

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD. AND
SAMSUNG ELECTRONICS AMERICA, INC.,
Petitioners,

v.

RADIAN MEMORY SYSTEMS, LLC,
Patent Owner.

Case IPR2025-01377

U.S. Patent No. 11,347,656

**DECLARATION OF DR. HEINER LITZ IN SUPPORT OF PETITION
FOR INTER PARTES REVIEW OF U.S. PATENT NO. 11,347,656**

TABLE OF CONTENTS

	<u>Page</u>
I. SUMMARY OF OPINIONS	1
II. QUALIFICATIONS	1
III. MATERIALS CONSIDERED	3
A. Person of Ordinary Skill in the Art	3
B. Claim Construction	4
C. Presumption of Validity	4
D. Obviousness	4
IV. LEVEL OF ORDINARY SKILL IN THE ART	7
V. DETAILED OPINIONS	7
A. Technology Background And State Of The Art	7
B. Summary Of Patents	10
VI. OVERVIEW OF PRIOR ART REFERENCES	24
A. Ban	24
B. Reiter	26
C. Sinclair-367	28
D. Yamada	30
E. ZBC, SBC-4, and SPC-4	31
VII. INVALIDITY OF CLAIMS	33
A. GROUND I: CLAIMS 1-12, 16, 20-23 ARE RENDERED OBVIOUS BY REITER IN LIGHT OF ZBC, SBC-4, AND SPC-4	33
B. GROUND II: CLAIMS 1, 13-14, 19 ARE RENDERED OBVIOUS BY REITER IN LIGHT OF ZBC, SBC-4, SPC-4, AND SINCLAIR-367	81
C. GROUND III: CLAIMS 15, 17-18 ARE RENDERED OBVIOUS BY REITER IN LIGHT OF ZBC, SBC-4, SPC-4, SINCLAIR-367, AND YAMADA	91

D.	GROUND IV: CLAIMS 1-14, 16, 19-23 ARE RENDERED OBVIOUS BY BAN IN LIGHT OF ZBC, SBC-4, SPC-4, AND Sinclair-367.....	96
E.	GROUND V: CLAIMS 15, 17-18 ARE RENDERED OBVIOUS BY BAN IN LIGHT OF ZBC, SBC-4, SPC-4, SINCLAIR-367, AND YAMADA.....	124

DECLARATION OF DR. HEINER LITZ, Ph.D.

I, HEINER LITZ, Ph.D. hereby declare as follows:

1. My name is Heiner Litz. I have been retained to testify as an expert in this Inter Partes Petition on behalf of Samsung Electronics Co., Ltd. (“SEC”) and Samsung Electronics America, Inc. (“SEA”) (collectively, “Samsung” or “Petitioner”).

2. I have prepared this declaration in connection with Samsung’s related petition for inter partes review (“IPR”) of U.S. Patent No. 11,347,656 (the “’656 Patent,” EX1001), which I am informed is being filed concurrently with this declaration. I have been asked to provide certain opinions relating to the patentability of the ’656 patent. Specifically, I have been asked to provide my opinion regarding whether claims 1-23 of the ’656 patent (“the Challenged Claims”) are invalid as obvious in view of the prior art.

I. SUMMARY OF OPINIONS

3. It is my opinion that the Challenged Claims are rendered obvious pursuant to Grounds I-V and the reasons discussed herein.

II. QUALIFICATIONS

4. My education, experience, and qualifications are detailed in my résumé, included as Exhibit 1 to this declaration. While I incorporate Exhibit 1 by reference, below is a brief summary of my background, including my background and experience relevant to this case.

5. I earned a degree in Computer Engineering from the University of Mannheim, Germany, in 2006. In 2012, I completed a Ph.D. in Computer Engineering from the same institution, graduating Summa Cum Laude. Between 2006 and 2012, my graduate research focused on computer architecture, hardware chip design, scalable storage systems, and high-performance computing networks.

6. In 2013, I began my academic career as a Postdoctoral Researcher at Stanford University, working in the research groups of Professors David Cheriton and Christos Kozyrakis. From 2012 to 2016, my research centered on programmable “Open-Channel” solid-state drives (SSDs). I introduced techniques to reduce access latency by shifting performance-critical tasks such as data placement from the SSD to the host processor. This work led to “RAIL: Predictable, Low Tail Latency for NVMe Flash”, which achieved a $7\times$ reduction in read access latency through host-orchestrated SSD data placement.

7. In 2017, I served as Visiting Faculty at Google, working on programmable storage devices within the Platforms Infrastructure Group.

8. In 2018, I joined the Computer Science Department at the University of California, Santa Cruz (UCSC), as an Assistant Professor. I was promoted to Associate Professor with tenure in 2023. In 2024, I was named the Kumar Malavalli Endowed Professor for Storage Systems Research, a position supported by a \$1 million gift from Kumar Malavalli.

9. Since 2023, I have served as Director of the Center for Research in Storage Systems (CRSS) at UCSC, a center comprised of five faculty and 15 Ph.D. students. CRSS advances research in storage systems and is supported by the National Science Foundation (NSF) and industry sponsors. Our work spans a range of topics, including programmable storage and emerging technologies such as Open-Channel, Zoned Namespace (ZNS), and Flexible Data Placement (FDP) SSDs. In the context of ZNS, my group proposed the group append feature to optimize ZNS SSD integration with file systems (*e.g.*, “Append is Near: Log-based Data Management on ZNS SSDs”).

10. In 2024, I was a Visiting Professor at the Massachusetts Institute of Technology (MIT) in Cambridge.

11. Over the course of my academic career, I have published more than 80 peer-

reviewed research papers, journal articles, and patents in the areas of storage systems and computer hardware. I have advised over 15 Ph.D. students and secured more than \$9 million in research funding. My honors include the NSF CAREER Award, Intel Outstanding Researcher Award, and multiple Faculty Awards from Google and Facebook. I received the Best Paper Award at MICRO 2023 and several IEEE MICRO Top Picks. I have also served as Program Chair for HotChips 2023 and NVMW 2024, and as Associate Editor for IEEE Computer Architecture Letters (CAL).

III. MATERIALS CONSIDERED

12. Counsel for Samsung has informed me of the following legal standards. I am not an attorney, and I am relying only on instructions from Samsung's attorneys for these legal standards.

A. Person of Ordinary Skill in the Art

13. I understand that issues of claim construction and invalidity are analyzed from the perspective of a person of ordinary skill in the art ("POSITA").

14. I understand that a POSITA is a hypothetical person who is used to analyzing the prior art without the benefit of hindsight. A POSITA is presumed to be one who thinks along the lines of conventional wisdom in the art.

15. I understand that the hypothetical POSITA is presumed to have knowledge of all references that are sufficiently related to one another and to the pertinent art, and to have knowledge of all arts reasonably pertinent to the particular problem that the claimed invention addresses.

16. I understand that factors considered in determining the level of ordinary skill in the art at the time of an invention include (1) the type of problems encountered in the art; (2) the prior art solutions to those problems; (3) the educational level of active workers in the field; (4) the rapidity with which innovations are made; and (5) the sophistication of the technology. I further

understand that the inventors' specific levels of skill are not relevant to this inquiry.

B. Claim Construction

17. I have been informed by counsel that the Patent Trial and Appeal Board (“PTAB”) applies the same claim construction standard used in district courts, where the claims are given their ordinary meaning as understood by one skilled in the art at the time of the invention, informed by the claim language itself, the specification, and the prosecution history. I also understand that “extrinsic evidence”—*i.e.*, evidence other than the patent and prosecution history—can be relevant in determining how a POSITA would understand terms of art used in the claims. I have been informed, however, that extrinsic evidence may not be used to contradict the meaning of the claims as described in the intrinsic evidence—*i.e.*, evidence in the claim language itself, the specification, and the prosecution history.

C. Presumption of Validity

18. I understand that a granted patent is presumed to be valid. I understand that the presumption of validity can be overcome if clear and convincing evidence is presented that proves the patent is invalid.

D. Obviousness

19. I understand that even if a patent claim is not anticipated, it is still 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 at the time the invention was made to a person of ordinary skill in the pertinent art.

20. I understand that a person of ordinary skill in the art 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 insight or hindsight in deciding whether a claim is obvious.

21. I also 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 or objective indicia of obviousness or non-obviousness.

22. I understand that an obviousness determination can be based on a single prior art reference or a combination of multiple prior art references or the knowledge of one of ordinary skill in the art. I understand that the prior art itself may provide a suggestion, motivation, or reason to combine or modify the teachings of the prior art, or that such a reason may come from other sources, such as the knowledge of a person having ordinary skill in the art, common sense, and market forces. I understand that the following rationales may support a finding of obviousness:

- combining prior art elements according to known methods to yield predictable results;
- simple substitution of one known element for another to obtain predictable results;
- use of known technique to improve similar devices (methods, or products) in the same way;
- applying a known technique to a known device (method or product) ready for improvement to yield predictable results;
- “obvious to try” - choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;
- some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

23. I understand that an obviousness determination requires a reasonable expectation of success in achieving the claimed invention.

24. I understand that secondary considerations or objective indicia of obviousness or

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 or lack of 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) the taking of licenses under the patent by others; (6) deliberate copying of the invention; (7) teaching away; and (8) the simultaneous invention of the claimed subject matter. 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.

25. I understand that any secondary considerations or objective indicia must bear a nexus to the claimed invention. Where the offered secondary consideration actually results from something other than what is both claimed and novel in the claim, there is no nexus to the merits of the claimed invention. I further understand that the patentee bears the burden of demonstrating that the relevant commercial success is attributable to the claimed invention, as opposed to other economic and commercial factors unrelated to the technical quality of the patented subject matter.

26. I understand that whether a prior art reference “teaches away” from the claimed invention is a fact to be considered in determining obviousness. I understand that it is improper to combine references where the references teach away from their combination. However, I understand that the nature of the teaching is highly relevant and must be weighed in substance – a known or obvious composition does not become patentable simply because it has been described as somewhat inferior to some other product for the same use. Furthermore, I understand that the prior art's mere disclosure of more than one alternative does not constitute a teaching away from any of these alternatives because such disclosure does not criticize, discredit, or otherwise discourage the solution claimed.

IV. LEVEL OF ORDINARY SKILL IN THE ART

27. In my opinion , a person of ordinary skill in the art at the time of the '656 Patent (below) had least a Bachelor of Science degree in electrical engineering, or an equivalent field, and at least three years of experience with SSDs or other flash memory devices.

28. I meet these criteria now and met them at the time of the alleged invention.

29. Under this formulation, my opinions expressed in this declaration apply.

V. DETAILED OPINIONS

A. Technology Background And State Of The Art

30. Solid-state drives (SSDs) are semiconductor-based data storage devices that provide large storage capacities, non-volatility, and high read and write performance. SSDs are a mature and widely adopted storage technology, present in nearly all modern smartphones, laptops, and servers.

31. SSDs store data on NAND flash memory, which is composed of numerous floating-gate transistors or Charge Trap Flash (CTF) memory cells. To achieve high storage density, SSDs group many of these transistors into units of variable sizes. Data can only be read, written, and erased at the granularity of these units. Notably, the read unit size—often 4KB—can differ from the write unit size, which is generally larger (*e.g.*, 64KB or more), and the erase unit size, which may have a size of *e.g.* 256KB.

32. A key characteristic of SSDs is that data cannot be modified in place without first performing an explicit erase operation. This constraint means each block of data can only be written once, unless it is being explicitly erased. To enable data modification, SSDs employ an *out-of-place update* strategy: when data is updated, the new version is written to a different location instead of overwriting the old one. This results in two versions of the same data element coexisting—the outdated version and the newly written version.

33. To ensure that read operations retrieve the most recent data, SSDs utilize a *flash translation layer* (FTL), which manages a *logical-to-physical (L2P) mapping table*. Applications issue I/O operations using logical block addresses (LBAs), while the SSD stores data at physical block addresses (PBAs). The L2P table ensures that each LBA always maps to the PBA containing the most recently written data.

34. For example, assume the SSD has a read granularity of 4KB and the application writes a 4KB block to LBA=2. The SSD stores it at PBA=12 and updates the L2P so that LBA=2 maps to PBA=12. Upon a subsequent overwrite of LBA=2, the SSD writes the new block to PBA=13, leaving PBA=12 unchanged. The L2P is updated to map LBA=2 to PBA=13. If the write unit size is larger (*e.g.*, 64KB), a write buffer is used to coalesce multiple 4KB writes (*e.g.*, 16 blocks) into a single 64KB write operation.

35. Since each write consumes a previously unwritten block, the SSD will eventually run out of such blocks. To reclaim space, SSDs support an *erase* operation, which may be directed to an erase block or erase unit. SSDs may also operate at another level of granularity—commonly 256KB—where erases, for example, are conducted for 64 consecutive 4KB blocks. This unit is also referred to as a *reclaim unit*, *subdivision*, or *zone*, all of which are functionally equivalent.

36. To safely erase a unit, it must not contain any *valid* blocks. A valid block is defined as a block that is still referenced by the L2P and thus represents the latest version of a data element. If no such erase unit is entirely free of valid blocks, the SSD performs *garbage collection*. During this process, the SSD itself rewrites valid blocks to new locations, updates the L2P accordingly, and invalidates the old PBAs. Once all valid blocks have been relocated, the erase unit can be safely erased. Garbage collection involves additional reads and writes, and therefore, should be minimized to improve performance and endurance, as NAND flash has a limited number of program/erase (P/E) cycles.

37. To optimize garbage collection, the SSD maintains metadata to track the number of valid blocks per erase unit. Selecting an erase unit with the fewest valid blocks minimizes the number of costly copy operations. Furthermore, it is beneficial to segregate frequently updated (*hot*) data from rarely updated (*cold*) data by placing them into separate erase units. This increases the likelihood of finding hot erase units with few valid blocks. One implementation approach is to maintain multiple open units to which writes can be applied: one for cold LBAs and one for hot LBAs. More than two units may be maintained, with write frequency tracked using finer-grained *temperature* metrics.

38. Due to the limited endurance of NAND flash cells (*e.g.*, a floating-gate transistor may only withstand ~10,000 P/E cycles), *wear-leveling* is essential. Wear-leveling moves cold data from infrequently used units, enabling those blocks to be erased and reused, thereby balancing wear across all units in the SSD.

39. In summary, the FTL performs several key tasks:

- L2P Management: Mapping LBAs to the most recent PBAs for servicing reads.
- Data Placement: Determining whether hot or cold data should be written to a given unit.
- Garbage Collection: Reclaiming space by relocating valid data and erasing blocks.
- Wear-Leveling: Ensuring uniform wear across erase units by migrating cold data.
- Erase Management: Erasing only those units that no longer contain valid data.

40. In traditional SSDs, the FTL is implemented entirely within the SSD controller. This architecture presents a simple hardware interface to software, exposing only read and write operations at fixed granularities (*e.g.*, 4KB).

41. An alternative approach is to implement the FTL—fully or partially—on the *host processor*. Host-based FTLs can leverage application-specific knowledge (*e.g.*, data temperature) to make more informed decisions about data placement, garbage collection, and wear-leveling. For instance, garbage collection can be deferred to periods when the application is idle, thereby improving performance and efficiency.

42. The sharing of memory management responsibilities between the host and the controller has been termed, *i.e.*, “Cooperative memory management.” For example, Samsung’s U.S. Pat. Pub. 2008/0189485A1, describes “method[s] of cooperative memory management between a host and a memory card device includes selecting at least one of several memory management functions to be performed by the host for the device.” EX1010, Abstract.

43. The operations implemented by the FTL are functionally equivalent regardless of whether they are executed on the SSD or on the host processor. In fact, in most commercial SSDs, the FTL is implemented in software (firmware) and executed on an embedded processor within the SSD. To enable FTL functionality on the host side, the SSD must expose additional interfaces to the host—such as support for explicit erase operations and fine-grained control over data placement for instance by allowing host software to directly read and write PBAs and access meta data from the SSD controller.

B. Summary Of Patents

1. The '656 Patent

44. The '656 Patent issued May 31, 2022, and is titled “Storage Drive With Geometry Emulation Based On Division Addressing And Decoupled Bad Block Management.”

2. The Challenged Claims

45. Below is a table described the Challenged Claims, including claims 1-23.

Claim 1	
1[pre]	A storage device comprising:
1[a]	flash memory having physical erase units; and
1[b]	circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:
1[b][i]	derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,
1[b][ii]	identify an addressed block device from the first address portion,
1[b][iii]	identify an addressed segment within the addressed block device from the second address portion, and
1[b][iv]	identify an addressed storage location within the addressed segment from the third address portion,
1[c]	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,
1[d]	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;
1[e]	wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests.
Dependent Claims	

2[pre]	The storage device of claim 1 wherein:
2[a]	said circuitry comprises at least one processor; and
2[b]	the storage device further comprises instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor.
3[pre]	The storage device of claim 1 wherein:
3[a]	the incoming data access requests are read requests received from a host;
3[b]	the storage device can receive a write request from the host, wherein the circuitry is to, for the received write request, derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion, identify an addressed segment within the addressed block device from the second address portion, identify a corresponding logical erase unit from the third address portion and identify a physical erase unit which is mapped to the corresponding logical erase unit;
3[c]	the storage drive is to, in association with fulfillment of the received write request, identify a physical storage location in which associated write data has been stored and which is associated with the corresponding logical erase unit, and to transmit information representing an address value to the host corresponding to the identified physical storage location; and
3[d]	for one of the read requests which is directed to data which has been written into the flash memory pursuant to the received write request, at least one of the first address portion, the second address portion or the third address portion associated with the one of the read requests is dependent on the address value which has been transmitted

	to the host in association with the data which has been written into the flash memory pursuant to the received write request.
4[pre]	The storage device of claim 1 wherein:
4[a]	the storage device is to receive a maintenance request from a host which addresses a block device and a segment within the block device addressed by the maintenance request; and
4[b]	the storage device comprises circuitry to erase, in association with execution of the maintenance request, one or more of the physical erase units which are mapped to the segment addressed by the maintenance request.
5[pre]	The storage device of claim 1 wherein:
5[a]	the flash memory is to have block devices, including the addressed block device; and
5[b]	in connection with a given one of the incoming data access requests, the circuitry is to:
5[c]	identify and retrieve a look-up table specific to the addressed block device which was identified from the first address portion derived from the given one of the incoming data access requests, and
5[d]	identify a physical storage location in the flash memory dependent on the look-up table.
6[pre]	The storage device of claim 1 wherein:
6[a]	the storage device further comprises circuitry to track metadata for the addressed segment, where said metadata indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased; and

6[b]	the storage device further comprises circuitry to transmit information to a host dependent on said tracked metadata.
7[pre]	The storage device of claim 1 wherein:
7[a]	the incoming data access requests are read requests received from a host;
7[b]	the storage device can receive write requests from the host, and the storage device comprises circuitry to issue commands to the flash memory in fulfillment of the write requests;
7[c]	the storage device is to assign an address value in association with fulfillment of each of the write requests, in order to identify a physical storage location in which associated write data has been stored, and comprises circuitry to transmit information representing each assigned address value to the host; and
7[d]	for one of the incoming data access requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, the addressed storage location identified from the third portion is dependent on the assigned address value which was transmitted to the host in association with the one of the write requests.
8[pre]	The storage device of claim 1 wherein:
8[a]	the storage device can receive a request from a host to copy host-selected data from a first storage location in flash memory to a second storage location in the flash memory, and the storage device comprises circuitry to issue commands to the flash memory, in response to the request from the host to copy, to read the host-selected data from the first storage location in the flash memory and write the host-selected

	data to the second storage location in the flash memory without sending the data being copied to the host; and
8[b]	the request from the host to copy is to designate destination address information corresponding to the second storage location, wherein the storage device is to identify from the destination address information a destination block device and an addressed segment within the destination block device, and wherein the second storage location corresponds to the addressed segment within the destination block device.
9[pre]	The storage device of claim 8 wherein
9[a]	the storage device is to identify from the destination address information a destination logical erase unit and is to select the second storage location in dependence on the identified destination logical erase unit.
10[pre]	The storage device of claim 1 wherein
10[a]	the storage device is to store information identifying a number of structures in the flash memory and is to transmit the information identifying the number of structures to a host, wherein the host is to issue at least one request to the storage device in which the at least one request is dependent on the information transmitted to the host identifying the number of structures, and wherein the storage device comprises circuitry to fulfill the at least one request by issuing commands to the flash memory for which a quantum of flash memory selected by the commands which are issued in response to the at least one request is dependent on the information transmitted to the host identifying the number of structures.
11[pre]	The storage device of claim 10 wherein

11[a]	the information transmitted to the host identifying the number of structures identifies an amount of flash memory that is to be erased as a unit.
12[pre]	The storage device of claim 1 wherein
12[a]	the addressed segment corresponds to an integer number of physical erase units for which the failure condition has not been detected, and the third address portion specifies a location within a sequential range of locations that correspond to the integer number of physical erase units.
13[pre]	The storage device of claim 1 wherein:
13[a]	the flash memory is to have block devices, including the addressed block device; and
13[b]	the storage device further comprises circuitry to:
13[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,
13[d]	compare the metadata to at least one threshold,
13[e]	upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and
13[f]	remap the logical erase unit corresponding to the first one of the physical erase units for which the failure condition has not been detected by said storage device to the

	second one of the physical erase units for which the failure condition has not been detected by said storage device.
14[pre]	The storage device of claim 1 wherein:
14[a]	the flash memory is to have block devices, including the addressed block device; and
14[b]	the storage device further comprises circuitry to:
14[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata is dependent on an erase count of associated physical memory space,
14[d]	compare the metadata to at least one threshold,
14[e]	upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and
14[f]	remap the logical erase unit corresponding to the first physical erase unit for which the failure condition has not been detected by said storage device to the second physical erase unit for which the failure condition has not been detected by said storage device.
15[pre]	The storage device of claim 1 wherein:
15[a]	the flash memory is to have block devices, including the addressed block device; and
15[b]	the storage device further comprises circuitry to:

15[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,
15[d]	compare the metadata to at least one threshold, and
15[e]	responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.
16[pre]	The storage device of claim 1 wherein:
16[a]	the flash memory is to have block devices, including the addressed block device; and
16[b]	the storage device further comprises circuitry to:
16[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, and
16[d]	receive a query, issued by a host device, and responsive to the query, compare the metadata to at least one threshold, to identify specific metadata, and transmit information to the host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.
17[pre]	The storage device of claim 1 wherein:
17[a]	the flash memory is to have block devices, including the addressed block device; and
17[b]	the storage device further comprises circuitry to:

17[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and
17[d]	receive a query, issued by a host device, wherein said query is specific to one of the block devices, and responsive to the query, identify specific metadata, associated with the specified one of the block devices, and transmit information to the host device that conveys the specific metadata.
18[pre]	The storage device of claim 1 wherein:
18[a]	the flash memory is to have block devices, including the addressed block device; and
18[b]	the storage device further comprises circuitry to:
18[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and
18[d]	compare the metadata to at least one threshold, and
18[e]	responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.
19[pre]	The storage device of claim 1 wherein:
19[a]	the flash memory comprises multiple planes; and
19[b]	the storage device is to, for each one of a set of the incoming data access requests having sequential addressing, identify a commonly addressed page and write data

	accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes.
20[pre]	The storage device of claim 1 wherein
20[a]	the division operation is applied to subdivide the third address portion into respective first and second bit fields.
Claim 21	
21[pre]	A storage device comprising:
21[a]	flash memory, the flash memory to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment including a mutually exclusive set of one or more logical erase units that each map to a physical erase unit, each segment being associated with only one of the block devices, each logical erase unit being associated with only one of the segments; and
21[b]	circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:
21[b][i]	derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,
21[b][ii]	identify an addressed block device from the first address portion,
21[b][iii]	identify an addressed segment within the addressed block device from the second address portion, and
21[b][iv]	identify an addressed storage location within the addressed segment from the third address portion,

21[c]	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,
21[d]	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;
21[e]	wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests.
Claim 22	
22[pre]	A storage device comprising:
22[a]	flash memory, the flash memory to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment including a mutually exclusive set of one or more logical erase units that each map to a physical erase unit, each segment being associated with only one of the block devices, each logical erase unit being associated with only one of the segments;
22[b]	circuitry to receive incoming data read requests and incoming data write requests and, for each of the incoming data read requests and incoming data write requests, accompanying address information, wherein said circuitry is to, for each of the incoming data read requests:

22[b][i]	derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,
22[b][ii]	identify an addressed block device from the first address portion,
22[b][iii]	identify an addressed segment within the addressed block device from the second address portion, and
22[b][iv]	identify an addressed storage location within the addressed segment from the third address portion,
22[c]	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,
22[d]	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;
22[e]	wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests; and
22[f]	wherein said circuitry is to, for each of the incoming data write requests:
22[f][i]	derive, from the accompanying address information, an associated first address portion, an associated second address portion and an associated third address portion,
22[f][ii]	identify an addressed block device from the associated first address portion,
22[f][iii]	identify an addressed segment within the addressed block device from the associated second address portion,

22[f][iv]	identify a write destination logical erase unit from the associated third address portion, and identify a physical erase unit which is mapped to the write destination logical erase unit,
22[f][v]	identify a physical storage location in which associated write data is to be stored and which is associated with the write destination logical erase unit, and an address value associated with the physical storage location in which the associated write data is to be stored; and
22[f][vi]	transmit information to a host representing the address value;
22[g]	wherein for one of the read requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, at least one of the first address portion, the second address portion or the third address portion associated with the one of the read requests is dependent on the address value which has been transmitted to the host in association with the data which has been written into the flash memory pursuant to one of the write requests.
Claim 23	
23[pre]	A method of operation in a storage device having flash memory, the flash memory having physical erase units, and circuitry, the method comprising:
23[a]	receiving incoming data access requests with the circuitry, wherein each of the incoming data access requests is accompanied by address information;
23[b]	with the circuitry, processing the incoming address information to, for each of the incoming data access requests:
23[b][i]	derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion,

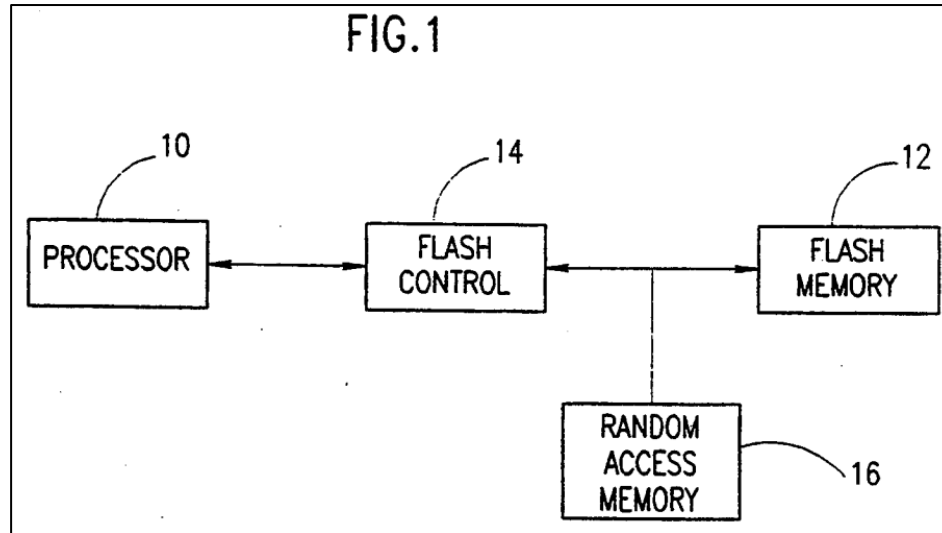
23[b][ii]	identify an addressed segment within the addressed block device from the second address portion,
23[b][iii]	identify an addressed storage location within the addressed segment from the third address portion,
23[b][iv]	perform a division operation to identify a logical erase unit within the addressed segment, and
23[b][v]	identify one of the physical erase units that corresponds to the identified logical erase unit;
23[c]	wherein the method further comprises causing the storage device to detect a failure condition of the physical erase units, to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and to utilize the remapped correspondence to service one or more of the incoming data access requests.

VI. OVERVIEW OF PRIOR ART REFERENCES

A. Ban

46. U.S. Patent No. 5,404,485 (“Ban”) was filed on March 8, 1993, and was issued on April 4, 1995. I understand that it is prior art under AIA 35 U.S.C. § 102(a)(1)-(2).

