributes are, for example, the

volatilities of memory 104 (DRAM) and memory 106 (Flash). *See id.* at 3:56-61.

666. Moreover, as explained above, a POSITA would have understood that each attribute would have an attribute qualifier, which would be different for each device profile. *Supra* § X.B.1.k. For example, Ware teaches that NOR Flash devices may have a read access time of less than 100 ns and an erase-write time of more than 1 millisecond, which “*differs from*” DRAM in which the read and write access times are less than 100 ns.” *Id.* at 4:19-29. As result, the ratio of write access time to read access time would be different for the two devices. A POSITA would have also understood that the attribute qualifiers for the shared attribute “volatility” for DRAM and Flash are different. More specifically, the DRAM attribute qualifier for the volatility attribute is “volatile,” while the Flash attribute qualifier for the volatility attribute is “non-volatile.” *Id.* at 3:10-12, 30-33; *see also id.* at 3:18:30 (explaining that Flash has “different attributes than DRAM,” such as being “non-volatile (i.e., Flash memory retains its contents without requiring periodic refreshing”). Likewise, a POSITA would understand the attribute “endurance” would have an attribute qualifier of “unlimited” for DRAM and “limited” for Flash. *Id.* at 3:18-30; *supra* § X.B.1.o. Accordingly, as explained above, it is my opinion that use of attribute qualifiers would have been obvious to a POSITA. *Supra* § X.B.1.k.

667. For at least these reasons, it is my opinion that Ware in view of a

POSITA's renders obvious the "electronic storage device of claim 80, wherein said difference includes different attribute qualifiers for at least one set of shared attributes from said first and second device profiles."

15. Claim 85 is Rendered Obvious by Ware in View of a POSITA's Knowledge

668. As explained below, it is my opinion that Ware, in view of a POSITA's knowledge, renders obvious Claim 85 of the '190 patent.

a. [Claim 85]: "The electronic storage device of claim 84, wherein said different attribute qualifiers include different attribute values."

669. It is my opinion that Ware renders obvious the electronic storage device of claim 84 for the reasons discussed above. *Supra* § X.B.14.a. It is also my opinion that Ware renders obvious the electronic storage device recited in claim 85.

670. As explained above, the use of different attribute qualifiers for shared attributes would have been obvious to a POSITA in view of Ware. For example, it would have been obvious to a POSITA that the "volatility" attribute would have different qualifiers such as "volatile" and "non-volatile," which are different values *Supra* § X.B.2.a.; *see also supra* § X.B.1.k (explaining that volatility would be represented by either 1 or 0, indicating that the memory is either volatile ("yes") or non-volatile ("no")). Likewise, it would have been obvious that the "endurance" attribute would have different attribute values such as "unlimited" and "limited."

Supra § X.B.2.a.

671. For at least these reasons, it is my opinion that Ware, in view of the knowledge of a POSITA, renders obvious the “electronic storage device of claim 84, wherein said different attribute qualifiers include different attribute values.”

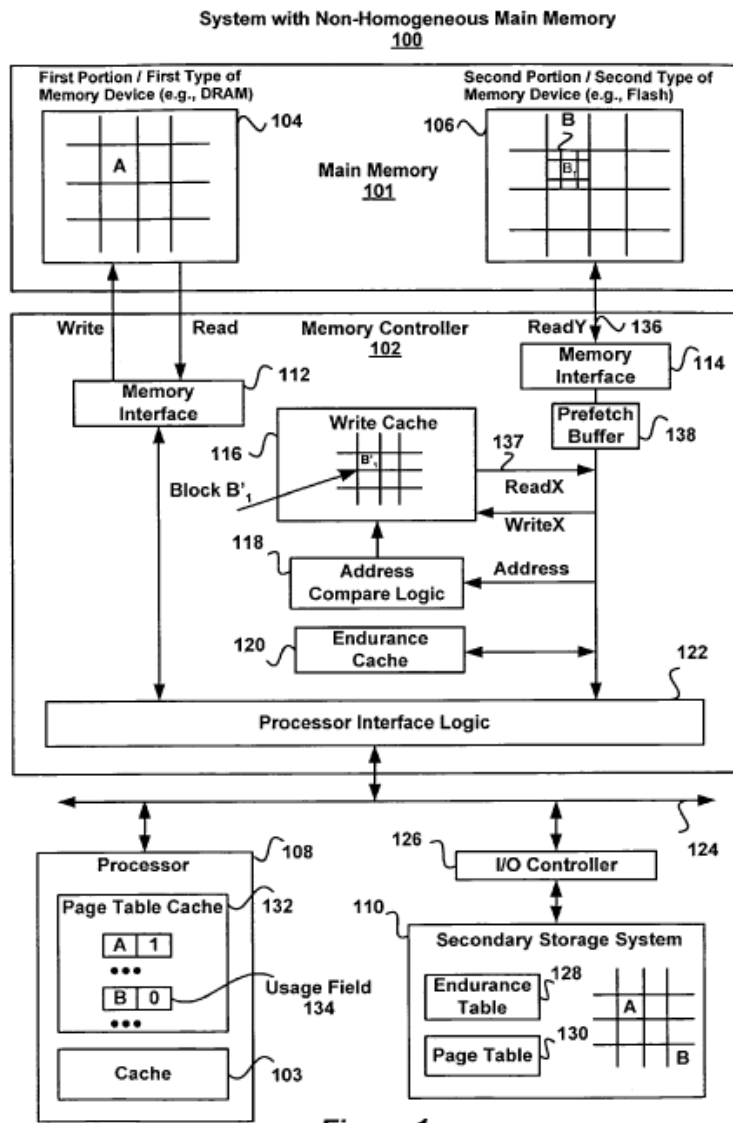
16. Claim 98 is Rendered Obvious by Ware in View of a POSITA’s Knowledge

