

Exhibit E

EXHIBIT E

Invalidity Claim Chart for U.S. Patent No. 11,830,546 Based on U.S. Patent No. US 8,634,240 to Gavens et al. (“Gavens”)

Claim Language	Citations to Prior Art
[1pre] A system for storing data comprising:	See Exhibit A1 ('298 patent) claim element 1pre.
[1a] memory space containing volatile memory space and nonvolatile memory space, wherein the nonvolatile memory space includes both multi-level cell (MLC) space and single level cell (SLC) space;	<p>Gavens discloses memory space containing volatile memory space and nonvolatile memory space, wherein the nonvolatile memory space includes both multi-level cell (MLC) space and single level cell (SLC) space.</p> <p>For example, Gavens discloses memory containing volatile RAM and non-volatile memory array that includes a first portion of SLC memory and a second portion of MLC memory.</p> <p>Col. 1:26: RAM.</p> <p>Col. 1:18: Non-volatile memory.</p> <p>Fig. 1: Memory array 200 in memory chip 100.</p> <p>Col. 4:32-34, 17:21-23: The first portion has each memory cell storing one bit of data and the second portion has each memory cell storing more than one bit of data.</p> <p>Col. 5:15-17, 50-53, 16:22-24, 20:31-34: The high density storage portion (D3) has each memory cell storing 3 bits of data. The low density storage portion (D1) has each memory cell storing 1 bit of data.</p> <p>Fig. 14A, col. 16:15-22: The array of memory cells 200 is partitioned into a first portion 410 of lower density and a second portion 420 of high density having each cell storing multiple bits.</p>

Claim Language	Citations to Prior Art
	<p>Fig. 7, col. 11:55-56, 12:6-7: A binary memory or single-level cell (SLC) memory has a population of cells with each cell in one of two possible states.</p> <p>Fig. 8, col. 12:11-12, 31-33: A multi-state memory or multi-level cell (MLC) memory has a population of cells with each cell in one of eight possible states.</p> <p>Abs., col. 6:18-21, 15:62-66, 16:16-22: The memory has a first portion operating with lower density storage and a second portion operating with a higher density.</p> <p>Fig. 15 ref. 500, col. 17:27-29: Configuring memory into a first portion having memory cells operating with a margin of error larger than that of a second portion.</p> <p>Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference:</p> <ul style="list-style-type: none"> • Gorobets US2011/0153912 fig. 1, para. 56: RAM 130 in the controller 100. • Gorobets US2011/0153912 fig. 11, para. 109: Cache RAM 102 in the controller. • Gorobets US2011/0153912 fig. 20, para. 130: RAM 511 in the controller and in the memory 503. • Gorobets US2011/0153912 para. 18: Controller RAM.
<p>[1b] at least one controller to operate memory elements and associated memory space for associated Write access operations to the memory elements and Read access</p>	<p>Gavens discloses at least one controller to operate memory elements and associated memory space for associated Write access operations to the memory elements and Read access operations from the memory elements.</p> <p>For example, Gavens disclose a controller that manages read operations and write operations on memory.</p> <p>Fig. 1 ref. 102, col. 8:17-18, 32-35: A controller 102 manages one or more memory chips 100 in the memory device 90. The controller 102 controls and manages higher level memory operations (e.g., write) on the memory array 200.</p>

Claim Language	Citations to Prior Art
operations from the memory elements;	Col. 2:56-58: A controller performs intelligent and higher level memory operations and interfacing.
[1c] a bank of nonvolatile memory, including: a plurality of MLC memory modules, each including at least one MLC nonvolatile memory element, and at least one SLC memory module including at least one SLC nonvolatile memory element;	<p>Gavens discloses a bank of nonvolatile memory, including: a plurality of MLC memory modules, each including at least one MLC nonvolatile memory element, and 2-state use of a partition of the MLC modules, which obviously may be substituted with at least one SLC memory module including at least one SLC nonvolatile memory element. (See Sutardja, Moshayedi and Lee '794 explained at Exhibit A1 [1g].)</p> <p>For example, Gavens discloses a bank of non-volatile memory chips including SLC memory and MLC memory.</p> <p>Fig. 1, col. 8:13-21: A memory device 90 includes one or more memory chips 100, which includes a memory array 200 of memory cells, which are capable of storing one or more bits of data.</p> <p>Col. 1:18: Non-volatile memory.</p> <p>Col. 9:49-54: A bank of NAND strings in the memory array comprises a series of memory transistors daisy-chained by their sources and drains.</p> <p>Col. 10:31-34: A bank of p sense modules across an array of memory cells operates in parallel to read or program a group of p memory cells in parallel.</p> <p>Fig. 14A, col. 16:15-22: The array of memory cells 200 is partitioned into a first portion 410 of lower density and a second portion 420 of high density having each cell storing multiple bits.</p> <p>Col. 4:32-34, 17:21-23: The first portion has each memory cell storing one bit of data and the second portion has each memory cell storing more than one bit of data.</p> <p>Col. 5:15-17, 50-53, 16:22-24, 20:31-34: The high density storage portion (D3) has each memory cell storing 3 bits of data. The low density storage portion (D1) has each memory cell storing 1 bit of data.</p>