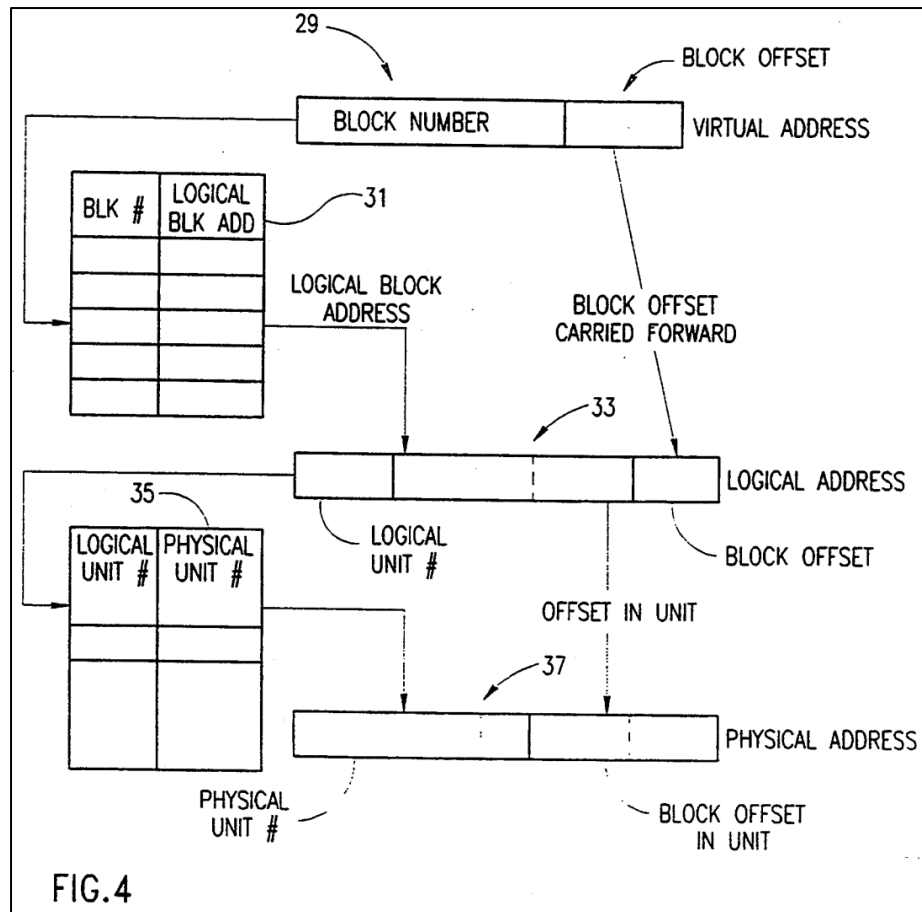
47. Ban teaches a flash memory system comprising flash controller 14, flash memory 12, and random access memory 16, where the flash memory system takes read and write commands from a host—processor 10. EX1005, 3:52-4:5, FIG. 1 (below).



48. The “object[ive] of [Ban’s] invention is the provision of a method (*i.e.*, software, firmware[,] or hardware) to control and manage access to a flash memory so that the flash memory appear to the computer operating system as a data storage device in which it is possible to read data from, and write data to, any flash memory location.” EX1005, 1:52-62. This is because, “in a typical computer system,” a “data storage device to achieve compatibility with the operating system program ... can read data from, and write data to any location” EX1005, 1:28-39.

49. Therefore, Ban discloses a method of managing flash memory to emulate random access memories, which allows reading and writing to any location, because emulating random access memories, like standard hard disk drives, means interoperability with existing operating systems. EX1005, 1:29-68, 3:52-4:5

50. Specifically, Ban’s “flash memory controller 14 provide[s] a fully rewritable virtual address space ... [to] emulate[] a random access memory” via a two-step address translation method. EX1005, 3:52-4:5. With the two-step address translation method, the address accompanying the host’s read or write command is (1) translated to a logical address and then (2) translated to a physical address. EX1005, FIG. 4:

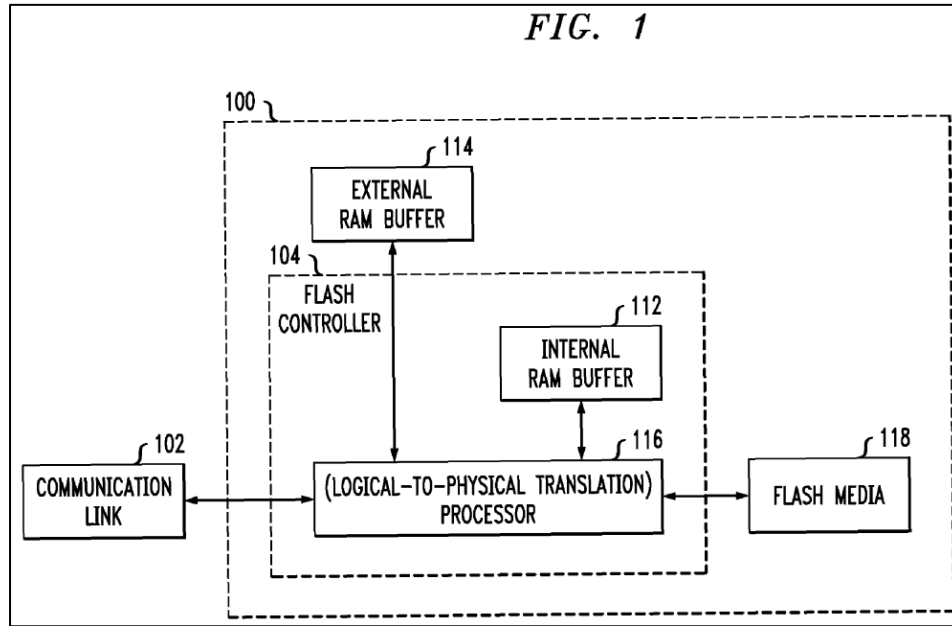


51. This obfuscates the underlying structures of flash memory, where the structures are addressed with a unit number and offset. EX1005, 3:64-4:5. Ban's flash memory is made up of units, thereby requiring a unit number and offset to address a location within flash memory. EX1005, 4:11-24, FIG. 4. Because a typical computing system using hard disk drive only provides a logical block address is provided, which does not include a unit number or offset, Ban's flash controller 14 translates incoming logical block addresses into a unit number and offset. EX1005, 3:52-4:10, 4:51-5:35, FIG. 4.

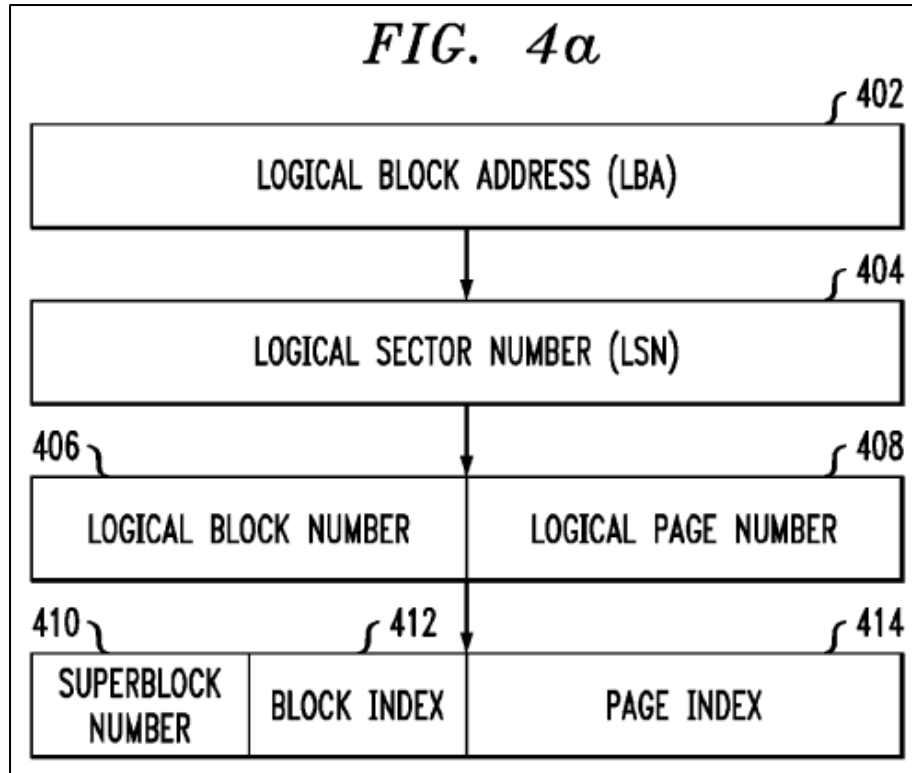
B. Reiter

52. U.S. Patent 8,301,861 ("Reiter") was filed on April 29, 2010, was published on March 24, 2011, and issued on October 30, 2012. I understand that it is prior art under AIA 35 U.S.C. §102(a)(1)-(2).

53. Similar to Ban, Reiter disclose a flash system 100, comprising flash controller 104 for receiving and executing read and write commands from communication link 102 sent by an external computer system (*i.e.*, host). EX1004, 6:63-66, 8:1-6, FIG. 1:



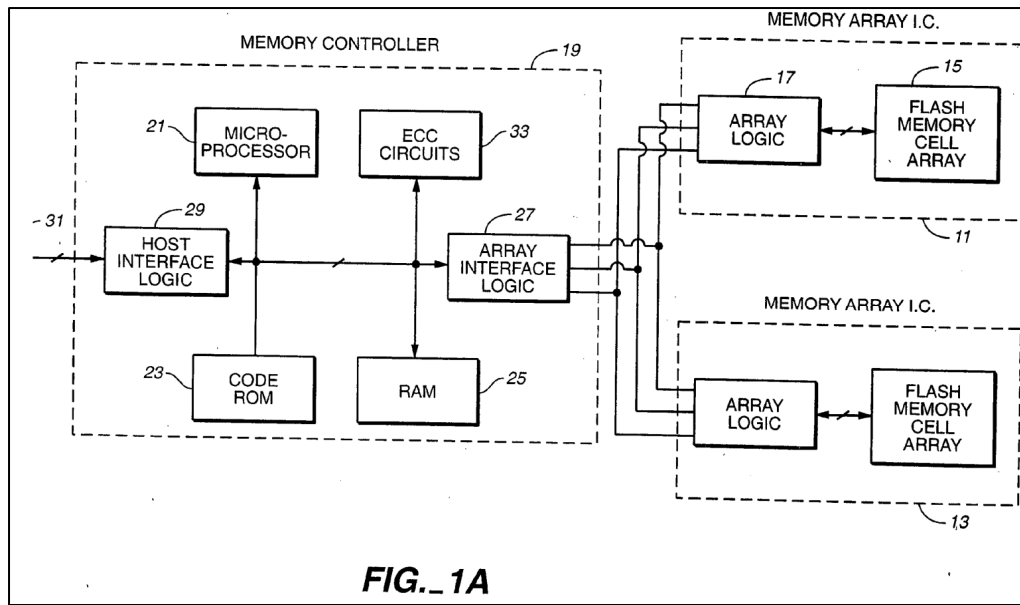
54. The read and write commands are accompanied by a logical address, which corresponds a physical address that details the location in flash media 118. And to execute the command at that location, flash controller 104 performs a logical-to-physical translation method to translate the logical address to a physical address. EX1004, 7:31-35, 8:1-10. Reiter discloses a four-level logical-to-physical translation method including (1) the incoming logical block address 402, (2) logical sector number 404, (3) a bitstring comprising logical block number 406 and logical page number 408, and (4) a bitstring comprising superblock number 410, block index 412, and page index 414. EX1004, 8:54-56, 8:66-9:2, 9:5-20, FIG. 4a:



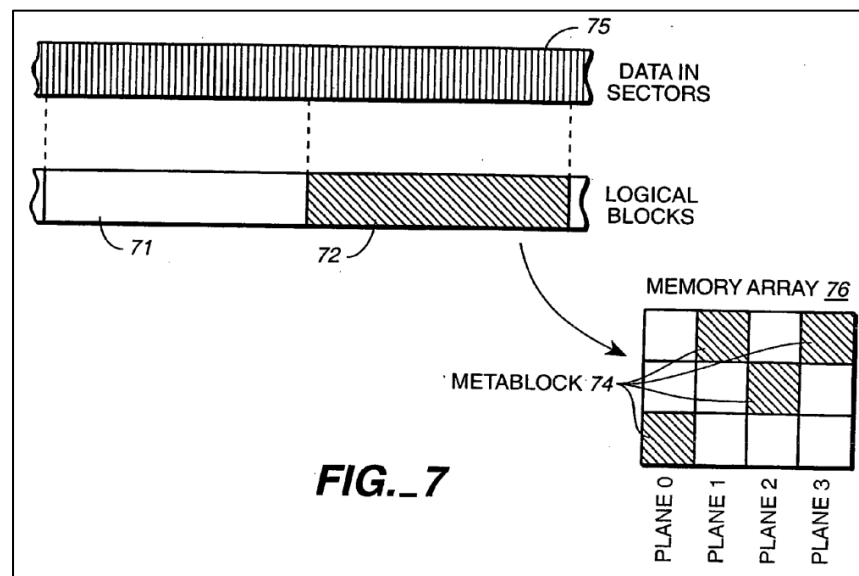
C. Sinclair-367

55. U.S. Patent Publication No. 2005/0144367 (“Sinclair-367”) was filed on December 16, 2004, published on June 30, 2005, and is prior art under AIA 35 U.S.C. § 102.

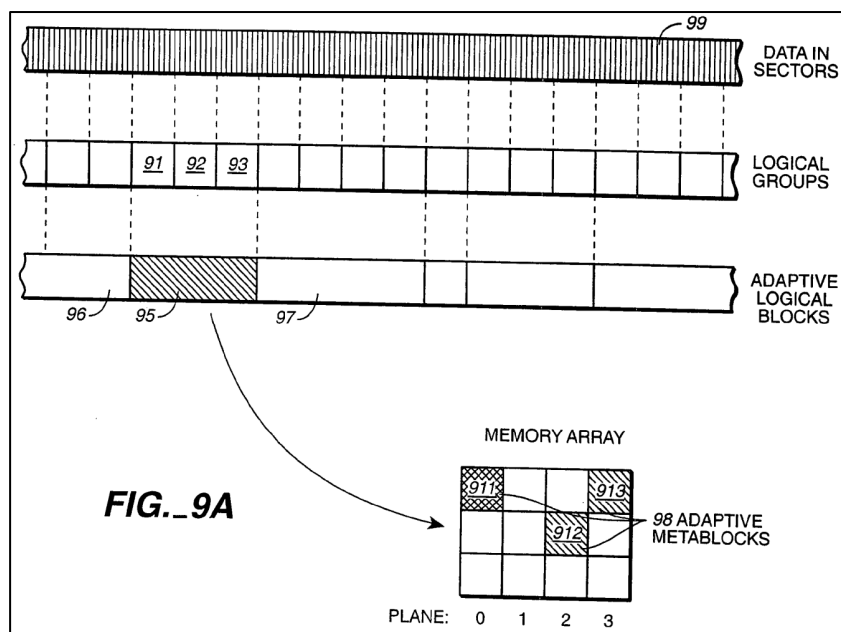
56. Sinclair-367 discloses a non-volatile memory array (*i.e.*, flash memory) via adaptive metablocks. EX1009, Abstract, FIG. 1A:



57. Metablocks are a logical grouping of erase units with each erase unit in a different plane so that reading, writing, and erasing across the metablock occurs simultaneously. EX1009, ¶¶[0114]-[0115], FIG. 7:



58. Adaptive metablocks are also a logical grouping of erase units, but do not have a fixed size. EX1009, ¶¶[0114]-[0115]. Adaptive metablocks might not have one erase unit in each plane (e.g., two erase units in one plane), limiting the extent of simultaneous operation. EX1009, ¶¶[0114]-[0115], [0119]-[0120], FIG. 9A:

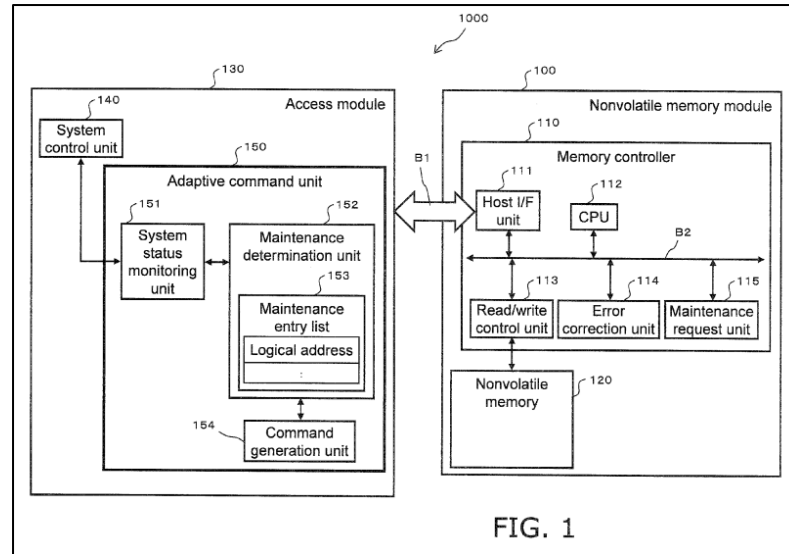


59. [INTENTIONALLY LEFT BLANK]

D. Yamada

60. U.S. Publication No. 2011/0271032 (“Yamada”) was filed on July 27, 2010, and was published on November 3, 2011. I understand that it is prior art under AIA 35 U.S.C. § 102(a)(1)-(2).

61. Yamada discloses a maintenance scheme for managing nonvolatile memory module 100 (*i.e.*, flash memory). Specifically, Yamada describes memory controller 110 including a “maintenance request unit,” and access module 130 (*i.e.*, host) including a maintenance determination unit 152 with command generation unit 154. EX1010, [0012], [0017], [0031].



62. Yamada describes that the host analyzes the metadata of blocks or groups of blocks to determine if certain thresholds have been met. EX1010, ¶¶[0013], [0032], [0080]-[0081]. If so, the controller transmits the metadata and address information to the host in “maintenance request” from the “maintenance request unit.” EX1010, ¶¶[0014], [0109]-[0111]. In response, the host may then issue a “maintenance command” from the “maintenance command unit,” commanding the controller to copy data from the block or blocks that need maintenance to new block(s). EX1010, ¶¶[0014], [0109]-[0111].

63. Moreover, Yamada discloses performing maintenance operations during the lulls between sets of normal storage operations. EX1010, Abstract. The maintenance determination unit, by being on the access module, monitors the access module’s status. EX1010, ¶¶[0018]-[0019], [0022]. When the maintenance determination unit finds downtime in normal processing, the maintenance determination unit performs requested maintenance operations. EX1010, ¶¶[0018]-[0019], [0022].

E. ZBC, SBC-4, and SPC-4

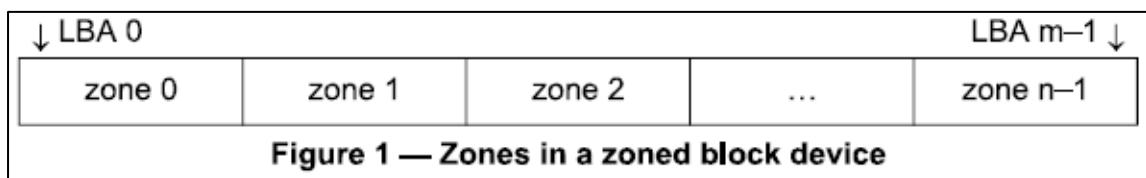
64. ZBC, SBC-4, and SPC-4 published on May 15, 2014, May 20, 2014, and May 17, 2014, respectively. EX1100, ¶¶2-10; EX1103; *see also* EX1109 (depicting T10’s website). I

understand that all three were published by T10 InterNational Committee for Information Technology Standards (“INCITS”), and publicly available through the T10 website. EX1100, ¶¶2-11; EX1101; EX1103; *see also* EX1109. I also understand that all three are prior art under at least 35 U.S.C. § 102(a)(1).

65. I am aware of INCITS, and T10 and T10’s website in particular. Both are well known. T10 publishes SCSI standards and makes publications readily available on T10’s website. *See, e.g.*, EX1101; EX1103. And SCSI protocols and command sets are popular interfaces for storage devices. *See* EX1200; EX1201; EX1202; EX1203.

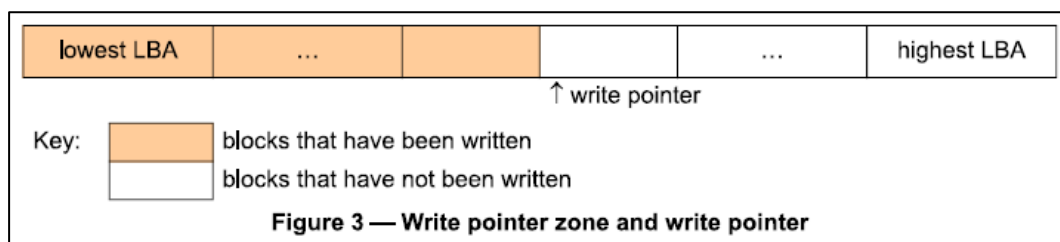
66. ZBC is another SCSI command set detailing an interface for a zoned block device model. EX1006, 1, Figure 0. Moreover, ZBC is an extension command set, thereby also accepting commands from core SCSI specifications such as SBC-4 and SPC-4. EX1006, x, 12-13, 22, Figure 0.

67. According to SPC-4, which is “shared across all device types,” (EX1006, Figure 0.) SCSI logical block devices (*e.g.*, logical units, volumes, and/or partitions) are created for physical storage devices. EX1008, 14, 61. Each logical block device has an independent address range that “shall begin with block zero and be contiguous up to the last block of that logical unit or within that volume or partition.” EX1008, 14. According to ZBC, a “logical unit,” volume, or partition can “be a zoned block device,” (EX1006, 1.) and with zoned block devices, the underlying address range is “divided into ranges called zones The entire capacity of a zoned block device is organized into a set of contiguous, non-overlapping zones.” EX1006, 13, Figure 1:



68. ZBC includes at least two different kinds of zoned block devices—(1) “host aware

zoned block device[s]” and (2) “host managed zoned block device”—and at least two different kinds of zones—(1) “sequential write preferred zone” and (2) “sequential write required zone.” EX1006, 4, 13-14, Table 14. Sequential write preferred zones and sequential write required zones include metadata such as the state of the zone and write pointers. The state of a zone is the zone condition, and includes “EMPTY,” “OPEN,” “READ ONLY,” “FULL,” and “OFFLINE.” EX1006, Table 15. Write pointers generally track the next available logical block address zones have write pointers. EX1006, 14, 16-17, Figure 3:



69. With each write command, the write pointer increments its logical block address until the end of the zone is reached or a reset write pointer command is applied. EX1006, Table 15. A reset write pointer command resets the write pointer back to the lowest logical block address in the zone and also resets the data in the zone to the manufacturer’s default initialization pattern. EX1006, 3, 15-18.

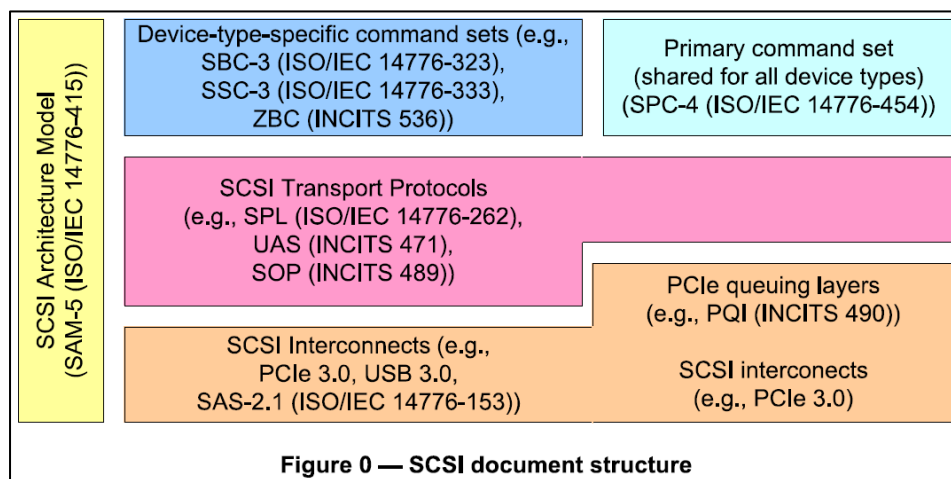
VII. INVALIDITY OF CLAIMS

A. GROUND I: CLAIMS 1-12, 16, 20-23 ARE RENDERED OBVIOUS BY REITER IN LIGHT OF ZBC, SBC-4, AND SPC-4

1. Motivation To Combine Reiter, ZBC, SBC-4, And SPC-4

70. A POSITA would be motivated to combine the teachings of ZBC, SBC-4, and SPC-4 (collectively, “ZBC”) with flash system 100 disclosed in Reiter. Specifically, a POSITA would be motivated to use ZBC, SBC-4, and SPC-4 as the standard governing requests sent to Reiter’s flash system 100.

71. **First**, to use ZBC, SBC-4, and SPC-4 as the governing standard, a POSITA would be motivated to combine ZBC, SBC-4, and SPC-4. This combination makes sense. ZBC, SBC-4, and SPC-4 are SCSI specifications intended to be used with one another. *See* EX1006, x, 1, 12-13, 22, Figure 0:

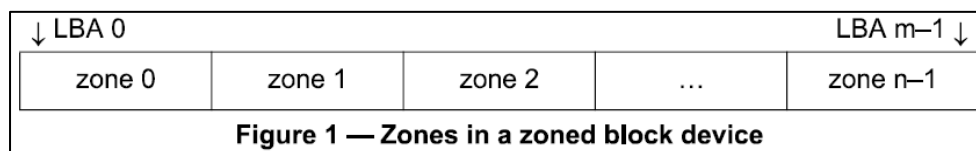


The “purpose of [ZBC] is to define the model and command set extensions to be used in conjunction with ... SPC-4” EX1006, x, 1. ZBC expressly discloses that it is “implemented in conjunction with ... SPC-4.” EX1006, 1. And SPC-4 is listed as “shared for all device types.” EX1006, Figure 0. ZBC expressly relies not only on SPC-4, but also SBC-4, throughout the specification. EX1006, x, 1, 12-13, 22, Figure 0.

72. **Second**, with a combined ZBC, SBC-4, and SPC-4, a POSITA would be motivated to apply that standard to govern Reiter’s flash system 100. Reiter does not disclose using, or otherwise requiring, a particular command set, but generally discloses support for the widely used SCSI protocols. EX1004, 7:9-18. Reiter states that “[c]ommunication link 102 might be a custom-designed communication link, or might conform to a standard communication protocol such as, for example ... SCSI ... protocol bus, ... Serial Attached SCSI ... protocol bus, ... or any other similar interface link for connecting a peripheral device to a computer.” EX1004, 7:9-18. Reiter also discloses “receiv[ing] ... requests ... such as read or write operations from ... external

devices,” where those “requests ... generally include a[] ... logical block address where data should be read or written.” EX1004, 8:1-6. Thus, given the ZBC being a SCSI extension and being able to fill Reiter’s lack of a detailed interface, a POSITA would be motivated to support commands such as those disclosed in ZBC to enable specific read and write operations based on Reiter’s memory structure.