672. As explained below, it is my opinion that Ware, in view of a POSITA’s knowledge, renders obvious Claim 98 of the ’190 patent.

a. [Claim 98 pre]: “An electronic storage device, comprising:”

673. To the extent the preamble is limiting, it is my opinion that Ware discloses this claim element. For example, Ware discloses “*a computer system 100* having a non-homogeneous main memory 101. *The computer system 100* is [sic] includes a *memory controller 102*, *main memory 101*, one or more processors 108 and *secondary storage 100* (e.g., a hard disk unit).” Ex-1006 at 3:38-43 (emphasis added); claim 15 (reciting a system comprising first and second memory devices, interface logic, and a processor”). As such, Ware’s “computer system 100” is an electronic storage device comprising memory and a memory controller.

674. Figures 1, below, illustrates the electronic storage device (computer system 100):



Electronic Storage
Device

675. Figure 4, below, further illustrates the electronic storage device (computer system 100):

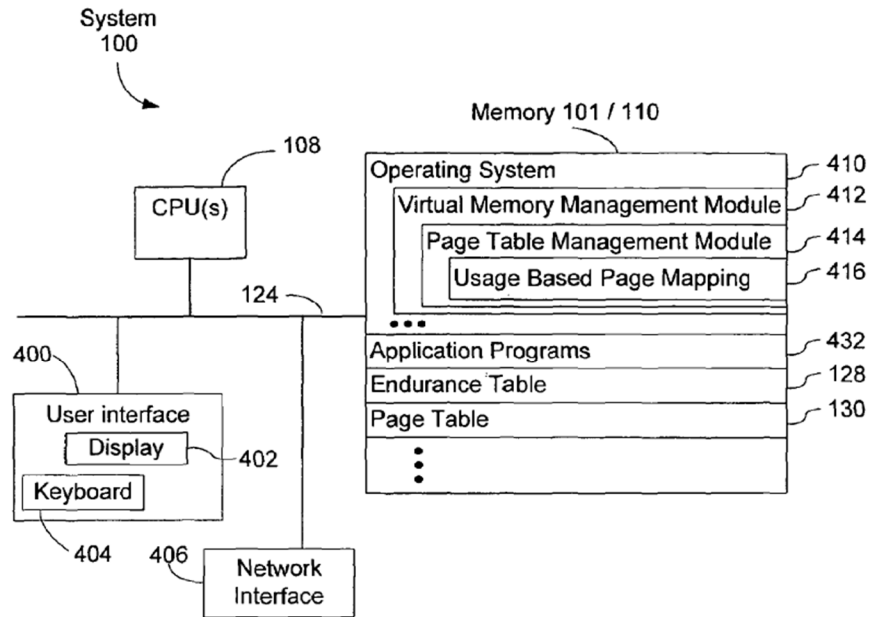


Figure 4

Electronic Storage Device

676. For at least these reasons, it is my opinion that Ware discloses “[a]n electronic storage device.

- b. **[Claim 98a]: “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile;”**

677. See above analysis for Claim 80a.

678. It is my opinion that Ware discloses this claim element. Ware teaches

that the “main memory 101 [that] includes ***a first portion 104*** implemented with one or more memory devices of a first type (*e.g., DRAM*), and a ***second portion 106*** implemented with one or more memory devices of a second type (*e.g., NOR Flash, often simply called Flash memory*).” Ex-1006 at 3:56-61 (emphasis added). Ware further teaches that “a usage field value of ‘A’ represents ***a page mapped to the first portion 104 of main memory 101***, while a value of ‘B’ represents ***a page mapped to the second portion 106*** of main memory 101.” *Id.* at 5:57-61(emphasis added). In this manner, Ware teaches first and second memory locations in the electronic device.

679. Ware also teaches that the first and second memory are directly coupled to the memory controller. For example, Ware explains that the first and second memory devices “are ***coupled to the memory controller 102*** via memory interfaces 112 and 114, respectively.” *Id.* at 4:4-6; *see also id.* at 7:22-25 (“The memory controller 102 includes memory interfaces 112, 114 for ***coupling the memory controller 102 to the memory devices*** in two or more distinct portions (104, 106) of main memory 101.”); claim 1 (reciting a controller comprising “a first memory interface adapted to be coupled to or more first memory devices” and “a second memory interface adapted to be coupled to one or more second memory devices”) (all emphasis added).

680. Annotated Figure 1, shown below, illustrates that the **memory controller** (memory controller 102) is directly coupled via **communication pathways** to a **first memory location** (field value A in a **first memory portion 104**) and a **second memory location** (field value B in a **second memory portion 106**):

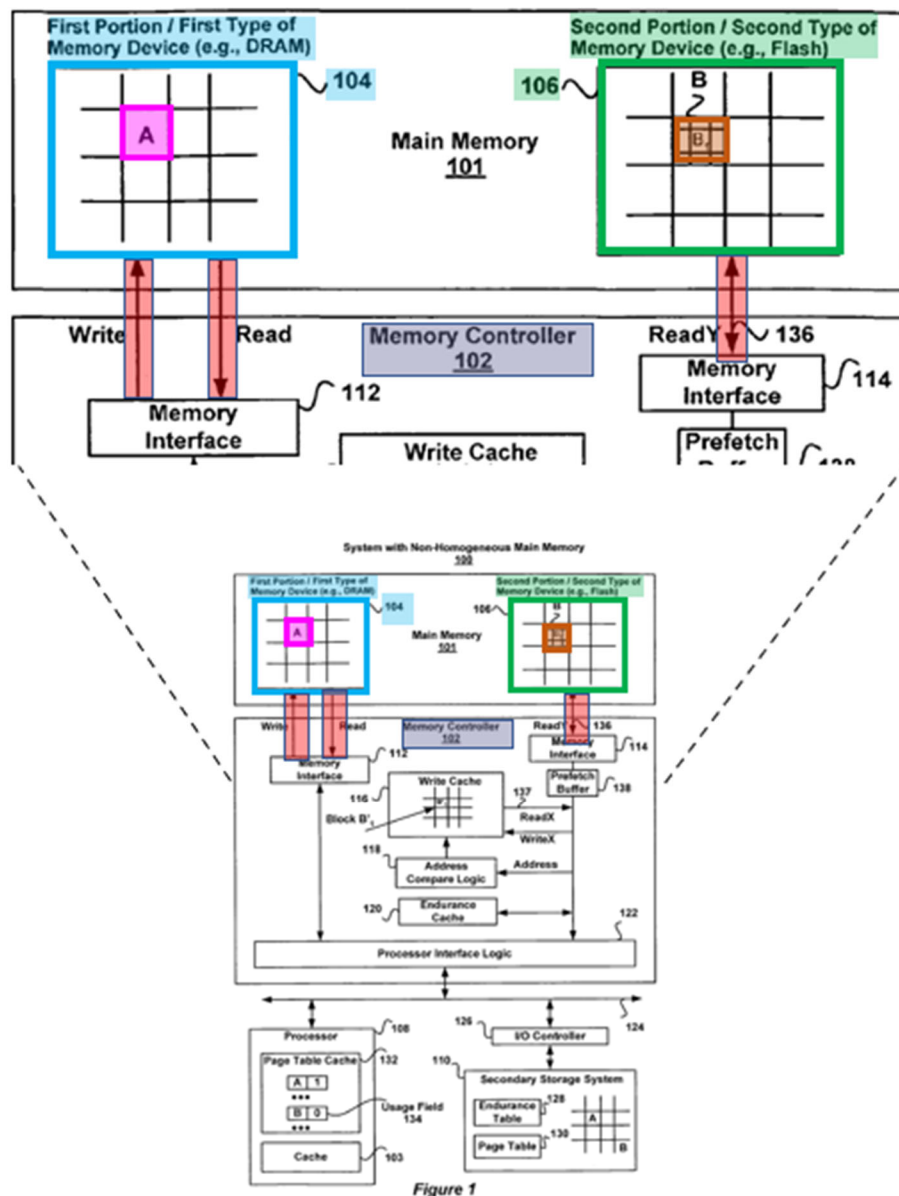


Figure 1

681. Ware further teaches that the first and second memory locations are associated with a device profile. For example, Ware discloses “one or more first memory devices of a first memory type having a *first set of attributes*, and a second memory interface adapted to be coupled to one or more second memory devices of a second memory type having a *second set of attributes*.” Ex-1006, Abstract (emphasis added); claim 15 (reciting “one or more first memory devices . . . having a first set of attributes” and “one or more second memory devices . . . having a second set of attributes.”). More specifically, Ware discloses a “usage field 208” that “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212*.” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches a first device profile, or set of attributes, associated with a first memory location (the first memory portion comprising DRAM) and a second device profile, or set of attributes, associated with a second memory location (the second memory portion comprising Flash).

682. Annotated Figure 2, shown below, illustrates the **device profile** (i.e., the usage field used to store a first and second sets of attributes), which is associated with both the **first memory location** (in the first memory device) and the **second memory location** (in the second memory device):