73. Moreover, a POSITA would be particularly motivated to implement Reiter to run on ZBC commands. SCSI protocols and commands such as those disclosed in ZBC are widely used, and would thus improve the compatibility and interoperability of Reiter’s flash system. And ZBC’s zoned block device model (EX1006, 1.) is well suited for Reiter’s flash system because it would enable effective utilization of Reiter’s superblock memory organization. *See* EX1004, FIG. 4c. ZBC manages an interface for a “zoned block device.” EX1006, 1. Zoned block device have an address space comprising logical block addresses (“LBAs”) “divided into ranges called zones.” EX1006, 13, 29, Table 16, Figure 1:



74. Zoned block devices’ zones, as governed by ZBC, are analogous to Reiter’s superblocks. Like zones, superblocks comprise “a logical collection of blocks representing a fixed range of LBAs.” EX1004, 9:11-13. ZBC’s zones and Reiter’s superblocks include metadata indicating the next available memory location within the zone and superblock, respectively. EX1006, 14, 16-17, Figure 3 (illustrating a zone with a write pointer indicating next available LBA); EX1004, 9:52-10:1, FIG. 4b (illustrating superblock summary page with “free page pointer” indicating next available memory location). And like ZBC, which use LBAs to access different zones, Reiter’s superblocks are addressed through the received LBA. EX1004, 8:54-

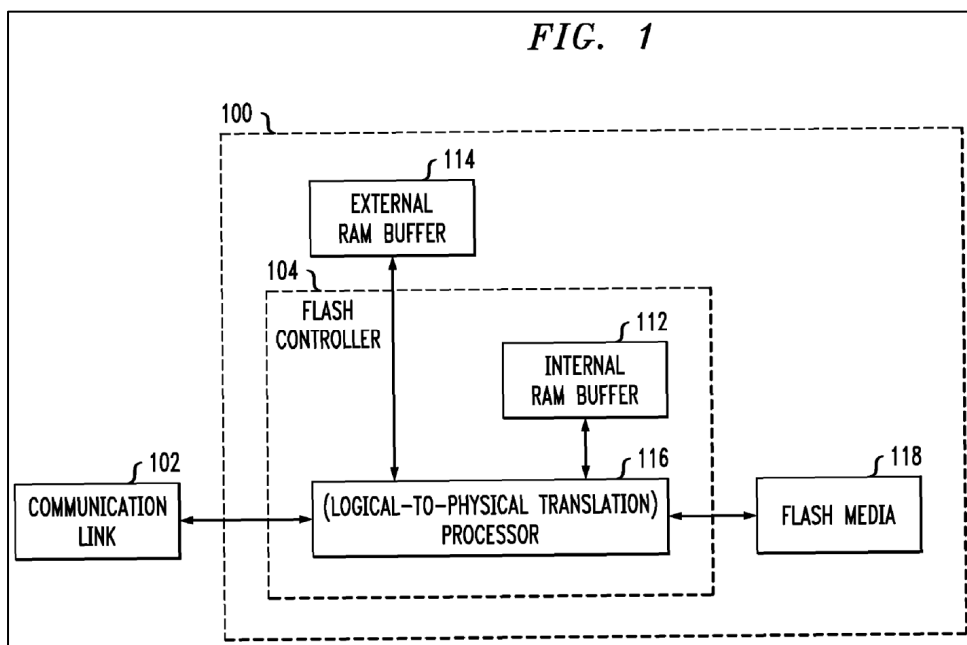
9:25, FIG. 4a (deriving superblock number 410 from logical block address 402); EX1006, 13.

75. A POSITA would have a reasonable expectation of success in adding support for ZBC in Reiter's flash system 100. SCSI commands and protocols such as ZBC were widely known and would have been familiar to a POSITA. SCSI specifications were intended to be used across a wide range of storage devices, including flash memory. A POSITA would have also understood that Reiter's superblock-based memory organization is suitable for zone-based commands like ZBC. Further, a POSITA would have understood that ZBC's zoned block device model is a logical construct that can be layered on top of any storage device.

76. Moreover, Reiter and ZBC disclose analogous art. Both relate to storage devices, including accessing blocks of memory.

2. Claim 1, Limitation 1[pre]: *A storage device comprising:*

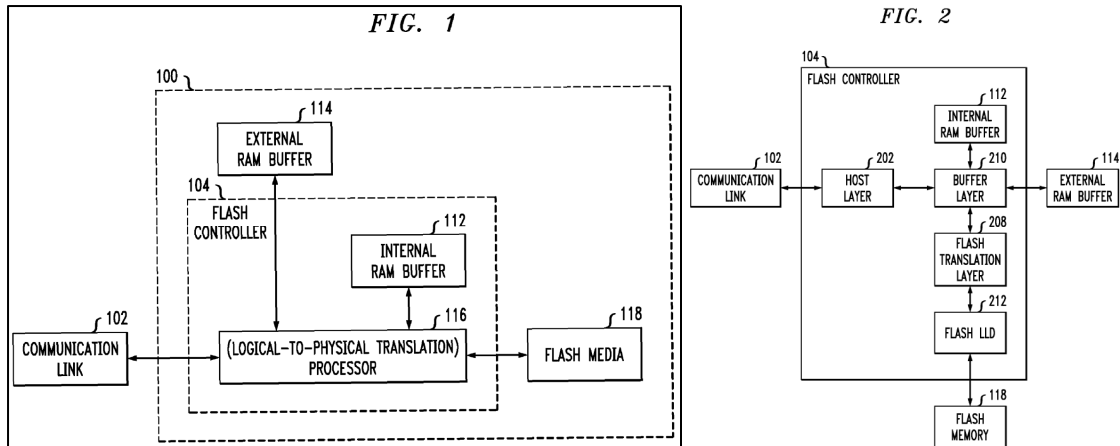
77. To the extent the preamble is limiting, Reiter discloses it. Specifically, Reiter discloses a "flash ... system 100" (*i.e.*, ***storage device***) "implementing a logical-to-physical translation" scheme. EX1004, 6:63-66, FIG. 1:



78. It is my opinion that Reiter renders this limitation obvious.

(a) **Limitation 1[a]: *flash memory having physical erase units; and***

79. Reiter's flash system 100 includes "flash media 118" or "flash memory 118" (*i.e., flash memory*). EX1004, 6:63-7:5, 7:56-61, FIGS. 1, 2:

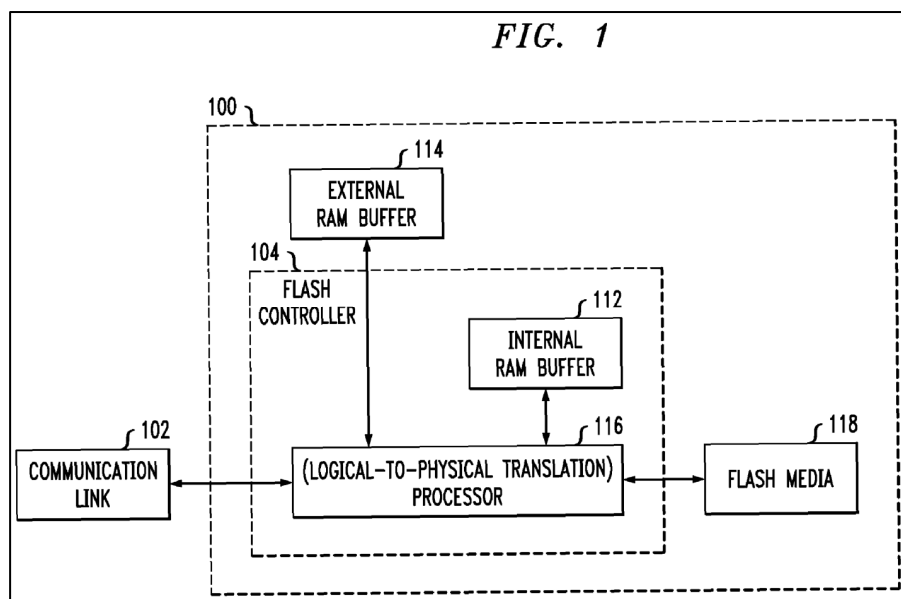


80. Reiter discloses that a "commonly employed type of flash memory technology is NAND flash memory." EX1004, 1:40-45. "There are three basic operations for NAND devices: read, write, and erase." EX1004, 1:51-52. Read and write operations are performed on "[p]ages," which "are typically arranged in blocks." EX1004, 1:52-57. "[A]n erase operation is performed on a block by block basis." EX1004, 1:56-57. Reiter also expressly discloses that its flash media comprises "blocks" that may be erased. EX1004, 6:60-62, 17:61, cls. 2, 9. Reiter thus discloses that its flash memory is made up of independently erasable blocks (*i.e., flash memory having physical erase units*).

81. It is my opinion that Reiter renders this limitation obvious.

(b) **Limitation 1[b]: *circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:***

82. Reiter's flash system 100 includes a "flash controller 104" (*i.e., circuitry*). EX1004, 6:63-7:2, FIG. 1:



83. Reiter’s “[F]lash controller 104 *receives* one or more requests for flash media access, such as read or write operations [(i.e., *incoming data access requests*)] from one or more external devices via communication link 102.” EX1004, 8:1-4. The “requests for access to flash media 118 generally include at least one logical block address (LBA) where data should be read or written” (i.e., *incoming data access requests, accompanying address information*). EX1004, 8:4-6.

84. For example, Reiter discloses a “flash media read operation 530” that is “*executed* by FTL 208” in flash controller 104. EX1004, 11:55-57, FIG. 2. The “flash media read operation” is executed “in response to a flash media read request received from an external device,” which “includes a logical block address (LBA).” EX1004, 11:55-61.

85. A POSITA would be motivated to implement Reiter’s flash controller 104 to accept and execute ZBC commands. Section VII.A.1. ZBC includes a read command—a request disclosed in Reiter (EX1004, 8:1-6.)—(i.e., *incoming data access requests*). EX1007, Tables 59, 61-63; *see also* EX1006, 22 (“The commands for host aware zoned block devices are defined in SBC-4.”). A ZBC read command includes a logical block address (i.e., *for each of the incoming*

data access requests, accompanying address information)—also as disclosed with Reiter’s requests (EX1004, 8:1-6, 11:55-63, 13:3-46.). EX1007, Table 63:

Table 63 — READ (32) command									
Byte	Bit	7	6	5	4	3	2	1	0
0		OPERATION CODE (7Fh)							
1		CONTROL							
2		Reserved							
...									
5									
6		Reserved			GROUP NUMBER				
7		ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (0009h)							
9									(LSB)
10		RDPROTECT			DPO	FUA	RARC	Obsolete	Reserved
11		Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS							
...									
19									(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG							
...									(LSB)
23		EXPECTED LOGICAL BLOCK APPLICATION TAG							
24	(MSB)								(LSB)
25		LOGICAL BLOCK APPLICATION TAG MASK							
26	(MSB)								(LSB)
27		TRANSFER LENGTH							
28	(MSB)								
...									
31									(LSB)

86. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) **Limitation 1[b][i]:** *derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,*

87. From a ZBC read command (*i.e., accompanying incoming address information*), flash controller 104 *derives* (1) a number for a logical unit, volume, or partition (hereinafter “logical units”) (*i.e., first address portion*), (2) superblock number 410 (*i.e., second address portion*), and (3) logical sector number 404 (*i.e., third address portion*), or alternatively a bitstring of logical block number 406 and logical page number 408 (*i.e., third address portion*), or

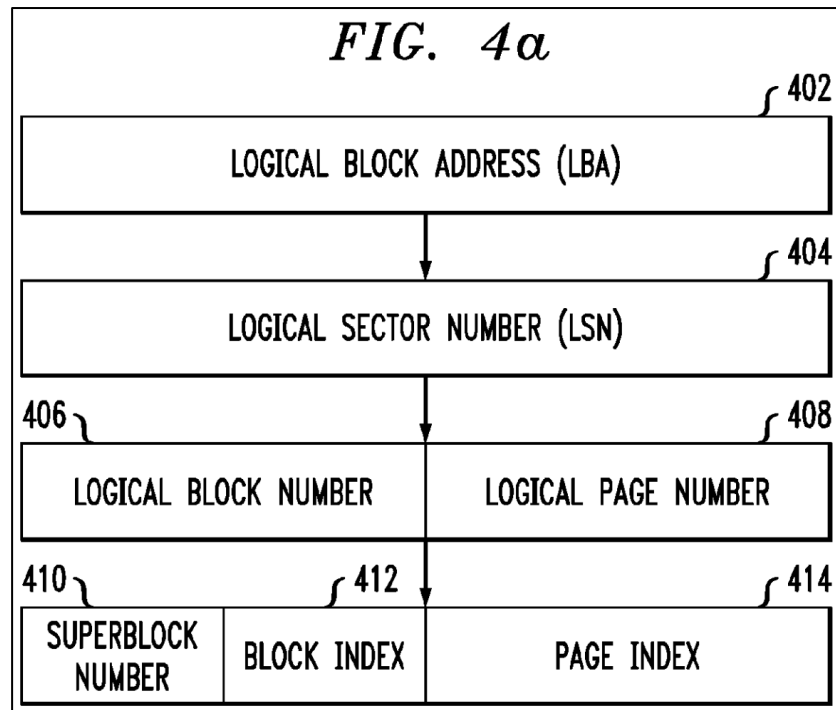
alternatively logical page number 408 (*third address portion*) or page index 414 (*third address portion*).

88. **First Address Portion:** A POSITA would be motivated to incorporate a logical unit number in ZBC commands, and motivated to modify flash controller 104 to decode (*i.e., derive*) that number (*i.e., first address portion*) from the command (*i.e., from the accompanying address information*). ZBC read commands target a “logical unit,” “volume,” or “partition.” See EX1008, 14, 61; *see also* EX1006, x, 1, 4. An “objective of ... [ZBC]” is “to permit an application client to communicate with a logical unit that declares itself to be a zoned block device” EX1006, 1. But logical units are logical structures, where multiple can exist on the same physical storage device. See EX1008, 14 (describing a logical unit as an “externally addressable entity **within a SCSI target device** that implements a SCSI device model”), 462 (“request[ing] ... the peripheral device[’s] logical unit inventory”—a list of logical unit numbers). And each logical unit “shall begin with block zero and be contiguous up to the last logical block of that logical unit” EX1008, 61. With multiple logical units, and overlapping LBA spaces, an LBA by itself is not enough to identify a location on a physical storage device. A POSITA would look to incorporate its commands to distinguish the same LBA to two different logical units.

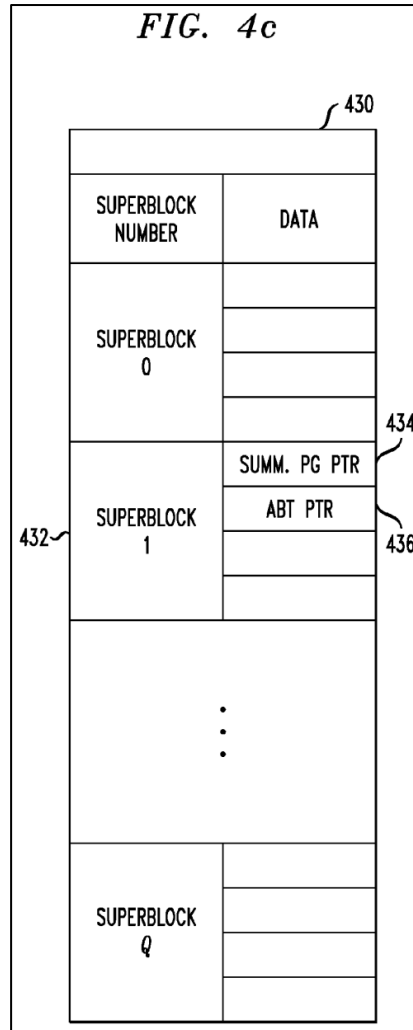
89. A POSITA would be motivated to incorporate a separate address found in SPC-4 into ZBC commands. Section VII.A.1. SPC-4 recites that a “logical unit” is “[a]n **externally addressable** entity within a SCSI target device” EX1008, 14. Addressing a logical unit is done with a “logical unit number”—“[a]n encoded 64 bit identifier for a logical unit.” EX1008, 14; *see also* EX1006, 4. ZBC already mentions logical unit number. EX1006, 4. A POSITA would be motivated to more fully incorporate that address in its commands via the teachings in SPC-4.

90. **Second Address Portion:** A POSITA would be motivated to use Reiter’s flash

controller 104 to *derive* superblock number 410 (*i.e.*, *second address portion*) from the LBA in the ZBC read commands (*i.e.*, *from the accompanying address information*). A ZBC read command includes an LBA. EX1007, Table 63. Reiter discloses that read and write “request[s] ... include[] a[n] ... LBA” EX1004, 8:4-6, 8:62-67. “FIG. 4a shows an exemplary relation of a[n] ... LBA ... to a physical address ... as managed by FTL” EX1004, 8:56-59, FIG. 4a:



Translating an LBA results in a bitstring made up of superblock number 410, block index 412, and page index 414. EX1004, 8:66-9:25, FIG. 4a. Superblock number 410 is identified in the page global directory (PGD) 430 to obtain a “pointer 434 ... to the summary page associated with [the corresponding] Superblock.” See EX1004, 10:13-19, FIG. 4c:



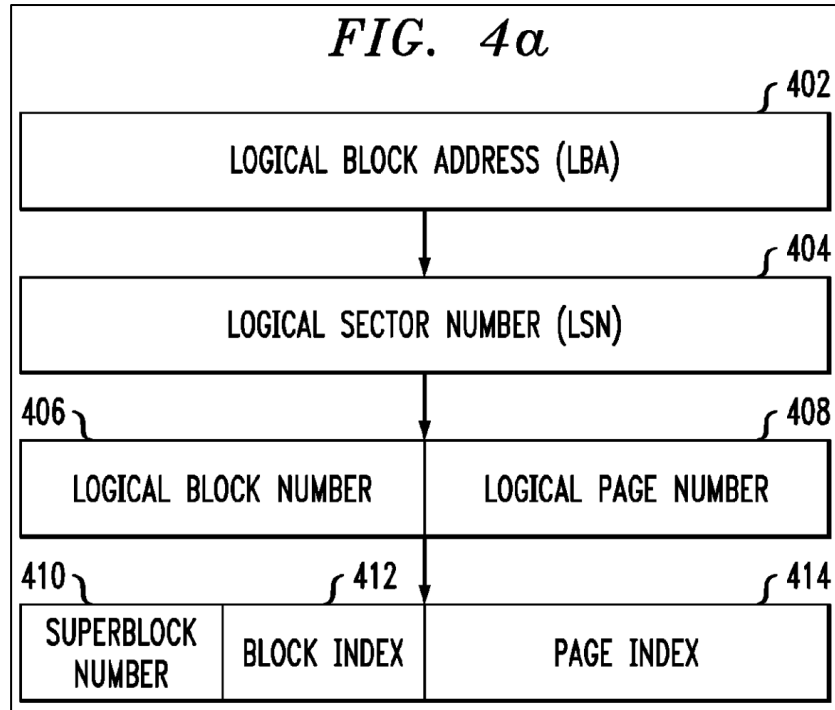
91. A POSITA would expect the ZBC LBA to be translated by Reiter’s flash controller 104.

92. A POSITA would understand the (1) LBA 402, (2) LSN 404, (3) logical block number 406 and logical page number 408, and (4) superblock number 410, block index 412, and page index 414 in FIG. 4a to be bitstrings. Each is depicted as one unit. EX1004, FIG. 4a (above). This matches the single unit depiction of logical block address 402 and logical sector number 404 bitstrings in the prior translation steps. EX1004, FIG. 4a (above). Other prior art references depicting bits as one block with separate portions of the block associated with different values (EX1004, FIG. 4a (above for superblock number 410, block index 412, and page index 414))

describe that block as a bitstring, and the values as bitfields within the bitstring. *See* EX1005, 5:7-10 (“The logical unit number is the high order binary digits of the logical address and may be derived from the logical address by a bit shift operation.”), FIG. 4; EX1210, ¶[0157], FIG. 33.

93. A POSITA would also be motivated to store or keep the bitfields of each translation level together in bitstrings (as depicted in FIG. 4a). This creates a simple system where one translation step is only started after completing the prior completion step (*see* EX1004, FIG. 4a), because the resulting bitfields need to be produced and then combined into one bitstring prior to the next translation step. Ensuring lock-step translation of addresses simplifies management where (1) bitfields are stored all over the place and gathered from different parts of memory and (2) some bitfields are further along translation than other bitfields associated with the same address. This simplicity becomes increasingly important with concurrent and/or out-of-order execution of read/write/erase commands.

94. **Third Address Portion:** A POSITA would be motivated to use Reiter’s flash controller 104 to *derive* logical sector number 404 (*i.e.*, ***third address portion***), or alternatively bitstring of logical block number 406 and logical page number 408 (*i.e.*, ***third address portion***), or alternatively logical page number 408 or page index 414 (*i.e.*, ***third address portion***), from the LBA in the ZBC read commands (*i.e.*, ***from the accompanying address information***). A ZBC read command include an LBA. EX1007, Table 63. Reiter discloses that read and write “request[s] ... include[] a ... LBA” EX1004, 8:4-6, 8:62-67. “FIG. 4a shows an exemplary relation of a[n] ... LBA ... to a physical address ... as managed by FTL” EX1004, 8:56-59, FIG. 4a:



95. “LBA 402 corresponds to at least one logical sector number (LSN) 404.” EX1004, 8:66-67. Logical sector number 404 is translated to a bitstring with logical block number 406 and logical page number 408 bitfields. EX1004, FIG. 4a. Logical block number 406 is obtained by “dividing LSN 404 by a number of sectors per block,” and logical page number 408 is obtained by “dividing LSN 404 by a number of sectors per page ...” EX1004, 9:5-9. Logical page number 408 is “divid[ed] ... by a number of pages per block” to obtain page index 414. EX1004, 9:17-20. A POSITA would expect the ZBC LBA to be translated by Reiter’s flash controller 104.

96. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(d) Limitation 1[b][ii]: *identify an addressed block device from the first address portion,*

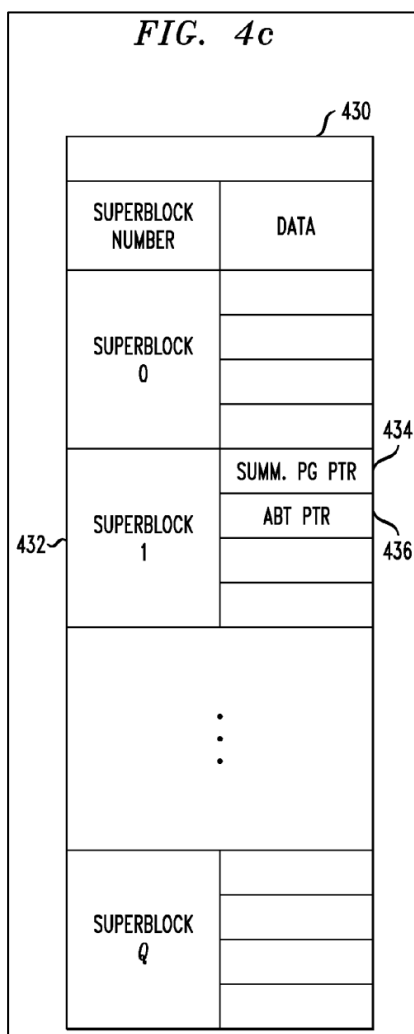
97. The number associated with a logical unit (*i.e., from the first address portion*) (Section VII.A.2.c.) *identif[ies]* the targeted logical unit (*i.e., addressed block device*). A logical unit has an independent range of logical block addresses starting at zero. EX1008, 61; *see also*

EX1001, 7:54-63, FIG. 5.

98. It is my opinion that ZBC, SBC-4, and SPC-4 render this limitation obvious.

(e) **Limitation 1[b][iii]:** *identify an addressed segment within the addressed block device from the second address portion, and*

99. Superblock number 410 (*i.e., from the second address portion*) *identif[ies]* the superblock, which a POSITA would be motivated to implement as a ZBC zone (*addressed segment*). Section VIII.C; *see also* EX1011, 22 (“Samsung’s Accused Instrumentalities ... identify an addressed zone (segment).”). For instance, superblock number 410 indexes page global directory (PGD) 430 to obtain the “summary page associated with [the corresponding] Superblock.” EX1004, 10:13-19, FIG. 4c:



100. A POSITA would be motivated to use the superblocks disclosed in Reiter as zones for ZBC commands (*i.e.*, ***addressed segment***). Section VII.A.1.

101. A ZBC zone constitutes ***an addressed segment*** because a zone “constitutes a fundamental unit of storage allocation operated upon by the host file system.” EX1001, 9:10-20. Under ZBC, “the zoned block device models described in this standard [include] (a) **host aware** zoned block device ... and (b) **host managed** zoned block device.” EX1006, 12. In other words, the host is aware of, and operates on, zones. As an example of host operations on zones, ZBC includes a “RESET WRITE POINTER command.” EX1006, 29, Table 16:

Table 16 — RESET WRITE POINTER command								
Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Fh)							
1	Reserved			SERVICE ACTION (14h)				
3	(MSB)							
...	ZONE ID							
9	(LSB)							
10								
...	Reserved							
13								
14	Reserved							RESET ALL
15	CONTROL							

102. “A reset write pointer operation sets the write pointer in a write pointer zone to the lowest LBA of th[e] ... zone,” where the zone is identified by the “ZONE ID” parameter in the command EX1006, 15, Table 16. The reset write pointer operation also resets the data in the zone to its initialization values. EX1006, 16-17. The host operates on zones by sending commands such as the reset write pointer command.

103. Superblock number 410 is derived from LBA 402, (EX1004, FIG. 4a.) which is from a ZBC command’s LBA (Section VIII.D.1.d.). Because a ZBC LBA is within the LBA range specified by the targeted logical unit, (*see* Sections VIII.D.1.d-e; EX1006, x, 1, 4; EX1008, 61.) superblock number 410 identifies a superblock ***within*** the targeted logical unit (*i.e.*, ***addressed***

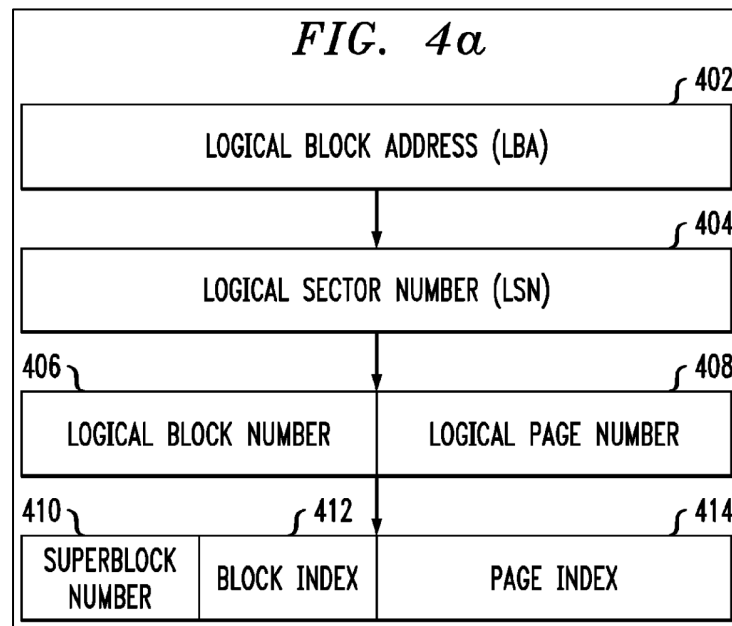
block device).

104. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(f) **Limitation 1[b][iv]:** *identify an addressed storage location within the addressed segment from the third address portion,*

105. Flash controller 104 translates logical sector number 404 (*i.e., from the third address portion*), and alternatively the bitstring of logical block number 406 and logical page number 408 (*i.e., from the third address portion*), and alternatively logical page number 408 or page index 414 (*i.e., from the third address portion*) to block index 412 and/or page index 414.

EX1004, 9:5-25, FIG. 4a:

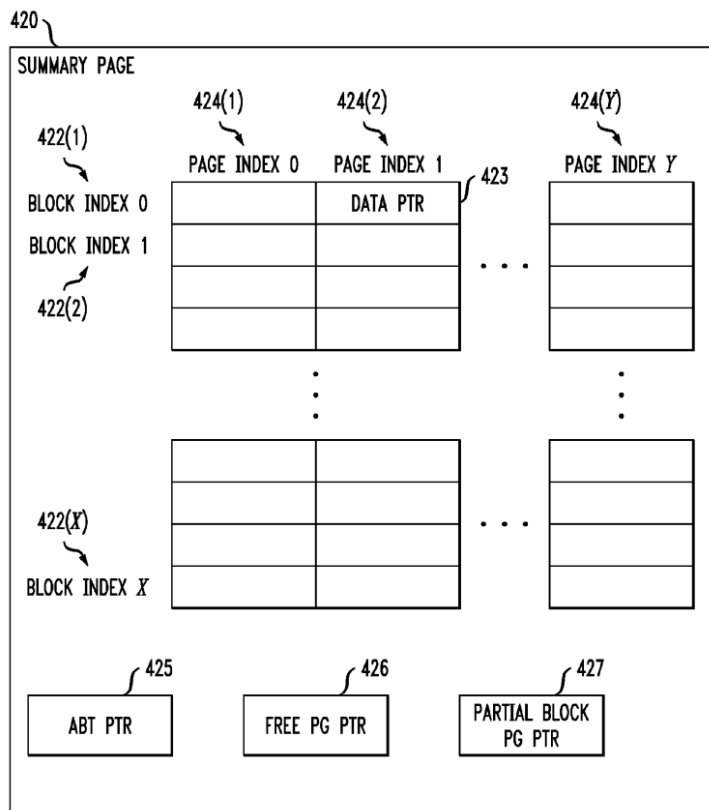


These indexes *identify* a logical block and/or logical page (*addressed storage location*) *within* the superblock (*i.e., zone (addressed segment)* (Section VIII.D.1.f.)) addressed by superblock number 410. EX1004, 9:52-63.