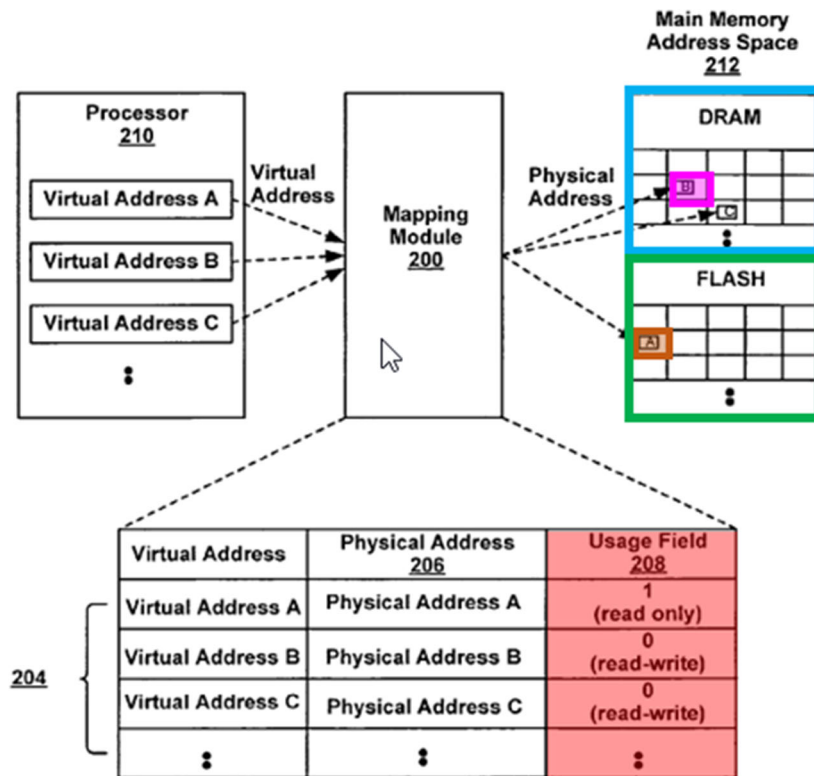


Figure 2

683. For at least these reasons, it is my opinion that Ware discloses “a memory controller directly coupled to and disposed to address a first memory location and a second memory location which are respectively associated with a first device profile and a second device profile.”

- c. [Claim 98b]: “wherein the memory controller is directly coupled to a host as the memory controller

addresses the first memory location or the second memory location;”

684. It is my opinion that Ware discloses this claim element. Ware teaches that the host “processor 108 is coupled to the memory controller 102 via bus 124 and processor interface logic 122.” Ex-1006 at 7:9-10. Ware also teaches that the first and second memory devices “are coupled to the memory controller 102 via memory interfaces 112 and 114, respectively.” *Id.* at 4:4-6; *see also id.* at 7:22-25 (“The memory controller 102 includes memory interfaces 112, 114 for coupling the memory controller 102 to the memory devices in two or more distinct portions (104, 106) of main memory 101.”).

685. Annotated Figure 1, shown below, further illustrates that the **memory controller** (memory controller 102) is coupled to the **host processor** (processor 108) via **bus 124** and **communication paths** as it addresses the **first memory location** or **second memory location**:

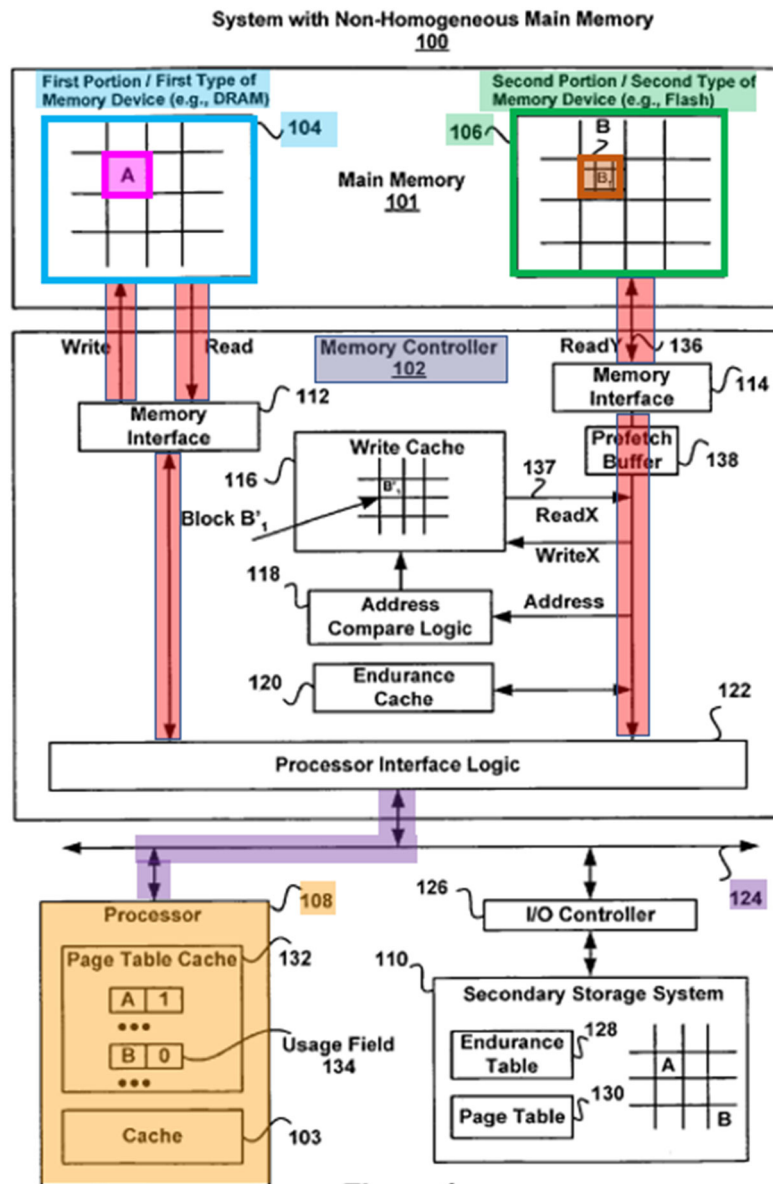


Figure 1

686. For at least these reasons, it is my opinion that Ware discloses “said memory controller directly coupled to a host as said memory controller addresses the first memory location or second memory location.

- d. [Claim 98c]: “wherein said first device profile is optimal for a data type subject to a memory

transaction, wherein said data type comprises one of a random data type or a sequential data type;”

687. It is my opinion that Ware discloses this claim element. As explained above, Ware discloses a first memory location comprising DRAM having a device profile. A POSITA would have understood that a random read command results in accessing a byte of data at some random location in the memory. Thus, a POSITA would understand that the device profile for DRAM would be optimal for a random data type. Indeed, Ware teaches that “[m]ain memory is typically implemented with Dynamic Random [data type] Access Memory (DRAM),” which “has a near-***optimal combination of operational attributes***”—comprising a device profile—“for implementing main memory.” Ex-1006, 1:65-2:6; *see also id.* at 11:1-4 (disclosing a “usage field 208” or profile that “stores data indicative of the usage model of the page, which can be used by processor 210 to ***exploit the unique attributes of a physical memory device***,” such as DRAM’s suitability for random data) (all emphasis added).³⁹

³⁹ Similarly, a POSITA would have understood that a second device profile for flash memory, particularly NAND flash, utilizes page read commands that result in sequentially accessing bytes of data located in a row location (page) of the memory. *See supra* note 6. As a result, flash memory would have a device that is optimal for

688. For at least these reasons, it is my opinion that Ware discloses “wherein said first device profile is optimal for a data type subject to a memory transaction, wherein said data type comprises one of a random data type or a sequential data type.”

- e. **[Claim 98d]: “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device;”**

689. It is my opinion that Ware discloses this claim element. Ware teaches that, “[a]fter the initialization phase is complete, the memory controller waits [at] 302 for a read or write transaction request.” Ex-1006 at 12:12-14. POSITA would readily understand that performing the read or write transaction request requires the memory controller to identify command details for causing the memory transaction to be performed.

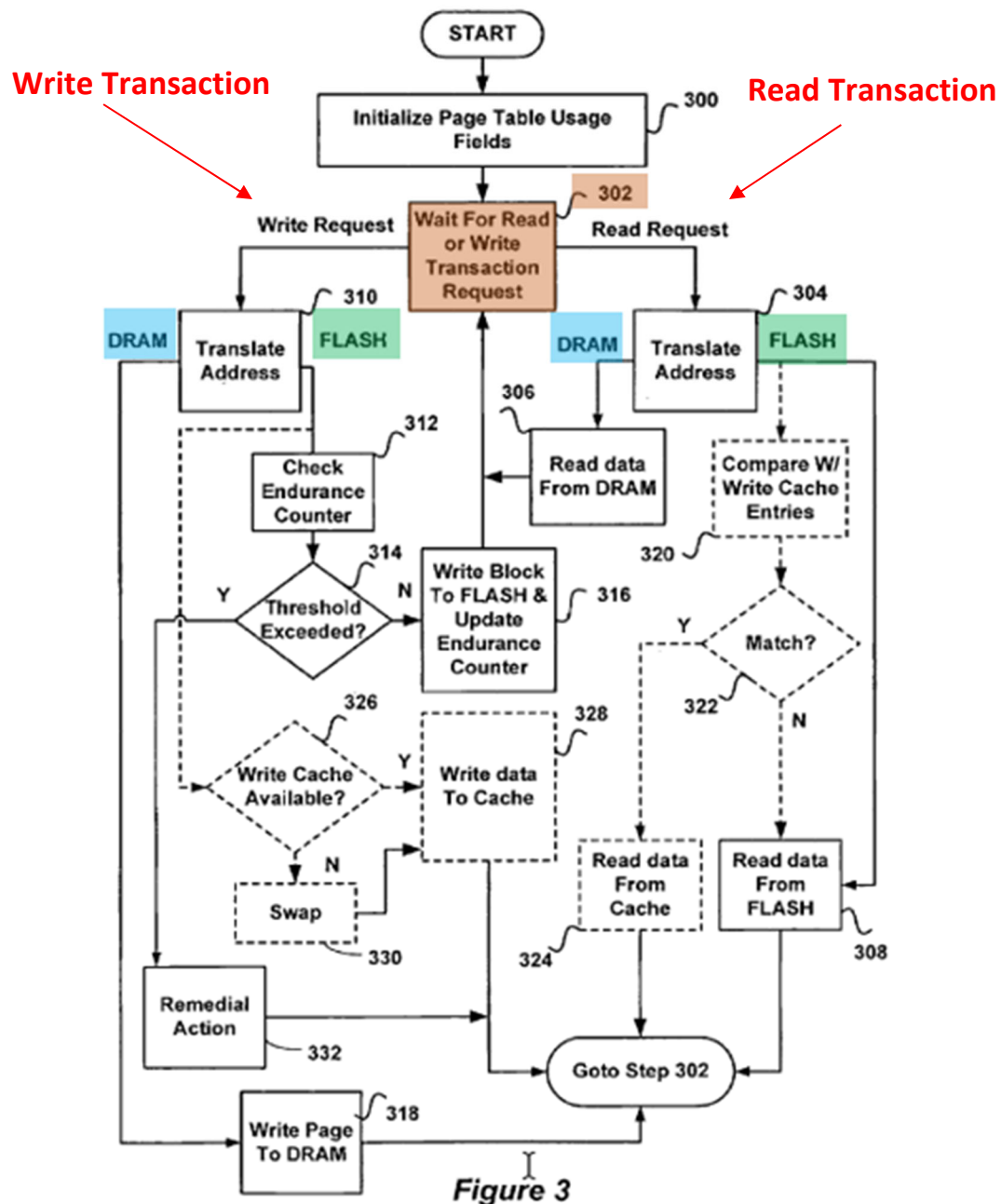
690. For example, Ware teaches that “[i]f a read request is received, the virtual address is translated into a physical address at step 304.” *Id.* at 12:14-15. Moreover, “[o]ne aspect of step 304 is reading the usage field for the page. If the usage field indicates that the page (i.e., the page containing the specified address from which data is to be read) should be read from DRAM, then the data (which may contain program code) is read at step 306 from DRAM and returned to the