106. Specifically, block index 412 and page index 414 select a block within the superblock and page within that block, respectively, from a superblock's summary page. EX1004,

9:52-63, FIG. 4b. The “summary page ... contains mapping data for the ... Superblock.” EX1004, 9:52-54. “For example, summary page 420 contains the block index and page index ... for all X blocks and Y pages in [the] Superblock” EX1004, 9:54-58, FIG. 4b:

FIG. 4b



107. Because logical sector number 404, and alternatively bitstring of logical block number 404 and logical page number 406, and alternatively logical page number 406 and page index 414, are translated into block index 412 and/or page index 414, Reiter discloses *identify[ing] an addressed storage location ... from the third address portion.*

108. Further, Patent Owner agrees with this mapping as its infringement contentions alleges that “identify[ing] an addressed storage location[] ... includ[es] a logical block, page, and/or erase unit.” EX1011, 27.

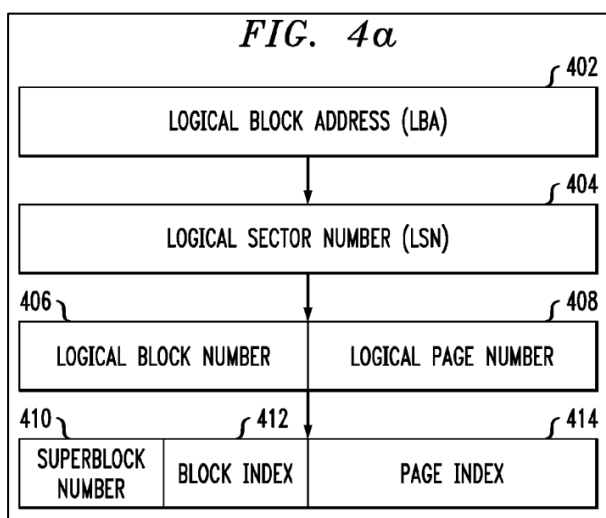
109. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4

renders this limitation obvious.

- (g) **Limitation 1[c]:** *wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,*

110. Reiter discloses flash controller 104 (*i.e.*, **circuitry**) performing integer division on logical sector number 404 to *identify* logical block number 406 (*i.e.*, **logical erase unit**). Alternatively, a POSITA would be motivated to apply integer division to subdivide out (*i.e.*, *identify*) logical block number 406 bitfield (*i.e.*, **logical erase unit**) from the bitstring of logical block number 406 and logical page number 408. Alternatively, a POSITA would be motivated to apply integer division to subdivide out (*identify*) block index 412 bitfield (**logical erase unit**) from the bitstring of superblock number 410, block index 412, and page index 414.

111. **Integer Division On Logical Sector Number 404:** Reiter discloses that “[e]ach LBA can be divided into a logical block number and logical page number” EX1004, 3:48-50, 9:5-20. The first translation step is translation of LBA 402 to logical sector number 404. EX1004, 8:66-67, FIG. 4a:



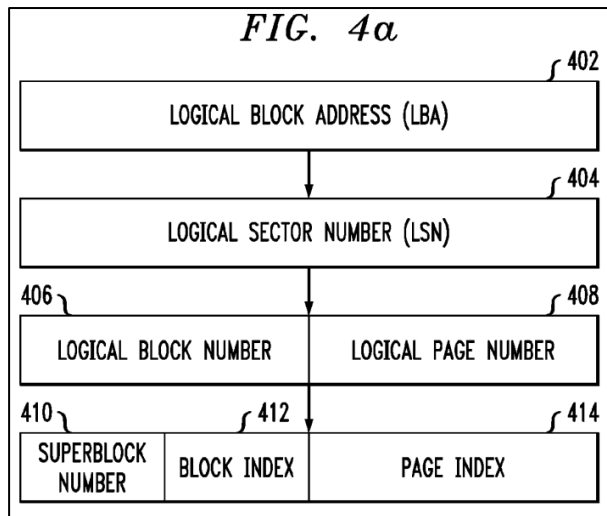
112. In the second step, “LSN 404 corresponds to ... logical block number (LBN) 406 and ... logical page number (LPN) 408.” EX1004, 9:5-6, FIG. 4a (above). “FTL 208 derives LBN

406 by **dividing** LSN 404 by a number of sectors per block of flash media 118.” EX1004, 9:6-8.

This integer division constitutes the *division operation to identify a logical erase unit*.

113. **Alternative, Integer Division To Subdivide Out Logical Block Number 406**

Bitfield: Alternatively, after the second step, a new bit string is formed including logical block number 406 and logical page number 408. EX1004, FIG. 4a (below).

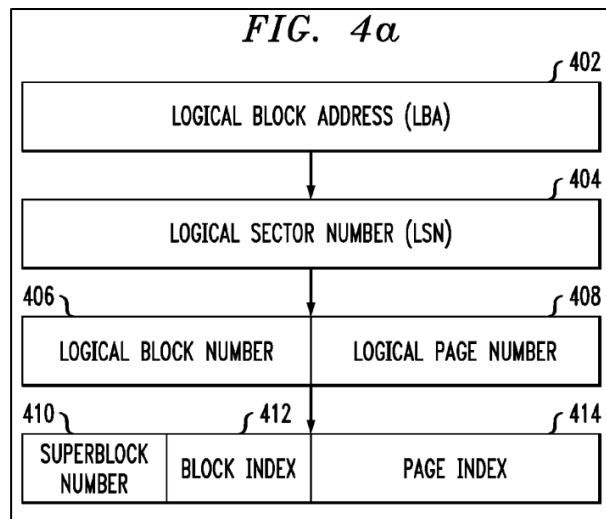


114. A POSITA would be motivated to divide by a certain power-of-two to split the bitstring into a first and second bitfields. Integer division by a certain power-of-two extracts bitfields out of bitstrings. The quotient is the extracted most significant bitfields (*i.e.*, bits on the lefthand portion of a bitstring). Integer division is preferable because it is fast, as division by a power-of-two on a bitstring is implemented with a simple bit shift operation. Logical block number 406 is the higher order or most significant bitfield of the bitstring. EX1004, FIG. 4a. Applying integer division on the bitstring causes superblock number 410 to be extracted through the quotient. Because superblock number 410 is a bitfield, a POSITA would be motivated to implement integer division via a bit shift operation to perform the extraction quickly.

115. Further, a POSITA would also be motivated to extract out logical block number 406. Reiter discloses later translation steps that are only performed on logical block number 406

and logical page number 408. EX1004, 9:13-20. For instance, “FTL 208 derives Superblock number 410 and block index 412 from LBN 406 by dividing LBN 406 by a number of blocks per Superblock” EX1004, 9:13-17. Similarly, “Page index 414 is derived from LPN 408 by dividing LPN 408 by a number of pages per block” EX1004, 9:17-20. Dividing to subdivide the bit string into bitfields alternatively constitutes the division operation to identify logical block number 406 (logical erase unit).

116. **Alternative, Integer Division To Subdivide Out Block Index 412 Bitfield:** Alternatively, after translating logical block number 406 and logical page number 408, a new bit string is formed including superblock number 410, block index 412, and page index 414. EX1004, FIG. 4a:



117. Similar to the prior theory, a POSITA would be motivated for the sake of speed to divide by a power-of-two to split the bitstring into the corresponding bitfields.

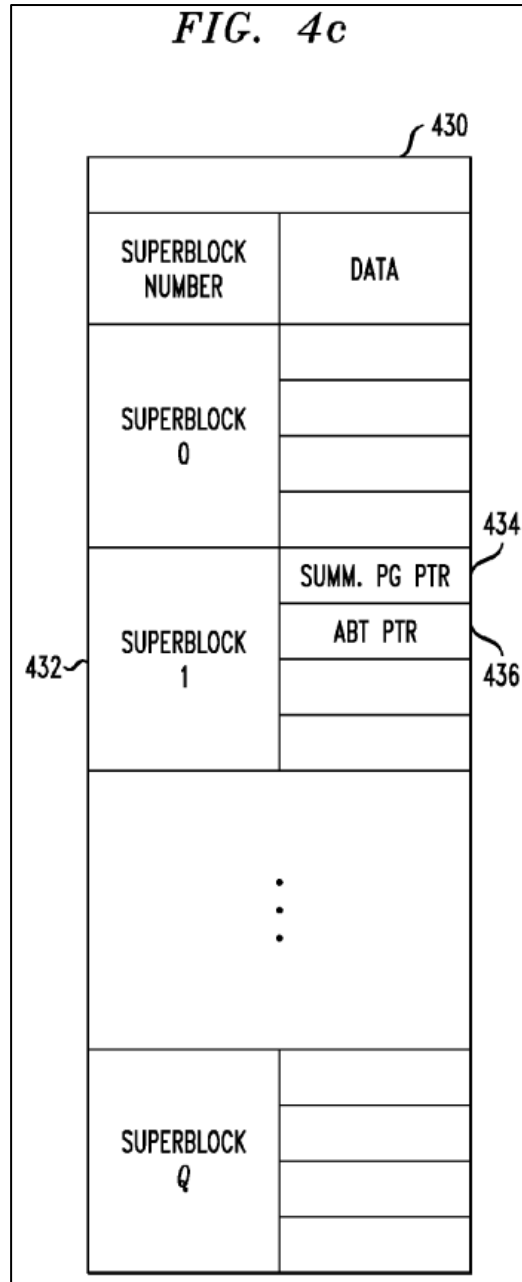
118. **Logical Erase Unit:** The resulting logical block number 406, or alternatively block index 412, is a *logical erase unit*. This is explicit in the name—“**logical block** number” and “**block index**.” **Logical** block number 406 and block **index** 412 are logical units. Block refers to an erase unit, as “an erase operation is performed on a block by block basis.” EX1004, 1:56-57.

119. Additionally, logical block number 406 and block index 412 are abstractions of a physical block (*i.e.*, erase unit) because logical block number 406 and block index 412 map to a physical block. “LBN 406 ... corresponds to Superblock number 410 and block index 412.” EX1004, 9:5-6. “[A] Superblock generally is a logical collection of blocks” EX1004, 9:11-14. And the block index 412 indexes the superblock’s list, identifying a logical block within the collection and, with page index 414, selecting “data pointer 423, which points to the physical address.” See EX1004, 9:13-17, 9:52-58, FIGs. 4b-c.

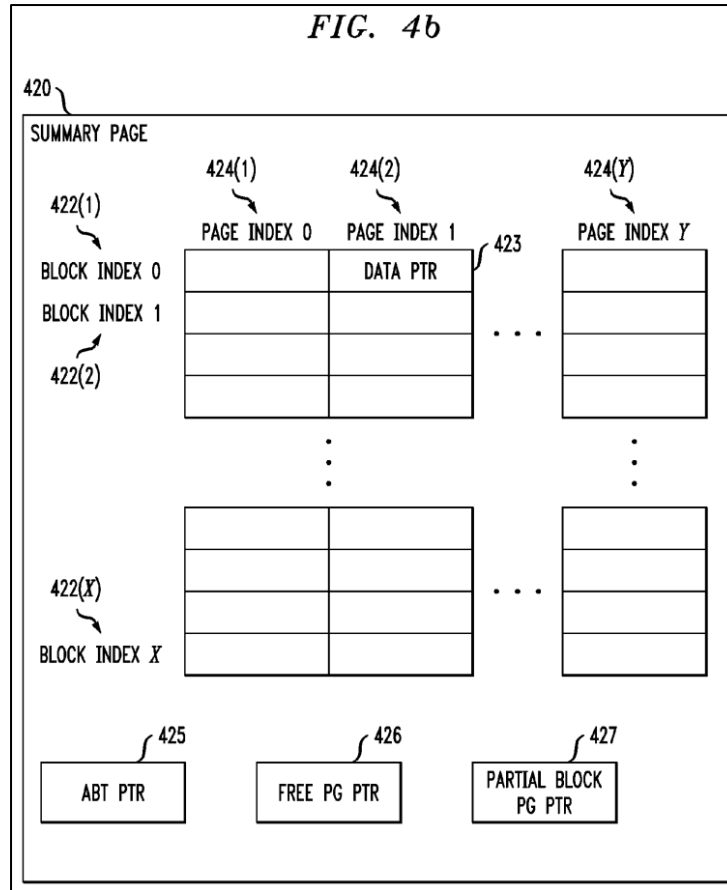
120. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(h) **Limitation 1[d]: *wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;***

121. Reiter discloses flash controller 104 (*i.e.*, ***circuitry***) translates logical block number 406 (*i.e.*, ***identified logical erase unit***), and alternatively block index 412 (*i.e.*, ***identified logical erase unit***), (Section VII.A.2.g.) to a physical block number (*i.e.*, ***identify one of the physical erase units that corresponds to the identified logical erase unit***). EX1004, 3:41-55. “LBN 406 ... corresponds to Superblock number 410 and block index 412.” EX1004, 9:5-6, FIG. 4a. Superblock number 410 selects the summary page “contain[ing] ... mapping data for the associated Superblock.” EX1004, 9:52-54, FIG. 4c:



“[A] Superblock generally is a logical collection of blocks,” where the “summary page 420 contains ... block indices 422(1)-422(X) and page indices 424(1)-424(Y) for all X blocks and all Y pages in [the s]uperblock.” EX1004, 9:54-58, FIG. 4b. Each pair of “block indices 422(1)-424(Y) and page indices 424(1)-424(Y) are stored as data pointer 423, which points to the physical address.” EX1004, 9:58-63, FIG. 4b:



Block index 412 with page index 414 selects one of those data pointers 423. EX1004, 9:11-17, 9:52-63, FIG. 4b. Data pointer 423 points to the underlying, selected physical block. X1004, 9:52-63, 10:1-3, FIG. 4b; *see also* EX1004, 2:59-62, 3:41-55 (listing various translation schemes translating an LBA to physical block number and page number, such as a “superblock mapping scheme,” which includes translating an “LBA” to a “logical block number and ... page number,” turning logical block number into a “superblock number and a PGD index offset,” turning logical page number into “a PMD index offset and PT index offset,” and translating superblock number, PGD index offset, PMD index offset, and PT index offset into a “physical block number and physical page number”). Because the underlying physical block corresponds to superblock index 412 and block index 414, which were derived from logical block number 406, flash controller’s

104 translation *identifies one of the physical erase units that corresponds to* logical block number 406, or alternatively block index 414 (*identified logical erase unit*).

122. It is my opinion that Reiter renders this limitation obvious.

- (i) **Limitation 1[e]:** *wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests.*

123. Reiter's flash storage system 100 (*i.e., storage device*) *detects* bad blocks (*i.e., failure condition of the physical erase units*). Upon detecting a bad block, a POSITA would be motivated to *remap* logical block number 406 (*i.e., logical erase unit*), and block index 412 (*i.e., logical erase unit*), from the bad block (*i.e., one of the physical erase units for which the failure condition is detected*) to instead a working block (*i.e., one of the physical erase units for which the failure condition has not been detected*).

124. **Detect A Failure Condition Of The Physical Erase Unit:** Reiter discloses "maintain[ing] a record of bad blocks" by scanning for bad blocks after powering on flash system 100. EX1004, 4:27-37, 17:13-15, 17:47-63. For example, "[o]n startup of flash ... system 100, mapping data ... requires reconstruction." EX1004, 17:13-15. This includes "scan[ning] and group[ing] ... blocks" into various "categories," including "bad blocks." EX1004, 17:47-52. Bad blocks are detected if (1) FTL 208 is "unsuccessful" "read[ing] metadata that is stored in the ... first page of the block" and (2) the following "erase operation of the [same] block was unsuccessful, the block cannot be erased and has failed." EX1004, 17:59-67. "[T]he block address is [then] added to the bad block list." EX1004, 18:4-8.

125. **Remap Each Logical Erase Unit Which Corresponds To One Of The Physical Erase Units For Which The Failure Condition Is Detected:**

126. When a block is added to the bad block list, a POSITA would be motivated to replace the bad block with a working block—remapping logical block number 406, and block index 412, to a new block. Data read from a bad block cannot be trusted, especially given the unsuccessful or erroneous read of the block during the scan at startup and subsequent failed erasure of the block. EX1004, 4:27-37, 17:13-15, 17:47-63. Keeping a bad block around risks reading erroneous data. As a first remapping step, Reiter discloses, upon detecting a bad block, adding the “block address ... to the bad block list ... 428.” EX1004, 18:4-8.

127. A POSITA would be motivated to use Reiter’s RAID functionalities to reconstruct the lost data, and change data pointer 423 from the old bad block to a new working block with the reconstructed data. Reiter discloses how flash system 100 “might be configured such that data is ‘striped across two or more [flash] dies analogously to hard drives in a redundant array of independent disks (RAID)” EX1004, 8:46-50. RAID allows data lost on one disk to be reconstructed with data from the other redundant disks, thereby reconstructing lost data. *See, e.g.*, EX1204, ¶¶[0007]-[0009]; EX1205, ¶¶[0003]-[0006]; EX1206, 97-99; EX1207, 1:22-50; EX1208, 110-15; EX1209, 164-65, 168, 170. Because the previous block is an unreliable bad block, a POSITA would understand that a new block is needed for the reconstructed data. And a POSITA would be motivated to change data pointers 423 in summary page 420 pointing to the old bad block to the new block with the reconstructed data. This allows the LBA addressing, including logical block number 406 and block index 412, for the data to be unchanged despite the intervening data loss. Data at logical block number 406 can be trusted and used for later reads (*i.e., utilize the remapped correspondence to service one or more of the incoming data access requests*). *See* EX1004, 17:13-21.

128. A POSITA would also be motivated to implement this failure detection and block replacement scheme. Reiter expressly acknowledges that “NAND devices can [only] be erased

relatively few times before device failure Therefore, over the operational life of an SSD, blocks of flash memory will fail and become unusable.” EX1004, 4:27-37. Moreover, the reduced life expectancy of SSDs, detection of bad blocks, and remapping bad blocks is well known in the prior art.

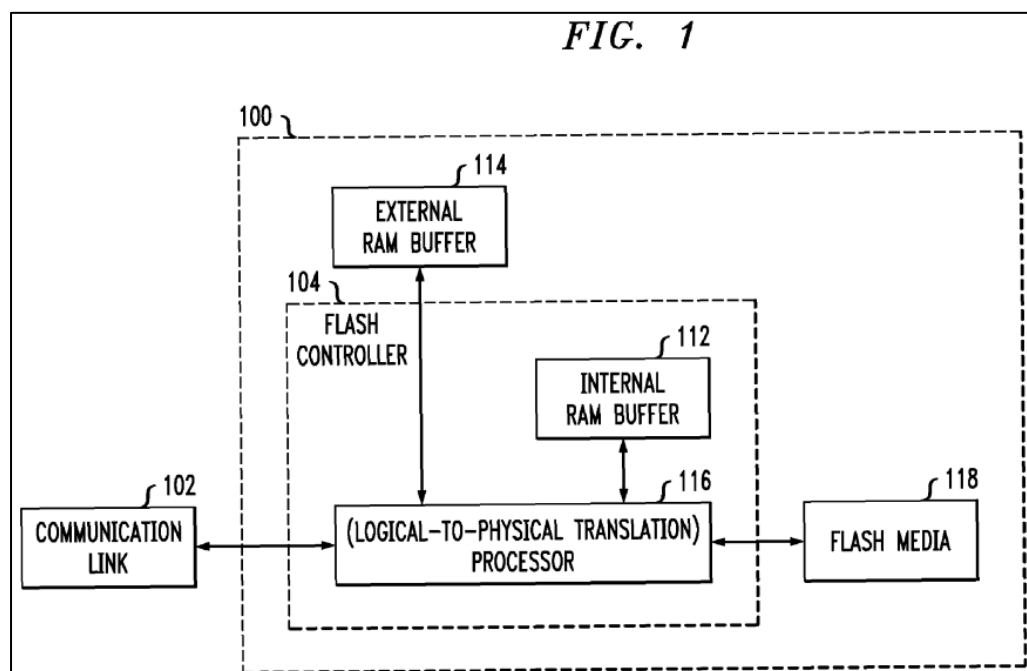
129. It is my opinion that Reiter renders this limitation obvious.

3. Claim 2, Limitation 2[pre]: *The storage device of claim 1 wherein:*

130. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 2[a]: *said circuitry comprises at least one processor; and*

131. Reiter’s flash controller 104 (*i.e., said circuitry*) *comprises* “processor 116” (*i.e., at least one processor*). EX1004, 7:21-22, 7:31-39, FIG. 1:



132. It is my opinion that Reiter renders this limitation obvious.

- (b) **Limitation 2[b]:** *the storage device further comprises instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor.*

133. Reiter's flash system 100 (*i.e.*, **storage device**) includes flash controller 104. EX1004, FIG. 1. And flash controller 104 includes processor 116. EX1004, 7:19-39, FIG.1. "Processor 116 includes software and/or firmware as needed for operation" (*i.e.*, **said instructions when executed control the at least one processor**), which a POSITA would understand as including **instructions stored on at least one physical storage medium**. EX1004, 7:31-34.

134. It is my opinion that Reiter renders this limitation obvious.

4. Claim 3, Limitation 3[pre]: *The storage device of claim 1 wherein:*

135. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 3[a]:** *the incoming data access requests are read requests received from a host;*

136. Given the similarity in claim language, my opinion for Section VII.A.2.b applies to this limitation.

- (b) **Limitation 3[b]:** *the storage device can receive a write request from the host, wherein the circuitry is to, for the received write request, derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion, identify an addressed segment within the addressed block device from the second address portion, identify a corresponding logical erase unit from the third address portion and identify a physical erase unit which is mapped to the corresponding logical erase unit;*

137. A POSITA would be motivated to implement Reiter's flash system 100 (**storage device**) to **receive** and execute ZBC commands from an external device (**from the host**). Section VII.A.2.b. The ZBC command set includes a write command. EX1007, Table 112: (below):

Table 112 — WRITE (32) command									
Byte	Bit	7	6	5	4	3	2	1	0
0		OPERATION CODE (7Fh)							
1		CONTROL							
2		Reserved							
...									
5									
6		Reserved			GROUP NUMBER				
7		ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Bh)							
9									(LSB)
10		WRPROTECT			DPO	FUA	Reserved	Obsolete	Reserved
11		Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS							
...									
19									(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG							
...									
23									(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG							
25									(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK							
27									(LSB)
28	(MSB)	TRANSFER LENGTH							
...									
31									(LSB)

138. Flash system's 100 translation of ZBC read command's LBA also applies to a ZBC write command, (*see* EX1004, 8:54-56; Sections VII.A.2.c-f.) thereby disclosing this limitation.

139. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) **Limitation 3[c]:** *the storage drive is to, in association with fulfillment of the received write request, identify a physical storage location in which associated write data has been stored and which is associated with the corresponding logical erase unit, and to transmit information representing an address value to the host corresponding to the identified physical storage location; and*

Limitation 3[d]: *for one of the read requests which is directed to data which has been written into the flash memory pursuant to the received write request, at least one of the first address portion, the second address portion or the third address portion associated with the one of the read requests is dependent on the address value which has been transmitted to the host in association with the data*

which has been written into the flash memory pursuant to the received write request.

140. Reiter discloses translating a read request's LBA to "data pointer 423, which points to the physical address." EX1004, 9:60-63. The same translation applies to write operations (*in association with fulfillment of the received write request*), except the resulting pointer is called the "active page address" (*identify a physical storage location in which associated write data has been stored and which is associated with the corresponding logical erase unit*). EX1004, 8:54-56, 13:55-60, FIG. 6b; Sections VII.A.2.g-h.

141. ZBC write commands also cause flash system 100 (*storage drive, in association with fulfillment of the received write request*) to transmit information back to the host the upon command completion. "The SCSI command set assumes an underlying request-response protocol." EX1008, xl. The response is "returned ... on completion of a SCSI command." EX1008, 11.

142. A POSITA would be motivated to modify the transmission to include information identifying the completed command (information representing an address value to the host corresponding to the identified physical storage location). Reiter discloses processing multiple commands in parallel. *See* EX1004, 8:50-53, 15:55-64, 16:62-65, FIG. 3. SPC-4 also discloses processing multiple commands in parallel. *See* EX1008, xl, 431, 438, 472, 821, 867, Table A.1 (referring to a "queue" or "task set"). Because concurrent commands can complete out of order, information is needed in the response to identify what command completed. Identifying the completed command allows the host to verify command completion or repeat commands that return errors. Because information identifying the command also indicates the LBA associated with the command, such information *represents an address value to the host corresponding to the identified storage location*.

143. Moreover, ZBC discloses incorporating such information in some circumstances. For instance, if a ZBC write command writes across two or more zones, the zoned block device flags “terminates [the] command” and “return[s] the write pointer[,] ... specified by the [LBA] field of that command[,] in the information field of the sense data.” EX1006, 18. “[S]ense data” is part of the SCSI response. EX1008, 21.

144. A POSITA would be motivated to restrict subsequent read commands (*for one of the read requests*) targeting LBAs subject to pending write operations (*for one of the read requests which is directed to data which has been written into the flash memory pursuant to the received write request*) until after receiving a completion response of the pending write operations (*dependent on the address value which has been transmitted to the host in association with the data which has been written into the flash memory pursuant to the received write request*). Incorporating command-identifying information into the SCSI response ensures command completion. Allowing read commands prior to write command completion risks reading incorrect data. Because incoming read requests are restricted to LBAs without pending write commands, the LBAs of incoming read commands that were previously written are dependent on the response’s command-identifying information (*address value which has been transmitted to the host in association with the data which has been written into the flash memory pursuant to the received write request*). The underlying *third address portion* is translated from the LBA (Section VII.A.2.c) and is therefore also *dependent on* the completion response.

145. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

5. Claim 4, Limitation 4[pre]: *The storage device of claim 1 wherein:*

146. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 4[a]:** *the storage device is to receive a maintenance request from a host which addresses a block device and a segment within the block device addressed by the maintenance request; and*

147. A POSITA would be motivated to implement Reiter’s flash system 100 (*i.e.*, *storage device*) to *receive* and execute ZBC commands. Section VII.A.2.b. ZBC includes a reset write pointer command (*i.e.*, *maintenance request from a host*). EX1006, 29, Table 16:

Table 16 — RESET WRITE POINTER command								
Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Fh)							
1	Reserved			SERVICE ACTION (14h)				
3	(MSB)							
...	ZONE ID							
9	(LSB)							
10	Reserved							
...								
13								
14	Reserved						RESET ALL	
15	CONTROL							

148. The reset write pointer command resets the write pointer to the zone’s starting logical block address. EX1006, 14, Figure 2:

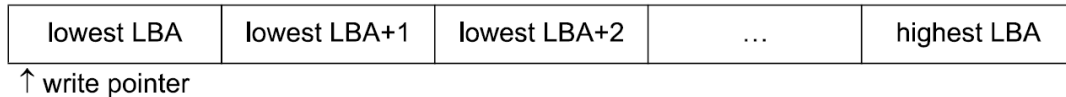


Figure 2 — Write pointer zone and write pointer after reset write pointer operation with no subsequent writes

149. Like a ZBC read command, a POSITA would be motivated to implement the reset write pointer command to *address* a logical unit (*i.e.*, *block device*). Section VII.A.2.b; *see* EX1008, 14, 61; *see also* EX1006, x, 1, 4, 29. Also like a ZBC read command, the reset write pointer command *addresses* a zone (*i.e.*, *segment*) via a “ZONE ID” parameter “specif[ying] the zone start LBA,” (EX1006, 29, Table 16.) which is *within* the logical unit (*i.e.*, *within the block device addressed*). Section VII.A.2.e.

150. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(b) **Limitation 4[b]: *the storage device comprises circuitry to erase, in association with execution of the maintenance request, one or more of the physical erase units which are mapped to the segment addressed by the maintenance request.***

151. A POSITA would be motivated to implement the reset write pointer command (*i.e., in association with execution of the maintenance request*) to cause Reiter's flash system 100 (*i.e., storage device*) to erase the physical blocks (*i.e., one or more physical erase units*) associated with the targeted zone (*i.e., mapped to the segment addressed by the maintenance request*). After applying a reset write pointer command, the zone's data is no longer accessible. *See* EX1006, 15-17. For instance, "[a] read operation on an LBA that has not been written since the last reset write pointer operation ... shall return user data set to the last initialization pattern ... set by ... the manufacturer[]," not user data. EX1006, 16-17.

152. If the data is no longer accessible, it should be erased immediately or at least immediately queued for erasure. Immediate erasure is a simpler and with less risk of error. Deferred execution risks an inappropriate order of operations, and requires complex management to avoid an inappropriate ordering. Alternatively, immediate queueing for erasure provides flexibility for erasing when normal read and write operations are not being processed. Reiter contemplates delays of normal operations for erasure. EX1004, 4:19-26. "[g]arbage collection routines," such as for "erasing out-of-date blocks," "impact performance of the flash memory system by utilizing processor resources and potentially delaying write operations" EX1004, 4:19-26.

153. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

6. Claim 5, Limitation 5[pre]: *The storage device of claim 1 wherein:*

154. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 5[a]: *the flash memory is to have block devices, including the addressed block device; and*

155. A POSITA would understand that a ZBC read commands targets one of the logical units (*i.e., addressed block device*) on flash system's 100 flash media 118 (*i.e., flash memory is to have block devices*). Section VII.A.2.c.

156. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

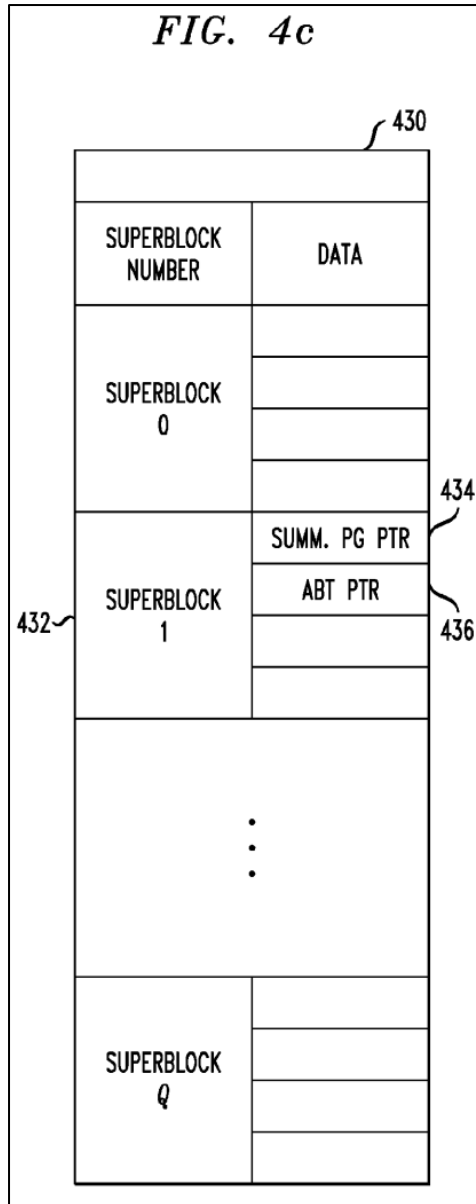
(b) Limitation 5[b]: *in connection with a given one of the incoming data access requests, the circuitry is to:*

157. As addressed in Section VII.A.6.c, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) Limitation 5[b][i]: *identify and retrieve a look-up table specific to the addressed block device which was identified from the first address portion derived from the given one of the incoming data access requests, and*

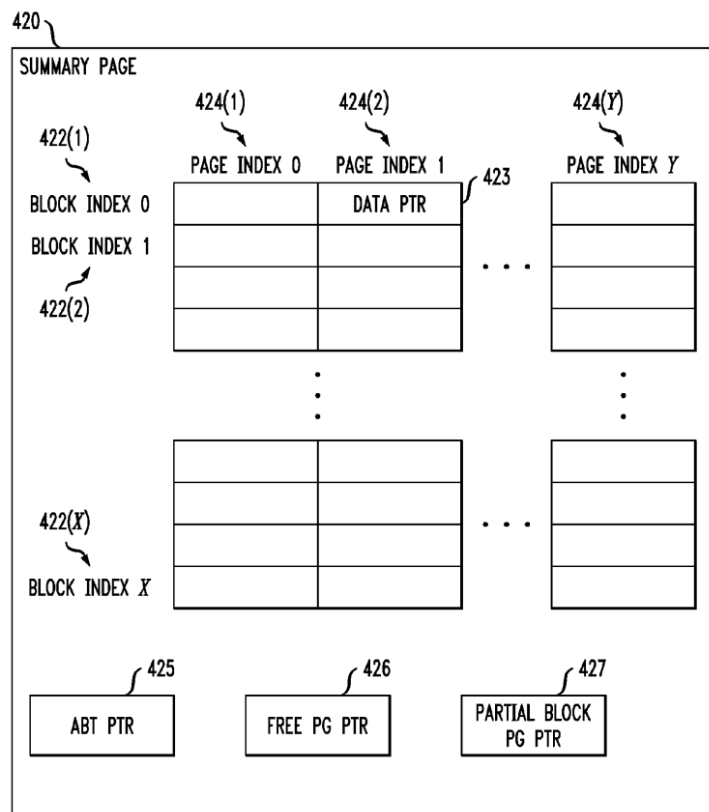
Limitation 5[b][ii]: *identify a physical storage location in the flash memory dependent on the look-up table.*

158. Reiter discloses a page global directory 430 including a summary page pointer 434 for each superblock, where the summary page pointer is selected by superblock index 410. EX1004, 10:13-19, FIG. 4c. The "summary page associated with [each] Superblock" is identified and retrieved with summary page pointer 434 (*i.e., identify and retrieve a look-up table*). EX1004, 10:13-19, FIG. 4c:



159. The “summary page” (*i.e.*, **look-up table**) “contains mapping data for the ... Superblock.” EX1004, 9:52-54. Summary page’s “block indices 422(1)-422(X) and page indices 424(1)-424(Y)” are indexed by block index 412 and page index 414 (*i.e.*, **dependent on the look-up table**) to obtain “data pointer 423, which points to the [corresponding] physical address” (*i.e.*, **identify a physical storage location in the flash memory**). EX1004, 9:60-63, FIG. 4b:

FIG. 4b



160. Summary pages are specific to the ZBC read command's targeted logical unit (addressed block device which was identified from the first address portion derived from the given one of the incoming access requests). Each logical unit has its own set of zones. *See* EX1006, 1, 13. Because a POSITA would be motivated to use Reiter's superblock as a ZBC zone, (Section VII.A.1.) each logical unit has its own superblocks. And "[e]ach Superblock has a summary page" EX1004, 9:52-53.

161. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

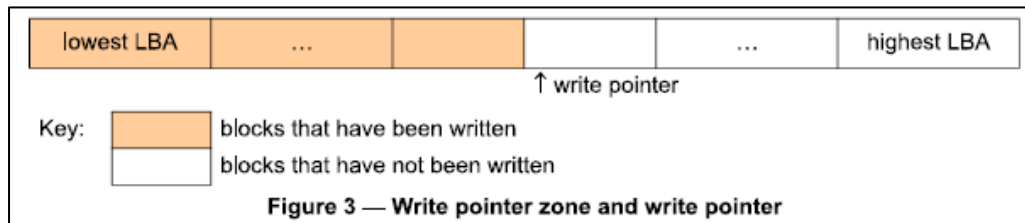
7. Claim 6, Limitation 6[pre]: *The storage device of claim 1 wherein:*

162. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 6[a]:** *the storage device further comprises circuitry to track metadata for the addressed segment, where said metadata indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased; and*

163. Reiter’s flash system’s 100 (*i.e.*, **storage device**) flash controller 104 (*i.e.*, **circuitry to**) maintains a partial block pointer (*i.e.*, **track metadata**) for each superblock (*i.e.*, **addressed segment**). EX1004, 9:52-10:1, FIG. 4b. The partial block pointer points to the “location of the next page of a partially written block” (*i.e.*, **metadata indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased**). EX1004, 9:62-10:1.

164. Alternatively, Patent Owner’s infringement contentions assert that a zone’s write pointer discloses the limitation. EX1011, 63. ZBC also relies on write pointers for zones. EX1006, 14, 17, Figure 3:



165. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

- (b) **Limitation 6[b]:** *the storage device further comprises circuitry to transmit information to a host dependent on said tracked metadata.*

166. A POSITA would be motivated to implement Reiter’s flash system 100 (*i.e.*, **storage device**) to execute ZBC commands. Section VII.A.2.b. ZBC includes a report zones command. EX1006, Table 12. The ZBC report zones command “requests that the device server return the zone structure of the zoned block device”—called “parameter data”—(*i.e.*, **transmit**

information to a host). EX1006, 24. This includes a list of zone descriptors.

Bit Byte	7	6	5	4	3	2	1	0
1	(MSB)							
...	ZONE LIST LENGTH (n-63)							
3								
4	Reserved							SAME
5	Reserved							
...								
63								
Zone descriptors list								
64	Zone descriptor [first]							
...								
127	⋮							
n-63	Zone descriptor [last]							
...								
n								

167. The zone descriptors include the zone's write pointer (*i.e., dependent on said tracked metadata*). EX1006, Table 13.

Table 13 — Zone descriptor format								
Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			RESET
2	Reserved							
...								
7								
8	(MSB)	ZONE LENGTH						
...								
15	(LSB)							
16	(MSB)	ZONE START LBA						
...								
23	(LSB)							
24	(MSB)	WRITE POINTER LBA						
...								
31	(LSB)							
32	Reserved							
...								
63								

168. The reported data is *dependent on* the write pointer (*i.e., tracked metadata*).

169. Patent Owner's infringement contentions asserts the same mapping. EX1018, 65.

The NVMe Zoned Namespace Specification is

“able to transmit a Report Zones data structure that includes a Zone Descriptor to the host, which includes the write pointer (metadata).” EX1011, 65.

170. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

8. Claim 7, Limitation 7[pre]: *The storage device of claim 1 wherein:*

171. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 7[a]: *the incoming data access requests are read requests received from a host;*

172. Given the similarity in claim language, my opinion for Section VII.A.4.a applies to this limitation.

(b) Limitation 7[b]: *the storage device can receive write requests from the host, and the storage device comprises circuitry to issue commands to the flash memory in fulfillment of the write requests;*

173. Given the similarity in claim language, my opinion for Sections VII.A.4.b-c applies to this limitation.

(c) Limitation 7[c]: *the storage device is to assign an address value in association with fulfillment of each of the write requests, in order to identify a physical storage location in which associated write data has been stored, and comprises circuitry to transmit information representing each assigned address value to the host; and*

174. Given the similarity in claim language, my opinion for Section VII.A.4.c applies to this limitation.

(d) Limitation 7[d]: *for one of the incoming data access requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, the addressed storage location identified from the third portion is dependent on*

the assigned address value which was transmitted to the host in association with the one of the write requests.

175. Given the similarity in claim language, my opinion for Section VII.A.4.c applies to this limitation.

176. The only difference is the that this limitation requires the *third [address] portion is dependent on the ... address value ... transmitted to the host in association with one of the write requests* rather than *at least one of the first address portion, the second address portion or the third address portion*. But dependence on the third address portion is also discussed with Section VII.A.4.c.

9. Claim 8, Limitation 8[pre]: *The storage device of claim 1 wherein:*

177. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 8[a]:** *the storage device can receive a request from a host to copy host-selected data from a first storage location in flash memory to a second storage location in the flash memory, and the storage device comprises circuitry to issue commands to the flash memory, in response to the request from the host to copy, to read the host-selected data from the first storage location in the flash memory and write the host-selected data to the second storage location in the flash memory without sending the data being copied to the host; and*

178. A POSITA would be motivated to implement Reiter's flash system 100 (*i.e., storage device*) to *receive* and execute ZBC commands. Section VII.A.2.b. ZBC includes an extended copy command (*i.e., request from a host to copy*). EX1007, Table 33; SPC-4, 204, Table 99; EX1006, 1, 22. The extended copy command "cop[ies] data from one set of copy sources (e.g., ... logical units) to a set of copy destinations (e.g., ... logical units)" on flash system 100 (*i.e., copy host-selected data from a first storage location in flash memory to a second storage location in flash memory*) EX1008, 287-89. Specifically, the extend copy command includes a

parameter list, which has a set of segment descriptors. EX1008, Table 128:

Table 128 — EXTENDED COPY(LID4) parameter list (part 1 of 2)								
Bit Byte	7	6	5	4	3	2	1	0
0	PARAMETER LIST FORMAT (01h)							
1	Reserved		STR	LIST ID USAGE		PRIORITY		
2	(MSB) _____							
3	HEADER CSCD DESCRIPTOR LIST LENGTH (0020h) _____							
4	(LSB) _____							
...								
14	Reserved							
15	Reserved					G_SENSE		IMMED
16	HEADER CSCD DESCRIPTOR TYPE CODE (FFh)							
17								
18								
19	Reserved							
20	(MSB) _____							
...								
23	LIST IDENTIFIER _____							
24	(LSB) _____							
...								
41	Reserved							
42	(MSB) _____							
43	CSCD DESCRIPTOR LIST LENGTH (n-47) _____							
44	(MSB) _____							
45	SEGMENT DESCRIPTOR LIST LENGTH (m-n) _____							
46	(MSB) _____							
47	INLINE DATA LENGTH (k-m) _____							
	(LSB) _____							
	CSCD descriptor list							
48								
...								
79	CSCD descriptor [ID 1] (see 6.4.5) _____							
	⋮							
n-31								
...								
n	CSCD descriptor [ID x] (see 6.4.5) _____							

Table 128 — EXTENDED COPY(LID4) parameter list (part 2 of 2)								
Bit Byte	7	6	5	4	3	2	1	0
	Segment descriptor list							
n+1								
...	Segment descriptor (see 6.4.6) [first] _____							
	:							
...								
m	Segment descriptor (see 6.4.6) [last] _____							
m+1								
...	Inline data							
k								

179. And the segment descriptors detail the source logical block address (*i.e.*, **first storage location**) and destination logical block address (*i.e.*, **second storage location**). EX1008,

Table 114:

Table 153 — Block device to block device segment descriptor													
Bit Byte	7	6	5	4	3	2	1	0					
0	DESCRIPTOR TYPE CODE (02h or 0Dh)												
1	Reserved						DC	CAT					
2	(MSB)	DESCRIPTOR LENGTH (0018h)						(LSB)					
3													
4	(MSB)	SOURCE CSD DESCRIPTOR ID						(LSB)					
5													
6	(MSB)	DESTINATION CSD DESCRIPTOR ID						(LSB)					
7													
8	Reserved												
9	Reserved												
10	(MSB)	BLOCK DEVICE NUMBER OF BLOCKS						(LSB)					
11													
12	(MSB)	SOURCE BLOCK DEVICE LOGICAL BLOCK ADDRESS						(LSB)					
...													
19													
20	(MSB)	DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS						(LSB)					
...													
27													

180. A POSITA would understand that executing the extended copy command (*i.e., in response to the request from the host to copy*) means flash controller 104 (*i.e., circuitry*) reading data from the source logical block address (*i.e., host-selected data from the first storage location in the flash memory*) and writing the data to the destination logical block address (*i.e., host-selected data to the second storage location in the flash memory*).

181. SPC-4 discloses executing the extended copy command only via the SCSI device—Reiter’s flash system 100 (*i.e., without sending the data being copied to the host*). SPC-4 discloses executing the extended copy command only via a “copy manager,” which “may be contained in ... a standalone SCSI target device.” EX1008, 245-46. Using a copy manager on the SCSI device—flash system 100—to copy data between logical units on the same SCSI device means that the copied data is not sent to the host.

182. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

- (b) **Limitation 8[b]:** *the request from the host to copy is to designate destination address information corresponding to the second storage location, wherein the storage device is to identify from the destination address information a destination block device and an addressed segment within the destination block device, and wherein the second storage location corresponds to the addressed segment within the destination block device.*

183. SPC-4 discloses that the extended copy command from the host (*i.e., the request from the host to copy*) includes a destination logical block address (*i.e., designate destination address information corresponding to the second storage location*). Section VII.A.9.a. A POSITA would be motivated to perform the same address translation as provided to a ZBC write commands, thereby disclosing this limitation. Section VII.A.4.b.

184. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

10. Claim 9, Limitation 9[pre]: *The storage device of claim 8 wherein*

185. As addressed in Section VII.A.9, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 9[a]:** *the storage device is to identify from the destination address information a destination logical erase unit and is to select the second storage location in dependence on the identified destination logical erase unit.*

186. A POSITA would be motivated to implement Reiter's flash system 100 (*i.e., storage device*) to perform the same address translation on the LBA of a ZBC extended copy command (*i.e., from the destination address information*) to *identify* logical block number 406 (*i.e., destination logical erase unit*). Section VII.A.9.a. Further translation, logical block number 406 (*i.e., in dependence on the identified destination logical erase unit*) *selects* data write pointer 423 or active page address (*i.e., the second storage location*). EX1004, FIGS. 4a-b.

187. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4

renders this limitation obvious.

11. Claim 10, Limitation 10[pre]: *The storage device of claim 1 wherein*

188. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 10[a]:** *the storage device is to store information identifying a number of structures in the flash memory and is to transmit the information identifying the number of structures to a host, wherein the host is to issue at least one request to the storage device in which the at least one request is dependent on the information transmitted to the host identifying the number of structures, and wherein the storage device comprises circuitry to fulfill the at least one request by issuing commands to the flash memory for which a quantum of flash memory selected by the commands which are issued in response to the at least one request is dependent on the information transmitted to the host identifying the number of structures.*

189. ZBC's report zones command "return[s]" (*i.e.*, ***transmit***) a "[l]ist of zones in the zoned block device" back *to [the] host* via "parameter data" that includes the length of the zone list and zone descriptors indicating the starting logical block address and length of each zone (*i.e.*, ***the information identifying the number of structures to a host***). EX1006, 25, Table 11, Tables 12, 13:

Table 12 — REPORT ZONES parameter data								
Bit Byte	7	6	5	4	3	2	1	0
1	(MSB)							
...	ZONE LIST LENGTH (n-63)							
3	(LSB)							
4	Reserved							SAME
5	Reserved							
...								
63								
Zone descriptors list								
64	Zone descriptor [first]							
...								
127								
	⋮							
n-63	Zone descriptor [last]							
...								
n								

Table 13 — Zone descriptor format								
Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			RESET
2	Reserved							
...								
7								
8	(MSB)							
...	ZONE LENGTH							
15	(LSB)							
16	(MSB)							
...	ZONE START LBA							
23	(LSB)							
24	(MSB)							
...	WRITE POINTER LBA							
31	(LSB)							
32	Reserved							
...								
63								

Because a POSITA would be motivated to map ZBC zones onto Reiter superblocks, (Section VII.A.1.) a POSITA would be understand storing, or be motivated to store, zone metadata in a

superblock's summary page, which stores a superblock's metadata. EX1004, 9:52-10:12; FIG. 4b.

190. Executing a valid reset write pointer command (Section VII.A.5.a.) (*i.e., host is to issue at least one request to the storage device*) depends on the information reported by the report zones command (*i.e., information transmitted to the host identifying the number of structures*). A valid reset write pointer command requires a "ZONE ID" parameter that "specifies the zone start LBA" EX1006, 29. The zone start LBA is provided by the report zones command. EX1006 25, Table 13 (above).

191. Reiter's flash system 100 executing a reset write pointer command (Limitation 4[a].) (*i.e., circuitry to fulfill the at least one request by issuing commands to the flash memory*) "sets a write pointer to the lowest LBA of a write pointer zone" (*i.e., quantum of flash memory selected by the commands which are issued in response to the at least one request is dependent on the information transmitted to the host identifying the number of structures*). EX1006, 3.

192. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

12. Claim 11, Limitation 11[pre]: *The storage device of claim 10 wherein*

193. As addressed in Section VII.A.11, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 11[a]: *the information transmitted to the host identifying the number of structures identifies an amount of flash memory that is to be erased as a unit.*

194. The list of zones reported by the report zones command (*i.e., information transmitted to the host identifying the number of structures*) also provides the length of each zone (*identifies an amount of flash memory that is to be erased as a unit*). The report zones command "return[s] ... parameter data" that includes a zone descriptor for each zone indicating the length of each zone. EX1006, 25, Tables 11, 12, Table 13:

Table 13 — Zone descriptor format								
Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			RESET
2	Reserved							
...								
7								
8	(MSB)	ZONE LENGTH						
...								
15		(LSB)						
16	(MSB)	ZONE START LBA						
...								
23		(LSB)						
24	(MSB)	WRITE POINTER LBA						
...								
31		(LSB)						
32	Reserved							
...								
63								

195. As discussed with Section VII.A.5.b, a POSITA would be motivated to implement the reset write pointer command to cause erasure of the targeted zone. Because the reset write pointer command erases an entire zone, the zone length *identifies an amount of flash memory that is to be erased as a unit*.

196. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

13. Claim 12, Limitation 12[pre]: *The storage device of claim 1 wherein*

197. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 12[a]: *the addressed segment corresponds to an integer number of physical erase units for which the failure condition has not been detected, and the third address portion***

specifies a location within a sequential range of locations that correspond to the integer number of physical erase units.

198. A POSITA would be motivated to use Reiter's superblocks as ZBC zones (*i.e., addressed segment*). Section VII.A.2.e. "[A] Superblock ... is a logical collection of blocks" (*i.e., integer number of physical erase units*). EX1004, 9:11-13. Further, logical sector number 404, or alternatively the bit string of logical block number 406 and logical page number 408, (*i.e., third address portion*) correspond to a location pulled from superblock's sequentially indexed blocks and pages of those blocks (*i.e., specifies a location within a sequential range of locations that correspond to the integer number of physical erase units*). EX1004, 9:5-25, 9:52-63, 12:33-35, 12:67-13:3, FIGS. 4a-c.

199. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

14. Claim 16, Limitation 16[pre]: *The storage device of claim 1 wherein:*

200. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 16[a]: *the flash memory is to have block devices, including the addressed block device; and*

201. Given the similarity in claim language, my opinion for Section VII.A.6.a applies to this limitation.

(b) Limitation 16[b]: *the storage device further comprises circuitry to:*

202. As addressed in Sections VII.A.14.c-d, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) Limitation 16[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, and*

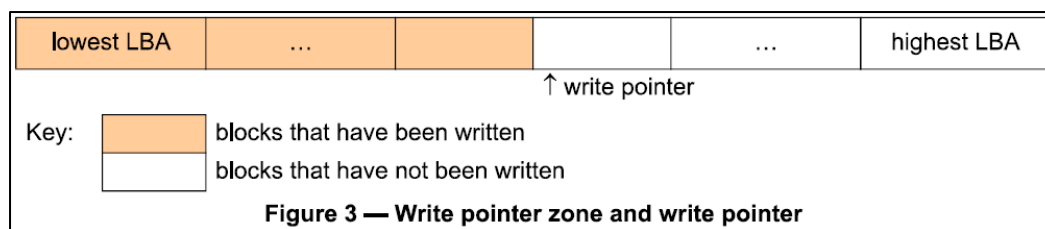
203. Reiter's flash controller 104 (*i.e., circuitry*) stores a RESET bit (*i.e., metadata*) for

each zone of a block device (*i.e., on an independent basis for flash memory associated with each one of the block devices*). A RESET bit is stored for each zone because a ZBC report zones command is optionally set to return the “[l]ist [of] ones for which the RESET bit is set to one.” EX1006, 25. A POSITA would understand that flash controller returning the list of zones with RESET bits set to one requires a RESET bit to be stored for each zone.

204. It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

- (d) **Limitation 16[b][ii]:** *receive a query, issued by a host device, and responsive to the query, compare the metadata to at least one threshold, to identify specific metadata, and transmit information to the host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.*

205. Reiter’s flash controller 104 (*i.e., circuitry*) *receives* a ZBC report zones command (*i.e., query*), that is sent by the host (*i.e., issued by a host device*). ZBC discloses that a ZBC report zones command is optionally set to return the “[l]ist [of] ones for which the RESET bit is set to one.” EX1006, 25. Following this report zones command (*i.e., responsive to the query*), flash controller 104 pull’s each zone’s RESET bit (*i.e., identify specific metadata*), checks each zone’s RESET bit is equal to one (*i.e., compare metadata to at least one threshold*), and returns the list of zones with a RESET bit is equal to one (*i.e., transmit information to the host device*). The list of zones is sent back via parameter data, which includes a list of zone descriptors. The zone descriptors include the zone’s write pointer logical block address, which details how much of the zone has user data (*i.e., such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata*). See EX1006, Figure 3:



It is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

15. Claim 20, Limitation 20[pre]: *The storage device of claim 1 wherein*

206. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 20[a]: *the division operation is applied to subdivide the third address portion into respective first and second bit fields.*

207. Given the similarity in claim language, my opinion for Section VII.A.2.g applies to this limitation.

208. It is my opinion that Reiter renders this limitation obvious.

16. Claims 21-23

209. Claims 21-23 are substantively identical to claim 1. Claim 21 is substantively identical to claim 1. Claim 21 adds that “block devices,” “segments,” and “storage locations” are at “hierarchically-superior,” “-intermediate,” and “-inferior level[s],” respectively, and that block devices and segments contain mutually exclusive sets of segments and logical erase units, respectively. This maps to limitations 1[b][iii]-[d] which require “***block device[s]***,” “***segment[s]*** ***within the ... block device[s]***,” “***erase unit[s]*** ***within the ... segment[s]***,” and “***storage location[s]*** ***within the ... segment[s]***.” And Ground I discloses logical units, each with its own LBA range, thereby its own zones/superblocks, where each zone/superblock contains its own logical blocks and pages. EX1004, FIGs. 4a-b (depicting a superblock’s summary page 420 containing a list of block indices 422(1)-(X), where block indices 422(1)-(X) each contain a set of page index indices

424(1)-(Y)); EX1006, 13-14, Figures 1-2; EX1008, 14, 61 (“[L]ogical block addresses on a logical unit ... shall begin with block zero and be contiguous up to the last logical block of that logical unit”); Sections VII.A.2.c-e. Claim 22 is substantively identical to claim 21 incorporated with dependent claim 3. Claim 23 is substantively identical to claim 1. Claim 1 is directed to “[a] storage device,” whereas claim 23 is directed to “[a] method of operation in a storage device.” This distinction is immaterial since Reiter discloses systems and steps in operating those systems. *E.g.*, EX1004, FIGs. 4a-c, 5a-b, 6a-b.

Claim 21	Claim 22	Claim 23	Corresponding Claim Limitation(s)
21[pre]	22[pre]	23[pre]	1[pre]
21[a]	22[a]	23[pre]	1[a] (1[b][iii]-1[d] for claims 21-22)
21[b]	22[b]	23[a], 23[b]	1[b] (3[b] for claim 22)
21[b][i]	22[b][i]	23[b][i]	1[b][i]
21[b][ii]	22[b][ii]	23[b][i]	1[b][ii]
21[b][iii]	22[b][iii]	23[b][ii]	1[b][iii]
21[b][iv]	22[b][iv]	23[b][iii]	1[b][iv]
21[c]	22[c]	23[b][iv]	1[c]
21[d]	22[d]	23[b][v]	1[d]
21[e]	22[e]	23[c]	1[e]
	22[f], 22[f][i], 22[f][ii], 22[f][iii], 22[f][iv]		3[b]
	22[f][v], 22[f][vi]		3[c]
	22[g]		3[d]

210. It is my opinion that Reiter renders claims 21-23 obvious.

B. GROUND II: CLAIMS 1, 13-14, 19 ARE RENDERED OBVIOUS BY REITER IN LIGHT OF ZBC, SBC-4, SPC-4, AND SINCLAIR-367

1. Motivation To Combine Sinclair-367 With The Combined System Of Reiter And ZBC, SBC-4, And SPC-4

211. A POSITA would have been motivated to combine Reiter’s flash system executing ZBC commands (Section VII.A.1.) with Sinclair-367’s bad block management, metadata storage,

and planes and metablocks.

212. Sinclair-367 manages bad blocks—identifying “defective cells within the erase block,” and remapping the “logical address” with “the address of any substitute” block, where the “primary erase block is defective.” EX1009, ¶¶[0107]-[0109], [0110]. This improves system resiliency. Because Reiter contemplates bad block management, (*see* Section VII.A.2.h.) a POSITA would have a reasonable expectation of success implementing bad block management in Reiter’s flash device.

213. Further, Sinclair-367 discloses storing in each erase unit a sector of metadata (e.g., error correcting codes (“ECC”) and number of program/erase cycles (*i.e.*, erase count)). EX1009, ¶¶[0106]-[0107], FIG. 3. ECCs recognize and repair damaged data. EX1009, ¶¶[0007], [0103]. Program/erase cycles determine the wear and tear on a physical block. EX1009, ¶[0108]. Cycle count “is used in a wear-leveling algorithm ... even[ing] out the usage (wear) of all the erase blocks,” increasing the resiliency and lifespan of the flash system. EX1009, ¶[0108]. Because a single sector is a small overhead for a block, yet reduces data loss, a POSITA would have a reasonable expectation of success storing a metadata sector in physical blocks.

214. Further, Sinclair-367 discloses dividing flash memory into planes, and forming metablocks by linking erase units in each plane. EX1009, ¶¶[0009]-[0012], [0112]-[0114]. This improve speed and throughput. Each plane operates independently—concurrently executing reading/writing. EX1009, ¶[0112]. And “to increase ... parallelism[,] erase blocks ... are logically linked to form metablocks.” EX1009, ¶[0114]. Because Sinclair-367 and Reiter disclose analogous flash systems, and planes and metablocks are common in the flash memory art, a POSITA would have a reasonable likelihood of success implementing them in Reiter’s flash system 100. Moreover, Reiter, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose analogous art. All relate to storage devices, including accessing blocks of memory.