a sequential data type.

requestor.” *Id.* at 12:17-22. Likewise, “[i]f the transaction request is a write request, the virtual address is translated into a physical address at step 310, which once again includes reading the usage field for the page containing the specified address. If the usage field indicates that the write data should be written to DRAM (i.e., to a first portion of main memory), then the write data is written to DRAM at step 318, and then the process returns to step 302 to wait for another transaction request.” *Id.* at 12:38-45.

691. A POSITA would understand that command details would be required for executing these steps of read or write request. As such, Ware requires identifying command details for identifying a first memory location and causing a memory transaction to be performed on the first memory location.

692. Annotated Figure 3, below, further illustrates determining a read or write commence at step 302, which requires **identifying command details** for causing the memory transaction to be performed for the **first memory device** (DRAM):



693. For at least these reasons, it is my opinion that Ware teaches “said memory controller identifies command details for causing the memory transaction to be performed, wherein said command details comprising the first memory device.”

- f. **[Claim 98e]: “the first device profile representing a first set of attributes of the first memory location, and the second device profile representing a second set of attributes of the second memory location;”**

694. It is my opinion that Ware discloses this claim element. For example, Ware discloses that both the first memory location (first memory portion 104) and second memory location (second memory portion 106) have a set of attributes comprising a device profile. As a result, Ware teaches “one or more first memory devices of a first memory type having a *first set of attributes*, and a second memory interface adapted to be coupled to one or more second memory devices of a second memory type having a *second set of attributes*.” Ex-1006, Abstract (emphasis added); claim 15 (reciting “one or more first memory devices . . . having a first set of attributes” and “one or more second memory devices . . . having a second set of attributes”). More specifically, Ware discloses a “usage field 208” that “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the *unique attributes of a physical memory device associated with the main memory address space 212*.” *Id.* at 11:1-4 (emphasis added). In one embodiment of Ware, such as Figure 2, the usage model is a one-bit flag that indicates whether the requested page is expected to be read-only or read-write. However, Ware discloses other embodiments: “In the above example, the usage field 208 is a one-bit flag that indicates whether the requested page is expected to be read-only or whether it is to

be read-Write. It should be apparent, however, that the usage field 208 can represent other page usage models, and include more or fewer bits.” *Id.* at 11:19-23. In this manner, Ware teaches a first device profile, or set of attributes, associated with a first memory location (the first memory portion comprising DRAM) and a second device profile, or set of attributes, associated with a second memory location (the second memory portion comprising Flash).

695. Annotated Figure 2, shown below, illustrates the **device profile** (i.e., the usage field used to store a first and second sets of attributes), which is associated with both the **first memory location** (in the first memory device) and the **second memory location** (in the second memory device):