2. **Claim 1, Limitation 1[pre]:** *The storage device of claim 1 wherein:*

Limitation 1[a]: *flash memory having physical erase units; and*

Limitation 1[b]: *circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:*

Limitation 1[b][i]: *derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,*

Limitation 1[b][ii]: *identify an addressed block device from the first address portion,*

Limitation 1[b][iii]: *identify an addressed segment within the addressed block device from the second address portion, and*

Limitation 1[b][iv]: *identify an addressed storage location within the addressed segment from the third address portion,*

Limitation 1[c]: *wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,*

Limitation 1[d]: *wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;*

215. As addressed in Sections VII.A.2.a-h, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) **Limitation 1[e]:** *the flash memory is to have block devices, including the addressed block device; and*

216. A POSITA would be motivated to modify Reiter's flash system 100 (*storage device*) to implement Sinclair-367's bad block management. Section VII.B.1. Sinclair-367 discloses "identification of any defective cells within the erase block," (*detect a failure condition of the physical erase units*) and remapping the "logical address," such as updating the Reiter's summary page 420 which "contains the ... [superblock] mapping data" for translating logical block number 406 or alternatively block index 412 (*logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected*) (EX1004, 9:52-63, FIGs.

4a-b.), with the “address of any substitute erase block in case the primary erase block is defective” *(instead correspond to one of the physical erase units for which the failure condition has not been detected)* for later use *(utilize the remapped correspondence to service one or more of the incoming data access requests)*. EX1009, ¶¶[0107]-[0109].

3. Claim 13, Limitation 13[pre]: *The storage device of claim 1 wherein:*

217. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 13[a]: *the flash memory is to have block devices, including the addressed block device; and*

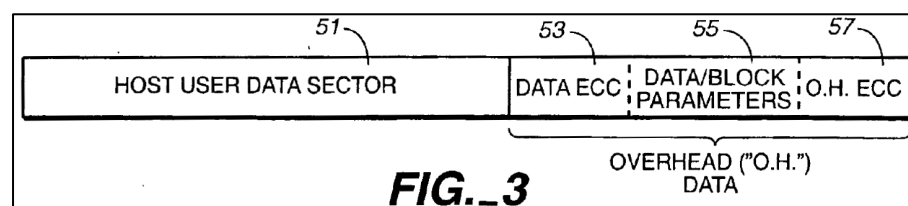
218. Given the similarity in claim language, my opinion for Section VII.A.6.a applies to this limitation.

(b) Limitation 13[b]: *the storage device further comprises circuitry to:*

219. As addressed in Sections VII.B.2.c-f, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) Limitation 13[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,*

220. With the writing capabilities of Reiter’s flash controller 104 (*i.e., circuitry*), Sinclair-367’s erase blocks use a “single sector of data” to *store* overhead data, including error correction codes (ECC) 53 for data and data/block parameters 55 (*i.e., metadata*). EX1009, ¶¶[0106]-[0110], FIG. 3:



221. Therefore, every block in a logical unit includes this single sector of overhead data (*i.e., independent basis for flash memory associated with each one of the block devices*). EX1009, ¶[0107]. Sinclair-367 also includes the “time of programming ... as a field in [the] sector or page overhead data” (*i.e., metadata representing a time since one or more physical storage locations associated with the one or more block devices has been a subject of a write operation*). EX1009, ¶[0109].

222. It is my opinion that Reiter renders this limitation obvious.

(d) Limitation 13[b][ii]: compare the metadata to at least one threshold,

223. A POSITA would be motivated to use Reiter’s flash controller 104 (*i.e., circuitry*) to **compare** the time of programming data in a physical block (*i.e., metadata*) against a retention period (*i.e., at least one threshold*). A retention period is the period of time where flash memory stores a value at a memory cell.

224. Sinclair-367 recognizes that flash memory has a retention period problem. For instance, Sinclair-367 discloses how a memory cell relies on “threshold voltage window divided into a number of ranges” to create storage states. EX1009, ¶¶[0007]-[0008]. But the maintaining the voltage level within one of the slim ranges associated with certain storage states is difficult. EX1009, ¶[0007]. “[S]torage levels do shift” shift over usage of the memory cell and surrounding memory cells. EX1009, ¶¶[0007]-[0008]. One way to predict and stop an intolerable shift in voltage levels is to check if the time since programming is nearing the average retention period of a memory cell. This can alert flash controller 104 of potential data loss, and provide an opportunity to save the data.

225. It is my opinion that Reiter renders this limitation obvious.

(e) Limitation 13[b][iii]: upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the

one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and

226. Upon nearing or reaching the average retention period (*i.e., upon detection that specific metadata has satisfied one or more thresholds*), a POSITA would be motivated to copy data out of the unreliable block past its retention period (*i.e., copy data stored at one of the one or more physical storage locations associated with the specific metadata*) and into a working physical block (*i.e., from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device*). EX1009, ¶[0136] (“Copying of data from a partially full metablock to a smaller metablock may be triggered by an elapse of time from the receipt of the stream of sectors of data 1305.”). Copying data into a new block, such as through garbage collection is faster than reading, erasing, and rewriting the same block. See also EX1009, ¶¶[0108]-[0109], [0114], [0132], [0177].

227. It is my opinion that Reiter renders this limitation obvious.

(f) **Limitation 13[b][iv]:** *remap the logical erase unit corresponding to the first one of the physical erase units for which the failure condition has not been detected by said storage device to the second one of the physical erase units for which the failure condition has not been detected by said storage device.*

228. Sinclair-367 discloses that copying data away from a failing block to a new block also includes remapping the logical address from the old physical address to the new physical address (*i.e., remap the logical erase unit corresponding to the first one of the physical erase units for which the failure condition has not been detected by said storage device to the second one of the physical erase units for which the failure condition has not been detected by said storage device*). For example, the “logical address of the data ... mapped into th[e] ... address of

any substitute erase block” if “the primary erase block is defective.” EX1009, ¶¶[0108]-[0109].

229. It is my opinion that Reiter renders this limitation obvious.

4. Claim 14, Limitation 14[pre]: *The storage device of claim 1 wherein:*

230. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 14[a]: *the flash memory is to have block devices, including the addressed block device; and*

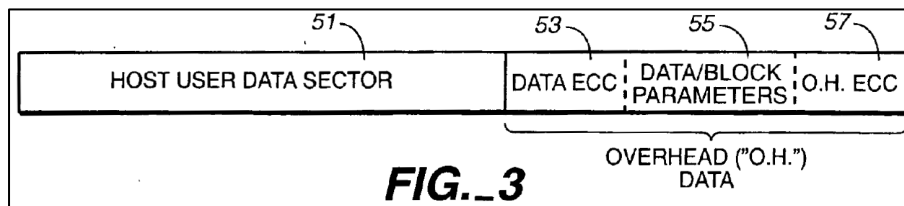
231. Given the similarity in claim language, my opinion for Section VII.A.6.a applies to this limitation.

(b) Limitation 14[b]: *the storage device further comprises circuitry to:*

232. As addressed in Sections VII.B.3.c-d, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) Limitation 14[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata is dependent on an erase count of associated physical memory space,*

233. With the writing capabilities of Reiter’s flash controller 104 (*i.e., circuitry*), Sinclair-367’s erase blocks use a “single sector of data” to *store* overhead data, including error correction codes (ECC) 53 for data and data/block parameters 55 (*i.e., metadata*). EX1009, ¶¶[0107]-[0110], FIG. 3:



234. Therefore, every block in a logical unit includes this single sector of overhead data (*i.e., independent basis for flash memory associated with each one of the block devices*).

EX1009, ¶[0107]. Sinclair-367 also discloses that “parameter 55 may include a quantity related to the number of program/erase cycles experiences by the erase block” (*i.e., wherein the stored metadata is dependent on an erase count of associated physical memory space*). EX1009, ¶[0109].

235. It is my opinion that Reiter renders this limitation obvious.

(d) **Limitation 14[b][ii]:** *compare the metadata to at least one threshold,*

Limitation 14[b][iii]: *upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and*

Limitation 14[b][iv]: *remap the logical erase unit corresponding to the first physical erase unit for which the failure condition has not been detected by said storage device to the second physical erase unit for which the failure condition has not been detected by said storage device.*

236. Sinclair-367 discloses that the “number of program/erase cycles” is “used in a wear leveling algorithm,” where “logical block addresses are regularly re-mapped to a different physical block addresses in order to even out the usage.” EX1009, ¶[0108].

237. A POSITA would understand, or be motivated to modify so, that the remapping hinges on the number of program/erase cycles (*i.e., metadata*) being high enough to indicate a heavily used erase block (*i.e., at least one threshold*). See EX1009, ¶[0108]-[0109]

238. Moreover, detection that an erase block is heavily used (*i.e., upon detection that specific metadata has satisfied the one or more thresholds*) causes logical block addresses to be remapped. See EX1009, ¶[0108]-[0109]. A POSITA would understand that remapping a logical address to a different physical block means the data of the old block is copied to the new block

stop data loss (*i.e.*, *copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device*). The certain threshold indicates a heavily used erase block. See EX1009, ¶¶[0108]-[0109], [0114], [0132], [0177].

239. It is my opinion that Reiter renders this limitation obvious.

5. Claim 19, Limitation 19[pre]: *The storage device of claim 1 wherein:*

240. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter, ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 19[a]: *the flash memory comprises multiple planes; and*

241. A POSITA would be motivated to implement Reiter’s *flash* media 118 as “divided into two or more planes” (*i.e.*, *multiple planes*) per Sinclair-367’s teachings. EX1009, ¶[0111]; Section VII.B.1.

242. It is my opinion that the combined system of Reiter, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

(b) Limitation 19[b]: *the storage device is to, for each one of a set of the incoming data access requests having sequential addressing, identify a commonly addressed page and write data accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes.*

243. A ZBC write command includes a “logical block address” and “transfer length.” EX1007, Table 112.

Table 112 — WRITE (32) command

Byte	Bit	7	6	5	4	3	2	1	0
0		OPERATION CODE (7Fh)							
1		CONTROL							
2		Reserved							
...									
5									
6		Reserved			GROUP NUMBER				
7		ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Bh)							
9									
10		WRPROTECT			DPO	FUA	Reserved	Obsolete	Reserved
11		Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS							
...									
19		(LSB)							
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG							
...									
23		(LSB)							
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG							
25									
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK							
27									
28	(MSB)	TRANSFER LENGTH							
...									
31									

244. “TRANSFER LENGTH ... specifies the ... logical blocks ... that shall be transferred ... starting [from] ... the LOGICAL BLOCK ADDRESS” EX1007, 198. With a large transfer length, a ZBC write command causes several write operations across *sequential addresses*.

245. To increase throughput for these large write commands, a POSITA would be motivated to implement Reiter’s *flash* media 118 with multiple *planes*, and metablocks spanning the multiple planes, as disclosed in Sinclair-367. EX1009, ¶¶[0111]-[0112]; Sections VII.B.4.a, VII.B.1. “All blocks of the metablock are erased together, and pages of each block are generally programmed and read simultaneously” (*i.e., identify a commonly addressed page and write data accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes*). EX1009, ¶[0112].

246. It is my opinion that the combined system of Reiter, ZBC, SBC-4, and SPC-4 renders this limitation obvious.

C. GROUND III: CLAIMS 15, 17-18 ARE RENDERED OBVIOUS BY REITER IN LIGHT OF ZBC, SBC-4, SPC-4, SINCLAIR-367, AND YAMADA

1. Motivation to Combine Yamada With The Combined System Of Reiter, ZBC, SBC-4, SPC-4, And Sinclair-367

247. A POSITA would have been motivated to implement Reiter's flash system to execute ZBC commands (Sections VII.A.1, VII.B.1.) and with Yamada's maintenance request and determination units in Reiter's flash system and host processor, respectively. Reiter complains that "[g]arbage collection routines impact performance of the flash memory system by utilizing processor resources and potentially delaying write operations" EX1004, 4:23-26. In Yamada, "maintenance is performed without being ... performed together with normal system processing," but "with sufficiently long processing time." EX1010, Abstract. The maintenance request unit on the flash device determines and requests maintenance operations from the maintenance determination unit. The maintenance determination unit monitors status of the host, "determin[es] that the access device is not engaged in normal processing," and performs maintenance. EX1010, Abstract, ¶¶[0018]-[0019], [0022]. Because maintenance scheme is implemented on a nonvolatile system analogous like that in Reiter, a POSITA would have a reasonable expectation of success implementing Reiter's flash system with this scheme.

248. Moreover, Reiter, ZBC, SBC-4, SPC-4, Sinclair-367, and Yamada disclose analogous art. All relate to storage devices, including the management and control of storage devices.

2. Claim 15, Limitation 15[pre]: *The storage device of claim 1 wherein:*

249. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined

system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

- (a) **Limitation 15[a]: *the flash memory is to have block devices, including the addressed block device; and***

250. Given the similarity in claim language, my opinion for Section VII.A.6.a applies to this limitation.

- (b) **Limitation 15[b]: *the storage device further comprises circuitry to:***

251. As addressed in Sections VII.C.2.c-e, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

- (c) **Limitation 15[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,***

252. Given the similarity in claim language, my opinion for Section VII.B.2.c applies to this limitation.

- (d) **Limitation 15[b][ii]: *compare the metadata to at least one threshold, and***

253. Given the similarity in claim language, my opinion for Section VII.B.2.d applies to this limitation.

- (e) **Limitation 15[b][iii]: *responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.***

254. Yamada's maintenance determination unit 152 (on Reiter's external device (*i.e.*, *host*) (Section X.B.)) receives a "logical address included in [a] maintenance request transmitted from ... maintenance request unit 115" (on Reiter's flash system 100 (Section X.B.)) (*i.e.*, *transmit information to a host device wherein such transmitted information conveys address*

information). EX1010, ¶[0089].

255. Further, a POSITA would be motivated to trigger a maintenance request upon detecting a time since programming nearing or reaching the average retention period (*i.e.*, *responsive to detection that specific metadata has satisfied the one or more thresholds*) recorded in a physical block per Sinclair-367’s teachings. EX1009, ¶¶[0106], [0110]. Yamada discloses triggering maintenance requests for garbage collection, including wear-level management for improving data retention. EX1010, Abstract, ¶¶[0009], [0011]. Sinclair-367 recognizes that flash memory has a retention period problem, as maintaining the voltage level within one of the slim ranges associated with certain storage states is difficult. *See* EX1009, ¶¶[0007]-[0008]. “[S]torage levels do shift” shift over usage of the memory cell and surrounding memory cells. *See* EX1009, ¶¶[0007]-[0008]. The problem disclosed by Sinclair-367 is recognized and resolved with Yamada’s maintenance requests and commands. EX1010, Abstract, ¶¶[0009], [0011], [0033], [0090].

256. It is my opinion that Reiter renders this limitation obvious.

3. Claim 17, Limitation 17[pre]: *The storage device of claim 1 wherein:*

257. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 17[a]: *the flash memory is to have block devices, including the addressed block device; and*

258. Given the similarity in claim language, my opinion for Section VII.A.6.a applies to this limitation.

(b) Limitation 17[b]: *the storage device further comprises circuitry to:*

259. As addressed in Sections VII.C.3.c-d, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

- (c) **Limitation 17[b][i]:** *store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and*

260. Given the similarity in claim language, my opinion for Section VII.B.3.c applies to this limitation.

- (d) **Limitation 17[b][ii]:** *receive a query, issued by a host device, wherein said query is specific to one of the block devices, and responsive to the query, identify specific metadata, associated with the specified one of the block devices, and transmit information to the host device that conveys the specific metadata.*

261. Yamada discloses that “non-volatile memory 1020 stores management information including the information about the importance level of data to enable the management information to be shared between different access modules.” EX1010, ¶[0190]. Because management information is information governing whether maintenance is requested to improve data retention, (EX1010, Abstract, ¶¶[0009], [0011], [0033], [0090]) a POSITA would be motivated to include Sinclair-367’s program/erase-cycle-count metadata in Yamada’s management information. EX1009, ¶¶[0107]-[0108], [0110]; *see* Section VII.B.4.c-d.

262. Moreover, a POSITA would understand, or be motivated to modify so, that Yamada’s sharing of such metadata means the access module *querying* (*i.e., issued by a host device*), where the query causes (*i.e., responsive to the query*) the storage device *identify* metadata and *transmit* metadata to the access module. An external computer system, device, or host does not expect to randomly receive unrequested data from a storage drive. SCSI makes this express with its request-and-response model. “The SCSI command set assumes an underlying request-response protocol.” EX1008, xl. The request is “submitted by an application client, conveying a SCSI command.” EX1008, 11. The response is “returned ... on completion of a SCSI command.” EX1008, 11. Further, ZBC, SBC-4, and SPC-4 include commands for requesting metadata-like

data, such as report zones, get LBA status command, inquiry command, and report LUNs command. EX1006, 24, EX1007, 120; EX1008, 347, 462.

263. It is my opinion that Reiter renders this limitation obvious.

4. Claim 18, Limitation 18[pre]: *The storage device of claim 1 wherein:*

264. As addressed in Sections VII.A.2, VII.B.2, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 disclose this limitation.

(a) Limitation 18[a]: *the flash memory is to have block devices, including the addressed block device; and*

265. Given the similarity in claim language, my opinion for Section VII.A.6.a applies to this limitation.

(b) Limitation 18[b]: *the storage device further comprises circuitry to:*

266. As addressed in Sections VII.C.4.c-e, it is my opinion that the combined system of Reiter and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(c) Limitation 18[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and*

267. Given the similarity in claim language, my opinion for Section VII.B.3.c applies to this limitation.

(d) Limitation 18[b][ii]: *compare the metadata to at least one threshold, and*

268. Given the similarity in claim language, my opinion for Section VII.B.3.d applies to this limitation.

(e) Limitation 18[b][iii]: *responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or*

more physical storage locations associated with the specific metadata.

269. Yamada discloses transmit information to a host device wherein such transmitted information conveys address information. Section VII.C.2.e.

270. Further, a POSITA would be motivated to trigger a maintenance request upon detecting a high number of program/erase cycles (*i.e., responsive to detection that specific metadata has satisfied the one or more thresholds*) recorded in a physical block per Sinclair-367's teachings. EX1009, ¶¶[0106]-[0107], [0110]-[0111]. Yamada discloses triggering maintenance requests for garbage collection, including wear-level management for improving data retention. EX1010, Abstract, ¶¶[0009], [0011]. Sinclair-367 discloses that program/erase counts for physical blocks are "used in a wear leveling algorithm ... in order to even out the usage (wear) of all erase blocks"—avoiding a high number of program/erase cycles in any one block. EX1009, ¶[0108].

271. It is my opinion that Reiter renders this limitation obvious.

D. GROUND IV: CLAIMS 1-14, 16, 19-23 ARE RENDERED OBVIOUS BY BAN IN LIGHT OF ZBC, SBC-4, SPC-4, AND Sinclair-367

1. Motivation To Combine Ban, ZBC, SBC-4, SPC-4, And Sinclair-367

272. A POSITA would have been motivated to use ZBC, alongside SBC-4 and SPC-4, (collectively, "ZBC") as the command set that governs requests sent to Ban's flash device, and incorporate the flash device with Sinclair-367's bad block management, metadata storage, and planes and metablocks.

273. **First**, A POSITA would be motivated to apply the teachings of ZBC, SBC-4, and SPC-4. Section VII.A.1.

274. **Second**, a POSITA would be motivated to apply ZBC's teachings to Ban's flash system. Ban does not require a specific command set, but expressly aims to be compatible with

“typical existing operating systems.” EX1005, 1:29-39, 52-57. Ban discloses that a host “writes data to, and reads data from, a flash memory 12 in blocks at specific address locations.” EX1005, 3:59-62. A POSITA would thus be motivated to include support for commands such as those disclosed in ZBC to enable specific read and write operations based on Reiter’s memory structure.

275. A POSITA would particularly be motivated to implement commands based on ZBC in Ban’s system because SCSI protocols and commands such as those disclosed in ZBC are widely used, and would thus improve the compatibility and interoperability of Ban’s flash system.

276. Commands based on ZBC’s zoned block device model (EX1006, 1) are particularly suitable for Ban’s flash device. ZBC allows for random-access—(see EX1006, 14, 16, Figures 5, 6.)—the “object of [Ban’s] invention” (EX1005, 1:52-62, 3:52-4:10.)—while also enabling effective utilization of flash memory. ZBC includes a reset write pointer command, which resets a zone’s write pointer to the zone’s lowest LBA, and makes the zone’s data no longer accessible. EX1006, 3, 17-18, Table 16. This lines up well with flash memory’s erase operation—unique to flash memory.

277. A POSITA would have a reasonable expectation of success in adding support for ZBC in Ban’s flash system. SCSI commands and protocols such as ZBC were widely known and would have been familiar to a POSITA, (EX1200; EX1201; EX1202; EX1203.) including at the time of Ban’s invention. EX1101 (listing SCSI standards published across 1993 to 2025); EX1203, 2-3 (listing SCSI standards from 1986 to 1999 and how “SCSI ... quickly grown to ... one of the most often used interfaces for connecting to and optimizing the performance of a wide variety of peripherals); *see also* Section VI.E. SCSI specifications were intended to be used across a wide range of storage devices. EX1203, 1 (“SCSI ... has quickly grown to become one of the most often-used interfaces for connecting to and optimizing the performance of a wide variety of peripherals). Further, a POSITA would have understood that ZBC’s zoned block device model is

a logical construct that can be layered on top of any storage device.

278. **Third**, Sinclair-367 manages bad blocks—identifying “defective cells within the erase block,” and remapping the “logical address” with “the address of any substitute” block, where the “primary erase block is defective.” EX1009, ¶¶[0107]-[0108]. This improves system resiliency. Because Ban’s address-translation method is intended for block substitution, (EX1005, 2:31-49, 2:57-3:2, 5:13-17.) a POSITA would have a reasonable expectation of success implementing bad block management in Ban’s flash device.

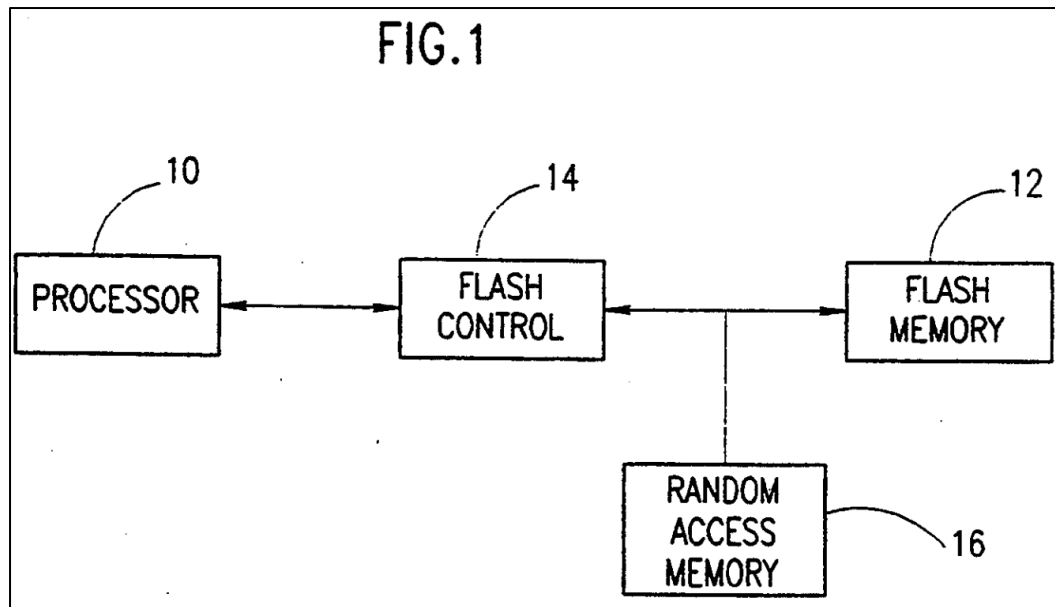
279. Further, Sinclair-367 discloses storing in each erase unit a sector of metadata, which is used to increase the flash device’s lifespan. Section VIII.IX.B. And a POSITA would have a reasonable expectation of success implementing this metadata sector in a flash device. Section VIII.IX.B.

280. Further, Sinclair-367 discloses dividing flash memory into planes, and forming metablocks by linking erase units in each plane. EX1009, ¶¶[0009]-[0012], [0111]-[0112]. This improve speed and throughput. Each plane operates independently—concurrently executing reading/writing. EX1009, ¶[0111]. And “to increase ... parallelism[,] erase blocks ... are logically linked to form metablocks.” EX1009, ¶[0112]. Because Sinclair-367 and Ban disclose analogous flash systems, and planes and metablocks are common in the flash memory art, a POSITA would have a reasonable likelihood of success implementing them in Ban’s flash device.

281. Ban, ZBC, and Sinclair-367 are analogous art—relating to storage devices, including techniques for accessing blocks of memory.

2. Claim 1, Limitation 1[pre]: *A storage device comprising:*

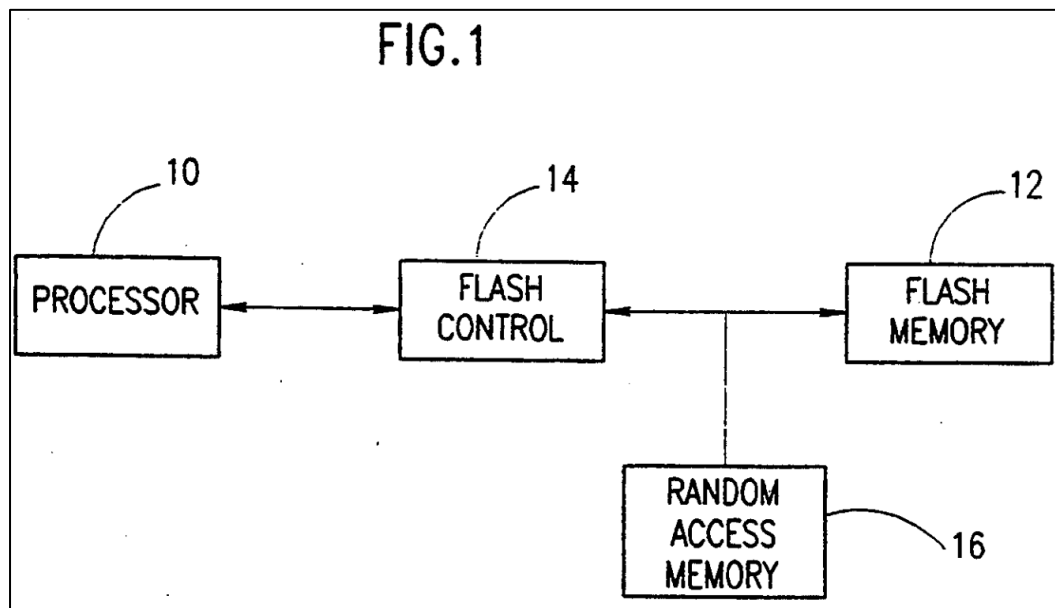
282. To the extent the preamble is limiting, Ban discloses it. Specifically, Ban discloses a flash device (*i.e.*, ***storage device***) with “flash memory controller 14,” “flash memory 12,” and “random access memory 16.” EX1005, 3:59-4:7, FIG. 1:



283. It is my opinion that Ban renders this limitation obvious.

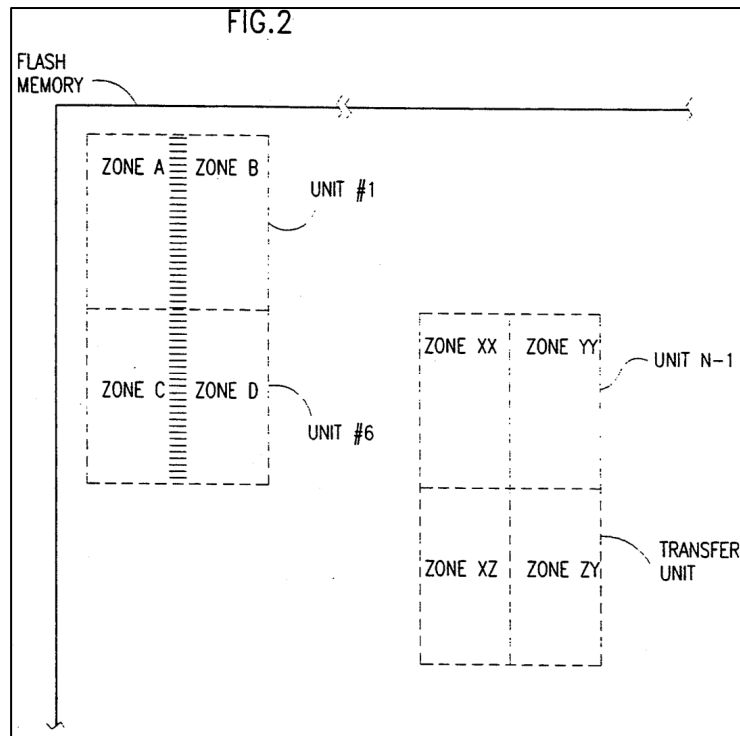
(a) **Limitation 1[a]:** *flash memory having physical erase units; and*

Ban's flash device includes *flash memory* 12. EX1005, 3:59-4:7, FIG. 1:



284. Flash memory 12 broken up into “zones” or “unit[s]” (*i.e., physical erase units*).

EX1005, 1:15-20, 4:11-14, FIG. 2:



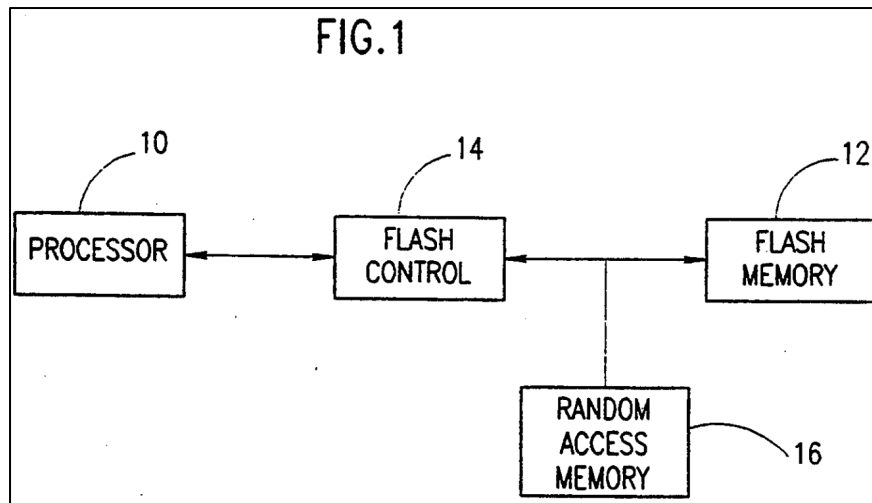
285. Zones and units are interchangeable as “[e]ach unit is comprised of at least one zone.” EX1005, 4:20-24. “Each unit contains blocks,” (EX1005, 2:11-19.) and a “byte[s] ... form a block.” EX1005, 2:11-15. Reading/writing is applied to bytes. EX1005, 2:11-15, 2:24-30. Erasing is performed with a “zone erase operation,” erasing every block in the zone. EX1005, 2:15-18, 2:67-3:2, 4:38-43.

286. It is my opinion that Ban renders this limitation obvious.

(b) **Limitation 1[b]:** *circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:*

287. Ban’s flash device includes flash controller 14 (*i.e.*, **circuitry**). EX1005, 3:59-4:7,

FIG. 1:



288. “[P]rocessor 10 ... issues ... read and write commands to ... specific address locations in a random access memory” (*i.e., incoming data access requests accompanying address information*). EX1005, 3:52-56. “[F]lash memory controller 14 provides a ... virtual address space” to “emulate[] ... random access memory” (*i.e., receive*). EX1005, 3: 64-4:5, FIG. 2.

289. It is my opinion that the combined system of Ban, ZBC, SBC-4, and SPC-4 renders this limitation obvious.

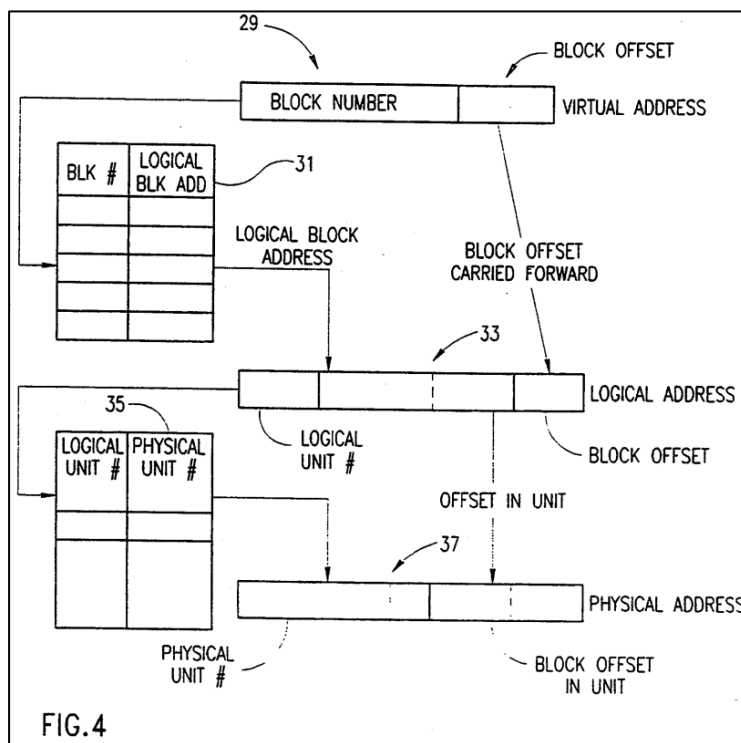
(c) Limitation 1[b][i]: *derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,*

290. From a ZBC read command (*i.e., accompanying incoming address information*), flash controller 14 **derives** (1) a number for a logical unit, volume, or partition (hereinafter “partitions”) (*i.e., a first address portion*), (2) virtual address 29 (*i.e., second address portion*), and (3) logical address 33 (*i.e., third address portion*).

291. **First Address Portion:** A POSITA would be motivated to incorporate a partition number in ZBC commands, and motivated to modify flash controller 14 to decode (*i.e., derive*) that number (*i.e., first address portion*) from the command (*i.e., from the accompanying address*

information). See Section VIII.C.1.d.

292. **Second Address Portion and Third Address Portion:** A POSITA would be motivated to use Ban’s flash controller 14 to *derive* virtual address 29 (*i.e.*, *second address portion*), which is further translated to logical address 33 (*i.e.*, *third address portion*), from the ZBC read command’s LBA (*i.e.*, *from the accompanying address information*). A ZBC read command includes an LBA. EX1007, Table 63. Flash controller 14 receives read commands targeting specific addresses—“computer generated addresses 29” (*i.e.*, “virtual address 29”). EX1005, 2:24-26, 2:31-33, 4:54-56, FIG 4 (below). Virtual address 29 is translated by flash controller 14 into logical address 33. EX1005, 5:2-12, FIG. 4:



293. It is my opinion that the combined system of Ban, ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(d) **Limitation 1[b][ii]:** *identify an addressed block device from the first address portion,*

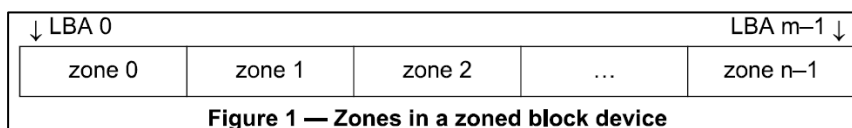
294. The partition number (*i.e.*, *from the first address portion*) *identif[ies]* a logical unit

(i.e., *addressed block device*). A logical unit has an independent range of LBAs starting at zero. EX1008, 61.

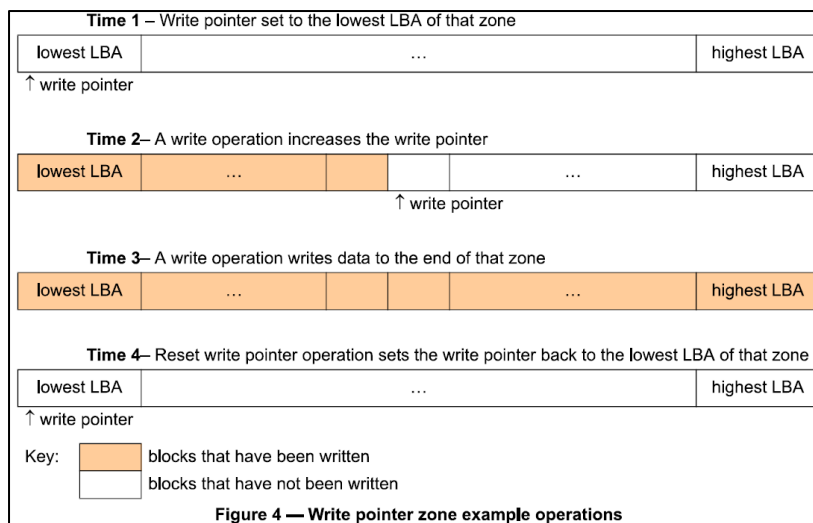
295. It is my opinion that the combined system of Ban, ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(e) **Limitation 1[b][iii]: identify an addressed segment within the addressed block device from the second address portion, and**

296. A POSITA would be motivated to modify Ban’s flash controller 14 to *identify* a zone number (i.e., *addressed segment*) from virtual address 29 (i.e., *second address portion*). A zone number is the zone that contains the ZBC command’s LBA—virtual address 29. EX1006, 13, Figure 1:



297. Identifying a zone number is helpful for pulling and/or updating necessary zone metadata. For instance, a write command changes a zone’s write pointer. EX1006, FIGS. 5-6, FIG. 4:



298. For a read command, zone metadata determines whether an LBA has been written since the last reset write pointer command, where, if not, the “manufacturer’s default initialization

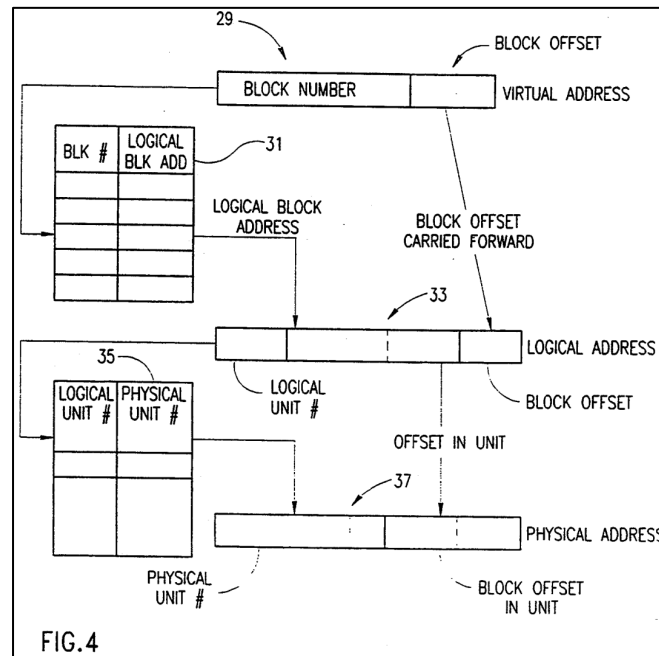
pattern” is returned. EX1006, 16-17.

299. A zone constitutes *an addressed segment* and is *within* the targeted logical unit (*i.e., addressed block device*). See Section VIII.C.1.f.

300. It is my opinion that the combined system of Ban, ZBC, SBC-4, and SPC-4 renders this limitation obvious.

(f) **Limitation 1[b][iv]: identify an addressed storage location within the addressed segment from the third address portion,**

301. Logical address 33 is translated to (*i.e., from the third address portion*) to physical address 37 (*i.e., identify an addressed storage location*). EX1005, 5:13-24, 5:29-35, FIG. 4:



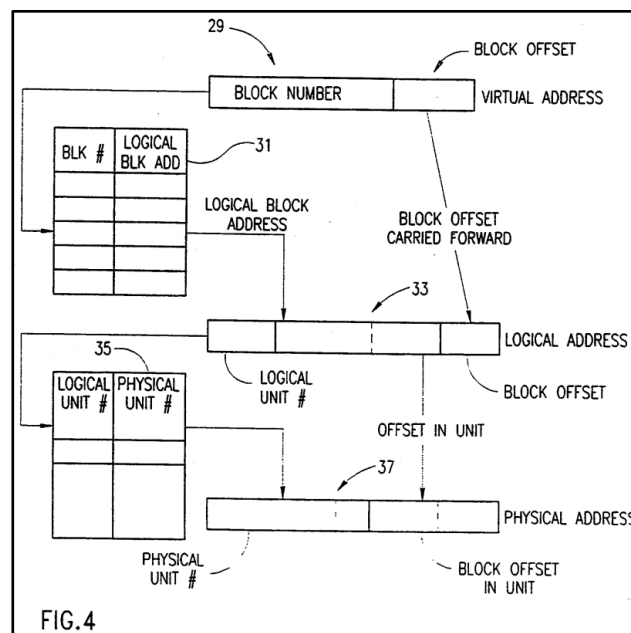
302. Physical address 37 is *within* the zone (*i.e., addressed segment*) because physical address 37 is derived from virtual address 29—ZBC command’s LBA—which falls within a zone. Limitation 1[b][iii]; EX1005, FIG. 4.

303. It is my opinion that the combined system of Ban, ZBC, SBC-4, and SPC-4 renders this limitation obvious.

- (g) **Limitation 1[c]:** *wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,*

304. Flash controller 14 (*i.e.*, **circuitry**) *uses* bit shifting (*i.e.*, **division operation**), *to identify* the logical unit number (*i.e.*, **logical erase unit**). And the logical unit number is derived from virtual address 29—ZBC command’s LBA—and therefore *within* the zone (*i.e.*, **addressed segment**). Section VII.D.2.e; EX1005, FIG. 4.

305. Ban discloses that the “logical unit number is the high order binary digits of the logical address and may be derived from the logical address by a bit shift operation.” EX1005, 4:67-5:12, FIG. 4:



306. The logical unit number is a **logical erase unit**. This is explicit in the name—“**logical unit** number.” Logical unit number is a logical unit. Unit, or zone, refers to an erase unit. Section VII.D.2.a.

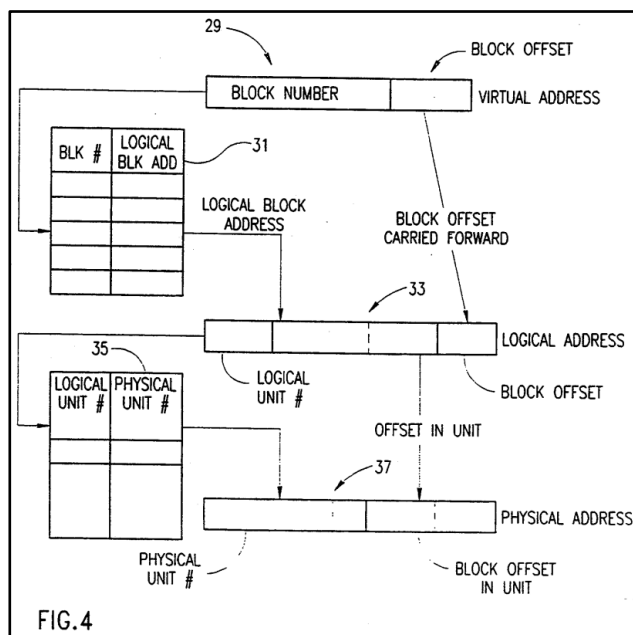
307. A logical unit number is an abstraction of a physical unit. It maps to the physical unit number. *See* EX1005, 5:13-17, FIG. 4.

308. It is my opinion that the combined system of Ban, ZBC, SBC-4, and SPC-4 renders

this limitation obvious.

- (h) **Limitation 1[d]:** *wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;*

309. Flash controller 14 (*i.e., circuitry is to*) translates the logical unit number (*i.e., identified logical erase unit*) to a physical unit number (*i.e., identify one of the physical erase units that corresponds to the identified logical erase unit*). EX1005, 5:13-17, FIG. 4:



310. It is my opinion that Ban renders this limitation obvious.

- (i) **Limitation 1[e]:** *wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests.*

311. A POSITA would be motivated to modify Ban’s flash device (*i.e., storage device*) 14 to implement Sinclair-367’s bad block management. Section XI.B.2. Sinclair-367 discloses “identification of any defective cells within the erase block,” (*i.e., detect a failure condition of the physical erase units*) and *remapping* the “logical address” (*i.e., logical erase unit which*

corresponds to one of the physical erase units for which the failure condition is detected) with “the address of any substitute erase block in case the primary erase block is defective” (*i.e., instead correspond to one of the physical erase units for which the failure condition has not been detected*) for later use (*i.e., utilize the remapped correspondence to service one or more of the incoming data access requests*). EX1009, ¶¶[0107]-[0108].

312. It is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

3. Claim 2, Limitation 2[pre]: *The storage device of claim 1 wherein:*

313. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 2[a]: *said circuitry comprises at least one processor; and*

Limitation 2[b]: *the storage device further comprises instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor.*

314. A POSITA would understand that Ban’s flash controller 14 (*said circuitry*) comprises at least one processor and instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor. “[C]ontroller 14 functions may be carried out in software, firmware[,] or hardware.” EX1005, 3:52-4:10.

315. It is my opinion that Ban renders this limitation obvious.

4. Claim 3, Limitation 3[pre]: *The storage device of claim 1 wherein:*

316. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 3[a]: *the incoming data access requests are read requests received from a host;*

317. My opinion for Section VII.D.2.b applies to this limitation.

- (b) **Limitation 3[b]:** *the storage device can receive a write request from the host, wherein the circuitry is to, for the received write request, derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion, identify an addressed segment within the addressed block device from the second address portion, identify a corresponding logical erase unit from the third address portion and identify a physical erase unit which is mapped to the corresponding logical erase unit;*

318. A POSITA would be motivated to implement Ban's flash device's (*i.e.*, **storage device**) flash controller 14 (*i.e.*, **circuitry**) to *receive* and execute ZBC commands (*i.e.*, *from the host*), including write commands (*i.e.*, *write request*). Section XI.B; EX1007, Table 112:

Table 112 — WRITE (32) command									
Byte	Bit	7	6	5	4	3	2	1	0
0		OPERATION CODE (7Fh)							
1		CONTROL							
2		Reserved							
...									
5									
6		Reserved			GROUP NUMBER				
7		ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Bh)							
9									(LSB)
10		WRPROTECT			DPO	FUA	Reserved	Obsolete	Reserved
11		Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS							
...									
19									(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG							
...									
23									(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG							
25									(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK							
27									(LSB)
28	(MSB)	TRANSFER LENGTH							
...									
31									(LSB)

319. Flash device's translation of ZBC read command's address information also applies to a ZBC write command, (Section VII.D.2.c-f.) thereby disclosing this limitation.

320. It is my opinion that Ban renders this limitation obvious.

- (c) **Limitation 3[c]:** *the storage drive is to, in association with fulfillment of the received write request, identify a physical storage*

location in which associated write data has been stored and which is associated with the corresponding logical erase unit, and to transmit information representing an address value to the host corresponding to the identified physical storage location; and

Limitation 3[d]: *for one of the read requests which is directed to data which has been written into the flash memory pursuant to the received write request, at least one of the first address portion, the second address portion or the third address portion associated with the one of the read requests is dependent on the address value which has been transmitted to the host in association with the data which has been written into the flash memory pursuant to the received write request.*

321. Ban discloses translating a read request's LBA to "data pointer 423, which points to the physical address." EX1005, FIGs. 4-5. The same translation applies to write operations (*in association with fulfillment of the received write request, identify a physical storage location in which associated write data has been stored and which is associated with the corresponding logical erase unit*). EX1005, 5:36-6:2, FIG. 6; Sections VII.D.2.c-h.

322. ZBC write commands also cause flash system 100 (*storage drive, in association with fulfillment of the received write request*) to transmit information back to the host the upon command completion. Section VII.A.4.c.

323. A POSITA would be motivated to modify the transmission to include information identifying the completed command (*information representing an address value to the host corresponding to the identified physical storage location*) because SPC-4 discloses processing multiple commands in parallel. See EX1008, xl, 431, 438, 472, 867, Table A.1; Section VII.A.4.c.

324. A POSITA would be motivated to restrict subsequent read commands (*for one of the read requests*) targeting LBAs subject to pending write operations (*for one of the read requests which is directed to data which has been written into the flash memory pursuant to the received write request*) until after receiving a completion response of the pending write operations (*dependent on the address value which has been transmitted to the host in association with the*

data which has been written into the flash memory pursuant to the received write request).

Section VII.A.4.c. The underlying *third address portion* is translated from the LBA, (Section VII.D.2.c) and is therefore also *dependent on* the completion response.

325. It is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

5. Claim 4, Limitation 4[pre]: *The storage device of claim 1 wherein:*

326. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 4[a]: *the storage device is to receive a maintenance request from a host which addresses a block device and a segment within the block device addressed by the maintenance request; and*

327. A POSITA would be motivated to implement Ban's flash device (*i.e.*, *storage device*) to *receive* and execute ZBC commands. Section VII.D.2.b. ZBC includes a reset write pointer command, which teaches this limitation. Section VII.A.5.a.

328. It is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

(b) Limitation 4[b]: *the storage device comprises circuitry to erase, in association with execution of the maintenance request, one or more of the physical erase units which are mapped to the segment addressed by the maintenance request.*

329. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.A.5.b, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.A.5.b applies to this limitation.

6. Claim 5, Limitation 5[pre]: *The storage device of claim 1 wherein:*

330. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 5[a]:** *the flash memory is to have block devices, including the addressed block device; and*

331. A POSITA would understand that a ZBC read commands targets one of the logical units (*i.e., addressed block device*) on Ban's flash device's flash memory 12 (*i.e., flash memory is to have block devices*). Section VII.D.2.c.

332. It is my opinion that Ban renders this limitation obvious.

- (b) **Limitation 5[b]:** *in connection with a given one of the incoming data access requests, the circuitry is to:*

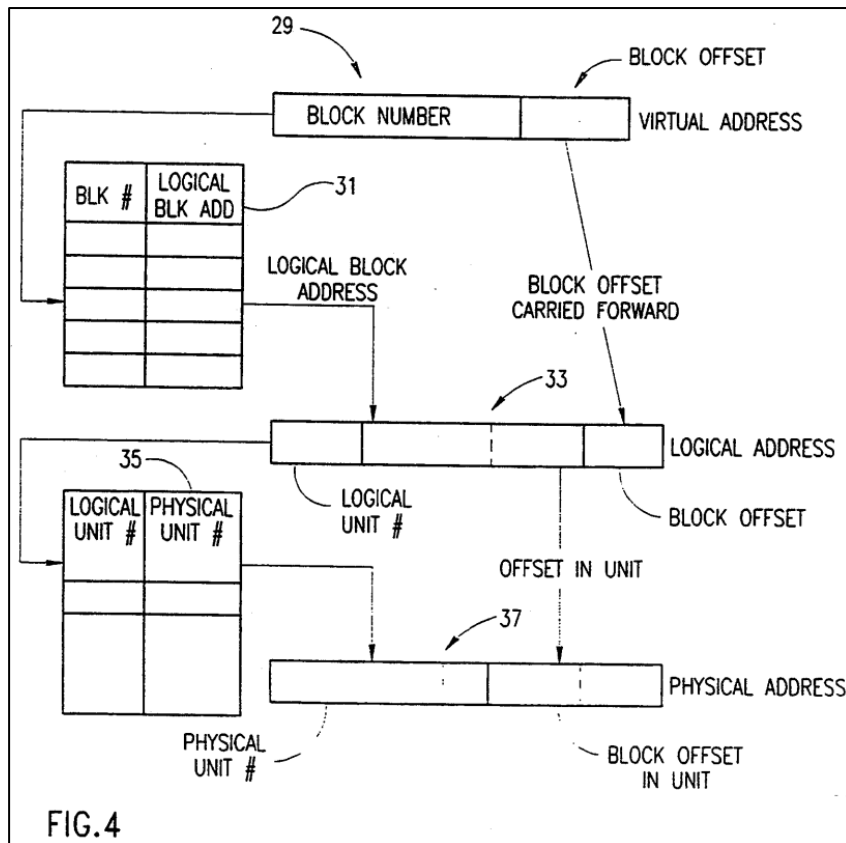
333. As addressed in Section VII.D.6.c, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

- (c) **Limitation 5[b][i]:** *identify and retrieve a look-up table specific to the addressed block device which was identified from the first address portion derived from the given one of the incoming data access requests, and*

Limitation 5[b][ii]: *identify a physical storage location in the flash memory dependent on the look-up table.*

334. Ban discloses translating a read command's virtual address 29 (*i.e., in connection with a given one of the incoming data access requests*) into logical address 33 and then a physical address with look-up tables 31 and 35 (*i.e., identify and retrieve a look-up table*) (*i.e., identify a physical storage location in the flash memory dependent on the look-up table*). EX1005, FIG.

4:



335. A POSITA would understand that the look-up tables 31 and 35 is specific to the ZBC read command's targeted logical unit (*addressed block device which was identified from the first address portion derived from the given one of the incoming access requests*). Logical units have overlapping logical address ranges, which means separate address translation structures should be used.

336. It is my opinion that Ban renders this limitation obvious.

7. Claim 6, Limitation 6[pre]: *The storage device of claim 1 wherein:*

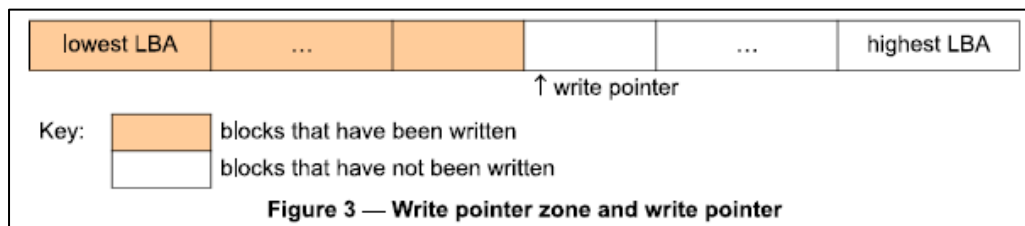
337. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 6[a]: *the storage device further comprises circuitry to track metadata for the addressed segment, where said metadata indicates an extent to which memory comprising at least one of the***

physical erase units mapped to the addressed segment can continue to receive additional write data without being erased; and

338. Ban’s flash device (*i.e.*, **storage device**) flash controller 14 (*i.e.*, **circuitry to** maintains a “usage map of all the blocks within [a] unit” (*i.e.*, **track metadata for the addressed segment**). EX1005, 2:57-60. The usage map includes “special characters to denote free blocks and to denote unusable blocks.” EX1005, 2:57-60. Because a unit is made up of blocks, showing the free and written blocks indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased.

339. Alternatively, Patent Owner’s infringement contentions assert that a zone’s write pointer discloses the limitation. EX1018, 63. ZBC also relies on write pointers for zones. EX1006, 14, 17, Figures 3:



(b) **Limitation 6[b]:** *the storage device further comprises circuitry to transmit information to a host dependent on said tracked metadata.*

340. A POSITA would be motivated to implement Ban’s flash device (*i.e.*, **storage device**) to execute ZBC commands. Section VII.D.2.b. ZBC includes a report zones command, (EX1006, Table 12.) which teaches this limitation. *See* Section VII.A.7.b.

341. It is my opinion that Ban renders this limitation obvious.

8. Claim 7, Limitation 7[pre]: *The storage device of claim 1 wherein:*

342. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 7[a]:** *the incoming data access requests are read requests received from a host;*

343. Given the similarity of claim language, my opinion for Section VII.D.4.a applies to this limitation.

- (b) **Limitation 7[b]:** *the storage device can receive write requests from the host, and the storage device comprises circuitry to issue commands to the flash memory in fulfillment of the write requests;*

344. Given the similarity of claim language, my opinion for Section VII.D.4.b-c applies to this limitation.

- (c) **Limitation 7[c]:** *the storage device is to assign an address value in association with fulfillment of each of the write requests, in order to identify a physical storage location in which associated write data has been stored, and comprises circuitry to transmit information representing each assigned address value to the host; and*

345. Given the similarity of claim language, my opinion for Section VII.D.4.c applies to this limitation.

- (d) **Limitation 7[d]:** *for one of the incoming data access requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, the addressed storage location identified from the third portion is dependent on the assigned address value which was transmitted to the host in association with the one of the write requests.*

346. Given the similarity of claim language, my opinion for Section VII.D.4.c applies to this limitation.

9. Claim 8, Limitation 8[pre]: *The storage device of claim 1 wherein:*

347. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 8[a]:** *the storage device can receive a request from a host to copy host-selected data from a first storage location in flash memory to a second storage location in the flash memory, and the storage device comprises circuitry to issue commands to the flash memory, in response to the request from the host to copy, to read*

the host-selected data from the first storage location in the flash memory and write the host-selected data to the second storage location in the flash memory without sending the data being copied to the host; and

348. A POSITA would be motivated to implement Ban's flash device (*i.e.*, **storage device**) to *receive* and execute ZBC commands. Section VII.D.2.b. ZBC includes an extended copy command (*i.e.*, **request from a host to copy**), (EX1006, 1, 22; EX1007, Table 33; EX1008, 287-89, Table 128.) which teaches his limitation. Section VIII.D.8.b, XI.B.