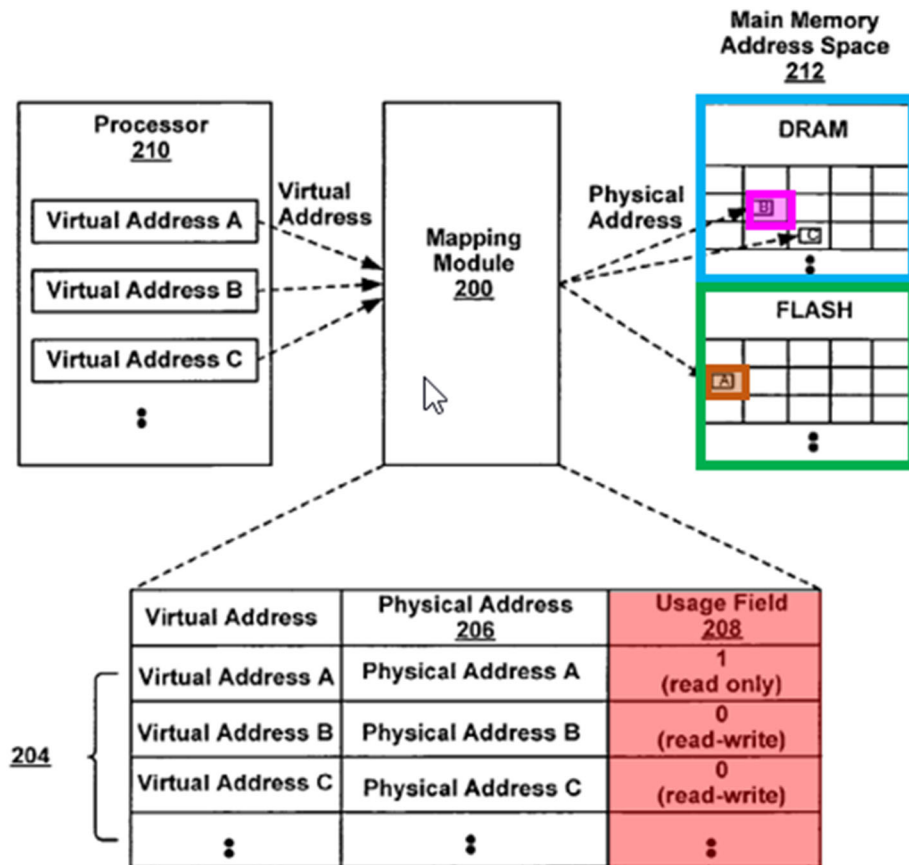


Figure 2

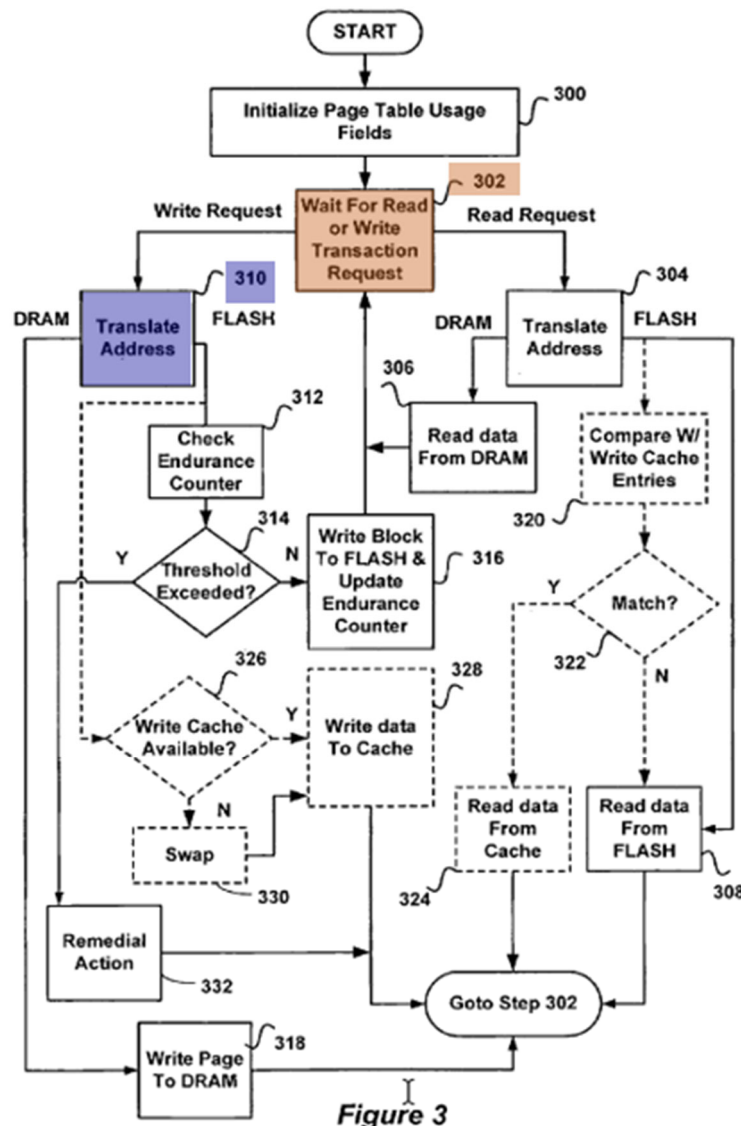
696. For at least these reasons, it is my opinion that Ware discloses “the first device profile representing a first set of attributes of the first memory location, and the second device profile representing a second set of attributes of the second memory location.”

- g. [Claim 98f]: “said memory controller obtaining the first set of attributes after identifying the command details;”

697. It is my opinion that Ware discloses this claim element. As explained above, Ware teaches identifying command details for causing a memory transaction to be performed. *Supra* § X.B.1.h; *see also supra* § X.B.1.i. Ware further teaches that the first of attributes are obtained after the command details are identified. For example, in relation to Figure 3 reproduced below, Ware teaches that once it has been determined that a transaction request is a write request at step 302—i.e., once the command details for a write transaction have been identified—***then*** a virtual address is “translated into a physical address at step 310, which once again ***includes reading the usage field*** for the page containing the specified address.” Ex-1006 at 12:38-41; *see also id.* at 12:41-45 (explaining that the usage field then indicates whether the write data should be written to DRAM). The usage field “stores data indicative of the usage model of the page, which can be used by processor 210 to exploit the ***unique attributes of a physical memory device associated with the main memory address space 212.***” *Id.* at 11:1-4 (emphasis added). In this manner, Ware teaches that, after identifying the command details for a read or write (as shown in step 302 of annotated Figure 3 below), a first set of attributes in the usage field, which describe the corresponding usage model for the first memory location in memory device 104, are obtained (as shown in step 310 of annotated Figure 3 below).

698. Annotated Figure 3, below, illustrates that—after the **command details**

are identified in step 302—the address is translated in step 310, which requires **obtaining the set of attributes**:



699. A POSITA would also understand that the memory controller obtains the first set of attributes after identifying the command details. As illustrated above in annotated Figure 3, the host 108 sends the command details to the memory

controller for identification (at step 302) via the processor interface logic 122 (which is part of the means for addressing). In particular, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses *and control signals* to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction*, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added).

700. For at least these reasons, it is my opinion that Ware discloses “said memory controller obtaining the first set of attributes after identifying the command details.”

h. [Claim 98g]: “and wherein said memory controller uses attributes from the first and second device profiles;”

701. It is my opinion that Ware in view of a POSITA’s knowledge renders obvious this claim element. As explained above, Ware teaches that the first and second device profiles (i.e., usage fields) include a variety of attributes. *See supra* § X.B.1.i. Ware also discloses using the attributes from each of these device profiles. For example, Ware teaches that “NOR Flash memory devices typically have different granularity for read and write operations. In particular, the minimum unit

for reading data from Flash memory may be a word (e.g., 16 bits) while the minimum unit for writing data to Flash memory is typically significantly larger than the minimum unit for reading, and is called either a block or a group of memory cells. In some Flash memory devices, each block includes between 1024 and 65,536 words, with blocks typically containing between 4 k (4096) and 32 k (32,768) Words.” Ex-1006 at 5:65-6:7. Ware thus teaches different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. Ware goes on to teach that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes,” thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. *See id.* at 6:7-8.

702. Similarly, Ware teaches use of attributes such as the ratio of write access time to read access time in each device profile. For example, “[a] typical NOR Flash memory device may have a read access time of less than 100 ns, and an erase-write time of more than 1 millisecond. Thus, the ratio of write access time to read access time may exceed 10,000 to 1, and will typically be greater than 100 to 1. This attribute differs from the write access time of, for example, DRAM, which is typically about the same as the read access time of DRAM devices. For example, the

read access time and write access time of a DRAM device are typically both less than 100 ns, and often both are between 20 ns and 40 ns. The ratio of write access time to read access time of DRAM devices is generally less than 4 to 1, and is typically less than 2 to 1.” *Id.* at 4:19-31.

703. Ware in view of a POSITA’s knowledge also renders the use of these attributes by a memory controller obvious. For example, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses *and control signals* to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address *to determine whether the first or second portion of main memory will receive a memory access transaction*, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added). A POSITA would understand that, by decoding bits of a physical address to “determine whether the first or second portion of main memory will receive a memory access transaction,” the processor interface logic in the memory controller uses attributes from the first and second device profiles.

704. For at least these reasons, it is my opinion that Ware in view of a POSITA’s knowledge renders obvious “wherein said memory controller uses attributes from the first and second device profiles.”

- i. **[Claim 98h]: “wherein said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”**

705. It is my opinion that discloses this claim element. As explained above, Ware teaches the use of a variety of different attributes, including different granularity for read and write operations resulting in the need to select a transfer size (granularity) for the memory transaction. *Supra* §§ X.B.1.i, X.B.1.o. Ware further teaches that “[s]ome Flash memory devices have blocks or groups of memory cells of two or more distinct sizes,” thereby disclosing that transfer size is a function of the data size (a particular size from one of the possible distinct sizes) of the memory transaction and the first set of attributes. Ex-1006 at 6:7-8.

706. Ware also discloses that “it may be noted that a page, which is the basic unit of data or storage in a virtual memory system is typically, but not necessarily, of a different size than a block in a Flash memory device. Thus, a page may be stored ***in a plurality of blocks***, or ***one block could store multiple pages***, or ***each page could be stored in a single block.***” *Id.* at 6:9-14 (emphasis added). In view of these size differences between pages and blocks, which can result in pages being stored either within a block or across blocks, the transfer size would be a function of the data size and the first set of attributes. *See also supra* §§ X.B.1.i, X.B.1.o.

707. Ware in view of a POSITA’s knowledge also renders using the memory

controller selecting a transfer size for the memory transaction obvious. For example, Ware teaches that “[t]he processor interface logic 122 may include signal conditioning circuitry and other devices for transmitting data, addresses ***and control signals*** to and from the bus 124. For example, the processor interface logic 122 may include decoding logic for decoding one or more bits of a physical address ***to determine whether the first or second portion of main memory will receive a memory access transaction***, as described below with respect to FIG. 2.” *Id.* at 7:46-54 (emphasis added). A POSITA would understand that, by decoding bits of a physical address to “determine whether the first or second portion of main memory will receive a memory access transaction,” the memory controller selects a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.

708. For at least these reasons, it is my opinion that Ware discloses “wherein said memory controller selecting a transfer size for the memory transaction, wherein the transfer size is a function of a data size of the memory transaction and the first set of attributes.”

XI. CONCLUSION

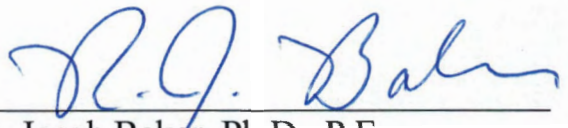
709. For the foregoing reasons, it is my opinion that claims 1, 5, 6, 29, 36, 37, 45, 51, 52, 59, 71, 76, 80, 84, 85, and 98 of the ’190 patent are obviated by

Declaration of R. Jacob Baker
Inter Partes Review of U.S. Patent No. 9,135,190

Cornwell and rendered obvious by Ware in view of a POSITA's knowledge.

710. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: March 22, 2023

By: 
R. Jacob Baker, Ph.D., P.E.