349. Given the disclosure of this limitation by ZBC's extended copy command in Section VII.A.9.a, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1) my opinion for Section VII.A.9.a applies to this limitation.

350. It is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

(b) **Limitation 8[b]:** *the request from the host to copy is to designate destination address information corresponding to the second storage location, wherein the storage device is to identify from the destination address information a destination block device and an addressed segment within the destination block device, and wherein the second storage location corresponds to the addressed segment within the destination block device.*

351. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.A.9.b, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1) my opinion for Section VII.A.9.b applies to this limitation.

10. Claim 9, Limitation 9[pre]: *The storage device of claim 8 wherein*

352. As addressed in Section VII.D.9, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) **Limitation 9[a]:** *the storage device is to identify from the destination address information a destination logical erase unit*

and is to select the second storage location in dependence on the identified destination logical erase unit.

353. A POSITA would be motivated to implement Ban's flash device (*i.e., storage device*) to perform its address translation on a ZBC extended copy command's LBA (*i.e., from the destination address information*) to *identify* a logical unit number (*i.e., destination logical erase unit*). Limitation 8[a]. Further translation, logical unit number (*i.e., in dependence on the identified destination logical erase unit*) *selects* physical unit number of the physical address (*i.e., the second storage location*). EX1005, FIG. 4.

354. It is my opinion that the combined system of Ban and ZBC, SBC-4, and SPC-4 renders this limitation obvious.

11. Claim 10, Limitation 10[pre]: *The storage device of claim 1 wherein*

355. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 10[a]:** *the storage device is to store information identifying a number of structures in the flash memory and is to transmit the information identifying the number of structures to a host, wherein the host is to issue at least one request to the storage device in which the at least one request is dependent on the information transmitted to the host identifying the number of structures, and wherein the storage device comprises circuitry to fulfill the at least one request by issuing commands to the flash memory for which a quantum of flash memory selected by the commands which are issued in response to the at least one request is dependent on the information transmitted to the host identifying the number of structures.*

356. ZBC's report zones command *transmits the information identifying the number of structures to a host*. Section VIII.D.10.b. A POSITA would understand *storing*, or be motivated to *store*, that metadata rather than wasting time re-calculating such information for every needing or requesting the metadata.

357. Given the disclosure of this limitation by ZBC's reset write pointer command in

Section VII.A.11.a, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.A.11.a applies to this limitation.

12. Claim 11, Limitation 11[pre]: *The storage device of claim 10 wherein*

358. As addressed in Section VII.D.11, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 11[a]: *the information transmitted to the host identifying the number of structures identifies an amount of flash memory that is to be erased as a unit.*

359. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.A.12.a, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.A.12.a applies to this limitation.

13. Claim 12, Limitation 12[pre]: *The storage device of claim 1 wherein*

360. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 12[a]: *the addressed segment corresponds to an integer number of physical erase units for which the failure condition has not been detected, and the third address portion specifies a location within a sequential range of locations that correspond to the integer number of physical erase units.*

361. A POSITA would be motivated to require that zones (*i.e., addressed segment*) contain a whole number of physical units (*i.e., corresponds to an integer number of physical erase units*) because a POSITA would also be motivated to implement the reset write pointer command to erase a zone. Section VII.D.5.b. If a zone contained a non-whole number of physical units, zone erasure erases units only partially within the zone. Because unit erasure erases everything in the unit, (EX1005, 2:38-43.) a reset write pointer command erases data outside the zone, risking data loss. This, or complexity to avoid data loss, is avoided by requiring whole units in each zone.

362. Logical address 33 (*i.e., third address portion*) *specifies a location within a sequential range of locations*. Logical address 33 includes a block offset—position within a logical unit’s sequential list of blocks. EX1005, 5:36-38. And because logical address 33 is derived from virtual address 29, which determines the zone, logical address 33 *specifies a location corresponding to the* zone’s whole number of units (*i.e., integer number of physical erase units*).

363. It is my opinion that Ban renders this limitation obvious.

14. Claim 13, Limitation 13[pre]: *The storage device of claim 1 wherein:*

364. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 13[a]: *the flash memory is to have block devices, including the addressed block device; and*

365. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.B.2.a, and how a POSITA would be motivated to incorporate ZBC’s teachings in Ban’s flash device, (Section VII.D.1) my opinion for Section VII.B.2.a applies to this limitation.

(b) Limitation 13[b]: *the storage device further comprises circuitry to:*

366. As addressed in Sections VII.D.14.c-f, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

(c) Limitation 13[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,*

367. Given the disclosure of this limitation by ZBC, SBC-4, SPC-4, and Sinclair-367 in Section VII.B.2.c, and how a POSITA would be motivated to incorporate ZBC’s teachings in Ban’s flash device, (Section VII.D.1) my opinion for Section VII.B.2.c applies to this limitation.

- (d) **Limitation 13[b][ii]:** *compare the metadata to at least one threshold,*

368. Given the disclosure of this limitation by ZBC, SBC-4, SPC-4, and Sinclair-367 in Section VII.B.2.d, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1) my opinion for Section VII.B.2.d applies to this limitation.

- (e) **Limitation 13[b][iii]:** *upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and*

369. Given the disclosure of this limitation by ZBC, SBC-4, SPC-4, and Sinclair-367 in Section VII.B.2.e, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.B.2.e applies to this limitation.

- (f) **Limitation 13[b][iv]:** *remap the logical erase unit corresponding to the first one of the physical erase units for which the failure condition has not been detected by said storage device to the second one of the physical erase units for which the failure condition has not been detected by said storage device.*

370. Given the disclosure of this limitation by ZBC, SBC-4, SPC-4, and Sinclair-367 in Section VII.B.2.f, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.B.2.f applies to this limitation.

15. Claim 14, Limitation 14[pre]: *The storage device of claim 1 wherein:*

371. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 14[a]:** *the flash memory is to have block devices, including the addressed block device; and*

372. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.B.3.a, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash

device, (Section VII.D.1.) my opinion for Section VII.B.3.a applies to this limitation.

(b) Limitation 14[b]: *the storage device further comprises circuitry to:*

373. As addressed in Sections VII.D.15.c-d, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

(c) Limitation 14[b][i]: *store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata is dependent on an erase count of associated physical memory space,*

374. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.B.3.c, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.B.3.c applies to this limitation.

(d) Limitation 14[b][ii]: *compare the metadata to at least one threshold,*

Limitation 14[b][iii]: *upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and*

Limitation 14[b][iv]: *remap the logical erase unit corresponding to the first physical erase unit for which the failure condition has not been detected by said storage device to the second physical erase unit for which the failure condition has not been detected by said storage device.*

375. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.B.3.d, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.B.3.d applies to this limitation.

16. Claim 16, Limitation 16[pre]: *The storage device of claim 1 wherein:*

376. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 16[a]:** *the flash memory is to have block devices, including the addressed block device; and*

377. Given the similarities in claim language, my opinion for Section VII.D.6.a applies to this limitation.

- (b) **Limitation 16[b]:** *the storage device further comprises circuitry to:*

378. As addressed in Sections VII.D.16.c-d, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 renders this limitation obvious.

- (c) **Limitation 16[b][i]:** *store metadata on an independent basis for flash memory associated with each one of the block devices, and*

379. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.A.14.c, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.A.14.c applies to this limitation.

- (d) **Limitation 16[b][ii]:** *receive a query, issued by a host device, and responsive to the query, compare the metadata to at least one threshold, to identify specific metadata, and transmit information to the host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.*

380. Given the disclosure of this limitation by ZBC, SBC-4, and SPC-4 in Section VII.A.14.d, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.D.1.) my opinion for Section VII.A.14.d applies to this limitation.

17. Claim 19, Limitation 19[pre]: *The storage device of claim 1 wherein:*

381. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 19[a]:** *the flash memory comprises multiple planes; and*

382. A POSITA would be motivated to implement Ban's *flash memory* 12 as "divided into two or more planes" (*i.e.*, *multiple planes*) per Sinclair-367's teachings. EX1009, ¶[0111];

Section VII.D.1.

383. It is my opinion that the combined system of Ban and Sinclair-367 renders this limitation obvious.

- (b) **Limitation 19[b]:** *the storage device is to, for each one of a set of the incoming data access requests having sequential addressing, identify a commonly addressed page and write data accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes.*

384. A ZBC write command includes a “logical block address” and “transfer length.”
EX1007, Table 112.

Table 112 — WRITE (32) command

Byte	Bit	7	6	5	4	3	2	1	0
0		OPERATION CODE (7Fh)							
1		CONTROL							
2		Reserved							
...									
5									
6		Reserved			GROUP NUMBER				
7		ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Bh)							
9									
10		WRPROTECT			DPO	FUA	Reserved	Obsolete	Reserved
11		Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS							
...									
19		(LSB)							
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG							
...									
23		(LSB)							
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG							
25									
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK							
27									
28	(MSB)	TRANSFER LENGTH							
...									
31		(LSB)							

385. “TRANSFER LENGTH ... specifies the ... logical blocks ... that shall be transferred ... starting [from] ... the LOGICAL BLOCK ADDRESS ...” EX1007, 198. With a large transfer length, a ZBC write command causes several write operations across *sequential*

addresses.

386. To increase throughput for these large write commands, a POSITA would be motivated to implement Ban’s *flash memory* 12 with multiple *planes*, and metablocks spanning the multiple planes, as disclosed in Sinclair-367. EX1009, ¶¶[0111]-[0112]; Sections VII.D.1, VII.D.17.a. “All blocks of the metablock are erased together, and pages of each block are generally programmed and read simultaneously” (*i.e., identify a commonly addressed page and write data accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes*). EX1009, ¶[0112].

387. It is my opinion that Ban renders this limitation obvious.

18. Claim 20, Limitation 20[pre]: *The storage device of claim 1 wherein*

388. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

(a) Limitation 20[a]: *the division operation is applied to subdivide the third address portion into respective first and second bit fields.*

389. It is my opinion that the analysis in Section VII.D.2.c applies to this limitation.

19. Claims 21-23

390. Claims 21-23 are substantively identical to claim 1. Claim 21 is substantively identical to claim 1. Section VII.A.16. Claim 21’s additional elements map to limitations 1[b][iii]-[d] which require “*block device[s]*,” “*segment[s] within the ... block device[s]*,” “*erase unit[s] within the ... segment[s]*,” and “*storage location[s] within the ... segment[s]*.” Moreover, the additional elements of claim 21 are disclosed in Ground IV—logical units containing zones, which contain logical blocks and pages. EX1005, FIG. 4 (translating an LBA to a zone number and offset number); EX1006, 13-14, Figures 1-2; EX1008, 14, 61 (“[L]ogical block addresses on a logical unit ... shall begin with block zero and be contiguous up to the last logical block of that logical

unit”); Sections VII.D.2.c-e. Claim 22 is substantively identical to claim 21 incorporated with dependent claim 3. EX1003, ¶¶209-10. Claim 23 is substantively identical to claim 1. claim 1 is directed to “[a] storage device,” whereas claim 23 is directed to “[a] method of operation in a storage device.” This distinction is immaterial since Ban discloses systems and steps in operating those systems. E.g., EX1005, FIGs. 4-5.

391. It is my opinion that Ban renders claims 21-23 obvious.

E. GROUND V: CLAIMS 15, 17-18 ARE RENDERED OBVIOUS BY BAN IN LIGHT OF ZBC, SBC-4, SPC-4, SINCLAIR-367, AND YAMADA

1. Motivation To Combine Yamada With The Combined System Of Ban, ZBC, SBC-4, SPC-4, And Sinclair-367

392. A POSITA would have been motivated to implement Ban’s flash device with Yamada’s maintenance request and determination units in Ban’s flash device and host processor, respectively. Ban mentions the necessity, but with little description, of a maintenance system, stating that “enabl[ing] reading and writing operations to continue without limitation, physical memory space is reclaimed periodically.” EX1005, 6:3-5. Yamada discloses a maintenance system, where “maintenance is performed without being ... performed together with normal system processing,” but “with sufficiently long processing time.” EX1010, Abstract, ¶¶[0018]-[0019], [0022]. Because Yamada and Ban implement analogous nonvolatile systems, a POSITA would have a reasonable expectation of success implementing Ban’s flash device with this scheme.

393. Moreover, Ban, ZBC, Sinclair-367, and Yamada disclose analogous art. All relate to storage devices.

2. Claim 15, Limitation 15[pre]: *The storage device of claim 1 wherein:*

394. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 15[a]:** *the flash memory is to have block devices, including the addressed block device; and*

395. Given the similarities in claim language, my opinion for Section VII.D.6.a applies to this limitation.

- (b) **Limitation 15[b]:** *the storage device further comprises circuitry to:*

396. As addressed in Sections VII.E.2.c-e, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, Sinclair-367, and Yamada renders this limitation obvious.

- (c) **Limitation 15[b][i]:** *store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,*

397. Given the similarities in claim language, my opinion for Section VII.D.14.c applies to this limitation.

- (d) **Limitation 15[b][ii]:** *compare the metadata to at least one threshold, and*

398. Given the similarities in claim language, my opinion for Section VII.D.14.d applies to this limitation.

- (e) **Limitation 15[b][iii]:** *responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.*

399. Given the disclosure of this limitation by Yamada in Section VII.C.2.e, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.E.1.) my opinion for Section VII.C.2.e applies to this limitation.

3. Claim 17, Limitation 17[pre]: *The storage device of claim 1 wherein:*

400. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban,

ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 17[a]:** *the flash memory is to have block devices, including the addressed block device; and*

401. Given the similarities in claim language, my opinion for Section VII.D.16.a applies to this limitation.

- (b) **Limitation 17[b]:** *the storage device further comprises circuitry to:*

402. As addressed in Sections VII.E.3.c-d, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, Sinclair-367, and Yamada renders this limitation obvious.

- (c) **Limitation 17[b][i]:** *store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and*

403. Given the similarities in claim language, my opinion for Section VII.D.15.c applies to this limitation.

- (d) **Limitation 17[b][ii]:** *receive a query, issued by a host device, wherein said query is specific to one of the block devices, and responsive to the query, identify specific metadata, associated with the specified one of the block devices, and transmit information to the host device that conveys the specific metadata.*

404. Given the disclosure of this limitation by Yamada in Section VII.C.3.d, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.E.1.) my opinion for Section VII.C.3.d applies to this limitation.

4. Claim 18, Limitation 18[pre]: *The storage device of claim 1 wherein:*

405. As addressed in Section VII.D.2, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, and Sinclair-367 disclose this limitation.

- (a) **Limitation 18[a]:** *the flash memory is to have block devices, including the addressed block device; and*

406. Given the similarities in claim language, my opinion for Section VII.D.6.a applies

to this limitation.

- (b) **Limitation 18[b]:** *the storage device further comprises circuitry to:*

407. As addressed in Sections VII.E.4.c-e, it is my opinion that the combined system of Ban, ZBC, SBC-4, SPC-4, Sinclair-367, and Yamada renders this limitation obvious.

- (c) **Limitation 18[b][i]:** *store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and*

408. Given the similarities in claim language, my opinion for Section VII.D.15.c applies to this limitation.

- (d) **Limitation 18[b][ii]:** *compare the metadata to at least one threshold, and*

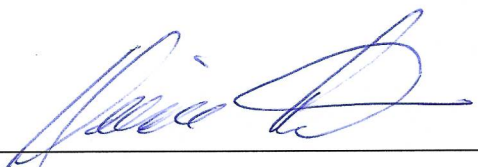
409. Given the similarities in claim language, my opinion for Section VII.D.15.d applies to this limitation.

- (e) **Limitation 18[b][iii]:** *responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.*

410. Given the disclosure of this limitation by Yamada in Section VII.C.4.e, and how a POSITA would be motivated to incorporate ZBC's teachings in Ban's flash device, (Section VII.E.1.) my opinion for Section VII.C.4.e applies to this limitation.

I, Heiner Litz, declare under penalty of perjury under the laws of the United States that the foregoing is true and correct.

Dated: August 5, 2025



Heiner Litz, Ph.D.

CLAIM LISTING

Claim 1	
1[pre]	A storage device comprising:
1[a]	flash memory having physical erase units; and
1[b]	circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:
1[b][i]	derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,
1[b][ii]	identify an addressed block device from the first address portion,
1[b][iii]	identify an addressed segment within the addressed block device from the second address portion, and
1[b][iv]	identify an addressed storage location within the addressed segment from the third address portion,
1[c]	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,
1[d]	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;
1[e]	wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests.
Claim 2	
2[pre]	The storage device of claim 1 wherein:
2[a]	said circuitry comprises at least one processor; and
2[b]	the storage device further comprises instructions stored on at least one physical storage medium, said instructions when executed to control the at least one processor.
Claim 3	
3[pre]	The storage device of claim 1 wherein:

3[a]	the incoming data access requests are read requests received from a host;
3[b]	the storage device can receive a write request from the host, wherein the circuitry is to, for the received write request, derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion, identify an addressed segment within the addressed block device from the second address portion, identify a corresponding logical erase unit from the third address portion and identify a physical erase unit which is mapped to the corresponding logical erase unit;
3[c]	the storage drive is to, in association with fulfillment of the received write request, identify a physical storage location in which associated write data has been stored and which is associated with the corresponding logical erase unit, and to transmit information representing an address value to the host corresponding to the identified physical storage location; and
3[d]	for one of the read requests which is directed to data which has been written into the flash memory pursuant to the received write request, at least one of the first address portion, the second address portion or the third address portion associated with the one of the read requests is dependent on the address value which has been transmitted to the host in association with the data which has been written into the flash memory pursuant to the received write request.
Claim 4	
4[pre]	The storage device of claim 1 wherein:
4[a]	the storage device is to receive a maintenance request from a host which addresses a block device and a segment within the block device addressed by the maintenance request; and
4[b]	the storage device comprises circuitry to erase, in association with execution of the maintenance request, one or more of the physical erase units which are mapped to the segment addressed by the maintenance request.
Claim 5	
5[pre]	The storage device of claim 1 wherein:
5[a]	the flash memory is to have block devices, including the addressed block device; and
5[b]	in connection with a given one of the incoming data access requests, the circuitry is to:

5[c]	identify and retrieve a look-up table specific to the addressed block device which was identified from the first address portion derived from the given one of the incoming data access requests, and
5[d]	identify a physical storage location in the flash memory dependent on the look-up table.
Claim 6	
6[pre]	The storage device of claim 1 wherein:
6[a]	the storage device further comprises circuitry to track metadata for the addressed segment, where said metadata indicates an extent to which memory comprising at least one of the physical erase units mapped to the addressed segment can continue to receive additional write data without being erased; and
6[b]	the storage device further comprises circuitry to transmit information to a host dependent on said tracked metadata.
Claim 7	
7[pre]	The storage device of claim 1 wherein:
7[a]	the incoming data access requests are read requests received from a host;
7[b]	the storage device can receive write requests from the host, and the storage device comprises circuitry to issue commands to the flash memory in fulfillment of the write requests;
7[c]	the storage device is to assign an address value in association with fulfillment of each of the write requests, in order to identify a physical storage location in which associated write data has been stored, and comprises circuitry to transmit information representing each assigned address value to the host; and
7[d]	for one of the incoming data access requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, the addressed storage location identified from the third portion is dependent on the assigned address value which was transmitted to the host in association with the one of the write requests.
Claim 8	
8[pre]	The storage device of claim 1 wherein:
8[a]	the storage device can receive a request from a host to copy host-selected data from a first storage location in flash memory to a second storage location in the flash memory, and the storage device comprises circuitry to issue commands to the flash memory, in response to the request from the host to copy, to read the host-selected

	data from the first storage location in the flash memory and write the host-selected data to the second storage location in the flash memory without sending the data being copied to the host; and
8[b]	the request from the host to copy is to designate destination address information corresponding to the second storage location, wherein the storage device is to identify from the destination address information a destination block device and an addressed segment within the destination block device, and wherein the second storage location corresponds to the addressed segment within the destination block device.
Claim 9	
9[pre]	The storage device of claim 8 wherein
9[a]	the storage device is to identify from the destination address information a destination logical erase unit and is to select the second storage location in dependence on the identified destination logical erase unit.
Claim 10	
10[pre]	The storage device of claim 1 wherein
10[a]	the storage device is to store information identifying a number of structures in the flash memory and is to transmit the information identifying the number of structures to a host, wherein the host is to issue at least one request to the storage device in which the at least one request is dependent on the information transmitted to the host identifying the number of structures, and wherein the storage device comprises circuitry to fulfill the at least one request by issuing commands to the flash memory for which a quantum of flash memory selected by the commands which are issued in response to the at least one request is dependent on the information transmitted to the host identifying the number of structures.
Claim 11	
11[pre]	The storage device of claim 10 wherein
11[a]	the information transmitted to the host identifying the number of structures identifies an amount of flash memory that is to be erased as a unit.
Claim 12	
12[pre]	The storage device of claim 1 wherein
12[a]	the addressed segment corresponds to an integer number of physical erase units for which the failure condition has not been detected, and the third address portion

	specifies a location within a sequential range of locations that correspond to the integer number of physical erase units.
Claim 13	
13[pre]	The storage device of claim 1 wherein:
13[a]	the flash memory is to have block devices, including the addressed block device; and
13[b]	the storage device further comprises circuitry to:
13[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,
13[d]	compare the metadata to at least one threshold,
13[e]	upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and
13[f]	remap the logical erase unit corresponding to the first one of the physical erase units for which the failure condition has not been detected by said storage device to the second one of the physical erase units for which the failure condition has not been detected by said storage device.
Claim 14	
14[pre]	The storage device of claim 1 wherein:
14[a]	the flash memory is to have block devices, including the addressed block device; and
14[b]	the storage device further comprises circuitry to:
14[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata is dependent on an erase count of associated physical memory space,
14[d]	compare the metadata to at least one threshold,
14[e]	upon detection that specific metadata has satisfied the one or more thresholds, copy data stored at one of the one or more physical storage locations associated with the

	specific metadata from a first one of the physical erase units for which the failure condition has not been detected by said storage device to a second one of the physical erase units for which the failure condition has not been detected by said storage device, and
14[f]	remap the logical erase unit corresponding to the first physical erase unit for which the failure condition has not been detected by said storage device to the second physical erase unit for which the failure condition has not been detected by said storage device.
Claim 15	
15[pre]	The storage device of claim 1 wherein:
15[a]	the flash memory is to have block devices, including the addressed block device; and
15[b]	the storage device further comprises circuitry to:
15[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, the metadata representing a time since one or more physical storage locations associated with the one of the block devices has been a subject of a write operation,
15[d]	compare the metadata to at least one threshold, and
15[e]	responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.
Claim 16	
16[pre]	The storage device of claim 1 wherein:
16[a]	the flash memory is to have block devices, including the addressed block device; and
16[b]	the storage device further comprises circuitry to:
16[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, and
16[d]	receive a query, issued by a host device, and responsive to the query, compare the metadata to at least one threshold, to identify specific metadata, and transmit information to the host device wherein such transmitted information conveys

	address information for data stored at one of the one or more physical storage locations associated with the specific metadata.
Claim 17	
17[pre]	The storage device of claim 1 wherein:
17[a]	the flash memory is to have block devices, including the addressed block device; and
17[b]	the storage device further comprises circuitry to:
17[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and
17[d]	receive a query, issued by a host device, wherein said query is specific to one of the block devices, and responsive to the query, identify specific metadata, associated with the specified one of the block devices, and transmit information to the host device that conveys the specific metadata.
Claim 18	
18[pre]	The storage device of claim 1 wherein:
18[a]	the flash memory is to have block devices, including the addressed block device; and
18[b]	the storage device further comprises circuitry to:
18[c]	store metadata on an independent basis for flash memory associated with each one of the block devices, wherein the stored metadata represents an erase count of associated physical memory space, and
18[d]	compare the metadata to at least one threshold, and
18[e]	responsive to detection that specific metadata has satisfied the one or more thresholds, transmit information to a host device wherein such transmitted information conveys address information for data stored at one of the one or more physical storage locations associated with the specific metadata.
Claim 19	
19[pre]	The storage device of claim 1 wherein:
19[a]	the flash memory comprises multiple planes; and
19[b]	the storage device is to, for each one of a set of the incoming data access requests having sequential addressing, identify a commonly addressed page and write data

	accompanying the set of incoming data access requests concurrently into respective ones of the multiple planes.
Claim 20	
20[pre]	The storage device of claim 1 wherein
20[a]	the division operation is applied to subdivide the third address portion into respective first and second bit fields.
Claim 21	
21[pre]	A storage device comprising:
21[a]	flash memory, the flash memory to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment including a mutually exclusive set of one or more logical erase units that each map to a physical erase unit, each segment being associated with only one of the block devices, each logical erase unit being associated with only one of the segments; and
21[b]	circuitry to receive incoming data access requests and for each of the incoming data access requests, accompanying address information, wherein said circuitry is to:
21[b][i]	derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,
21[b][ii]	identify an addressed block device from the first address portion,
21[b][iii]	identify an addressed segment within the addressed block device from the second address portion, and
21[b][iv]	identify an addressed storage location within the addressed segment from the third address portion,
21[c]	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,
21[d]	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;
21[e]	wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and

	is to utilize the remapped correspondence to service one or more of the incoming data access requests.
Claim 22	
22[pre]	A storage device comprising:
22[a]	flash memory, the flash memory to have one or more block devices at a hierarchically-superior level, segments within each of the block devices at a hierarchically-intermediate level, and addressable storage locations within each of the segments at a hierarchically-inferior level, each segment including a mutually exclusive set of one or more logical erase units that each map to a physical erase unit, each segment being associated with only one of the block devices, each logical erase unit being associated with only one of the segments;
22[b]	circuitry to receive incoming data read requests and incoming data write requests and, for each of the incoming data read requests and incoming data write requests, accompanying address information, wherein said circuitry is to, for each of the incoming data read requests:
22[b][i]	derive, from the accompanying address information, a first address portion, a second address portion and a third address portion,
22[b][ii]	identify an addressed block device from the first address portion,
22[b][iii]	identify an addressed segment within the addressed block device from the second address portion, and
22[b][iv]	identify an addressed storage location within the addressed segment from the third address portion,
22[c]	wherein the circuitry is to use a division operation to identify a logical erase unit within the addressed segment,
22[d]	wherein the circuitry is to identify one of the physical erase units that corresponds to the identified logical erase unit;
22[e]	wherein the storage device is to detect a failure condition of the physical erase units and is to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and is to utilize the remapped correspondence to service one or more of the incoming data access requests; and
22[f]	wherein said circuitry is to, for each of the incoming data write requests:

22[f][i]	derive, from the accompanying address information, an associated first address portion, an associated second address portion and an associated third address portion,
22[f][ii]	identify an addressed block device from the associated first address portion,
22[f][iii]	identify an addressed segment within the addressed block device from the associated second address portion,
22[f][iv]	identify a write destination logical erase unit from the associated third address portion, and identify a physical erase unit which is mapped to the write destination logical erase unit,
22[f][v]	identify a physical storage location in which associated write data is to be stored and which is associated with the write destination logical erase unit, and an address value associated with the physical storage location in which the associated write data is to be stored; and
22[f][vi]	transmit information to a host representing the address value;
22[g]	wherein for one of the read requests which is directed to data which has been written into the flash memory pursuant to one of the write requests, at least one of the first address portion, the second address portion or the third address portion associated with the one of the read requests is dependent on the address value which has been transmitted to the host in association with the data which has been written into the flash memory pursuant to one of the write requests.
Claim 23	
23[pre]	A method of operation in a storage device having flash memory, the flash memory having physical erase units, and circuitry, the method comprising:
23[a]	receiving incoming data access requests with the circuitry, wherein each of the incoming data access requests is accompanied by address information;
23[b]	with the circuitry, processing the incoming address information to, for each of the incoming data access requests:
23[b][i]	derive a first address portion, a second address portion and a third address portion, identify an addressed block device from the first address portion,
23[b][ii]	identify an addressed segment within the addressed block device from the second address portion,
23[b][iii]	identify an addressed storage location within the addressed segment from the third address portion,

23[b][iv]	perform a division operation to identify a logical erase unit within the addressed segment, and
23[b][v]	identify one of the physical erase units that corresponds to the identified logical erase unit;
23[c]	wherein the method further comprises causing the storage device to detect a failure condition of the physical erase units, to remap each logical erase unit which corresponds to one of the physical erase units for which the failure condition is detected to instead correspond to one of the physical erase units for which the failure condition has not been detected and to utilize the remapped correspondence to service one or more of the incoming data access requests.