Claim Language	Citations to Prior Art
	<p>Fig. 7, col. 11:55-56, 12:6-7: A binary memory or single-level cell (SLC) memory has a population of cells with each cell in one of two possible states.</p> <p>Fig. 8, col. 12:11-12, 31-33: A multi-state memory or multi-level cell (MLC) memory has a population of cells with each cell in one of eight possible states.</p> <p>Abs., col. 6:18-21, 15:62-66, 16:16-22: The memory has a first portion operating with lower density storage and a second portion operating with a higher density.</p> <p>Fig. 15 ref. 500, col. 17:27-29: Configuring memory into a first portion having memory cells operating with a margin of error larger than that of a second portion.</p> <p>Fig. 6, col. 11:9-10: A memory array organized in erasable blocks.</p> <p>Col. 11:4-8: The array of memory cells is divided into blocks of memory cells. Each block contains the minimum number of memory cells that are erased together.</p>
<p>[1d] at least one random access volatile memory element;</p>	<p>Gavens discloses at least one random access volatile memory element.</p> <p>For example, Gavens discloses volatile RAM.</p> <p>Col. 1:26: RAM.</p> <p>Fig. 16A: First section 411 for caching in first portion 410.</p> <p>Col. 4:35-37: The first portion serves as a cache for incoming data, so a cache copy of the input data is programmed into the cache.</p>

Claim Language	Citations to Prior Art
	<p>Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference:</p> <ul style="list-style-type: none"> • Gorobets US2011/0153912 fig. 1, para. 56: RAM 130 in the controller 100. • Gorobets US2011/0153912 fig. 11, para. 109: Cache RAM 102 in the controller. • Gorobets US2011/0153912 fig. 20, para. 130: RAM 511 in the controller and in the memory 503. • Gorobets US2011/0153912 para. 18: Controller RAM.
<p>[1e] a flash translation layer (FTL), wherein the at least one controller, or the FTL, or a combination of both maintain an address table in one or more of the memory elements;</p>	<p>Gavens discloses a flash translation layer (FTL), wherein the at least one controller, or the FTL, or a combination of both maintain an address table in one or more of the memory elements.</p> <p>For example, Gavens discloses that a controller implements a memory block management system that maintains a directory in memory.</p> <p>Fig. 1 ref. 102, col. 8:17-18, 32-39: The controller 102 manages one or more memory chips 100 in the memory device 90. The controller 102 manages higher-level memory operations, for example, read and write. A memory block management system implemented in the controller 102 maps the logical sectors to the physical structure of the memory array.</p> <p>Col. 2:56-58: A controller performs intelligent and higher level memory operations and interfacing.</p> <p>Fig. 15 ref. 550, col. 17:39-40: Identifying the last written copy as valid data for subsequent read.</p> <p>Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference. Gorobets US2011/0153912 figs. 8-9, paras. 95-99: The memory-side memory manager 300 is implemented in the controller 100 to manage the storage and retrieval of the data of host logical sectors among metablocks of the flash memory. The memory manager 300 comprises a back-end system 320, which includes a number of software modules including a media management layer 330 and a dataflow and sequencing layer 340. The media management layer 330</p>

Claim Language	Citations to Prior Art
	<p>organizes logical data storage within a flash memory meta-block structure. The dataflow and sequencing layer 340 sequences and transfers sectors of data between a front-end system and a flash memory. The dataflow and sequencing layer 340 includes a flash control layer 346. The memory manager 300 is preferably implemented in the controller 100 and translates logical addresses received from the host into physical addresses within the memory array.</p> <p>Col. 8:40-43: The entire disclosure of Paley US2010/0172180 is incorporated by reference. Paley US2010/0172180 figs. 8-9, paras. 155-159, 201-202: The memory-side memory manager 300 is implemented in the controller 100 to manage the storage and retrieval of the data of host logical sectors among metablocks of the flash memory. The memory manager 300 maintains system control data and directory data associated with its operations. The memory manager 300 comprises a back-end system 320, which includes a number of software modules including a media management layer 330 and a dataflow and sequencing layer 340. The media management layer 330 organizes logical data storage within a flash memory meta-block structure. The dataflow and sequencing layer 340 sequences and transfers sectors of data between a front-end system and a flash memory. The dataflow and sequencing layer 340 includes a flash control layer 346. The memory manager 300 is preferably implemented in the controller 100 and translates logical addresses received from the host into physical addresses within the memory array.</p> <p>U.S. Prov. Pat. Appl. No. 61/509,257 (admitted prior art as stated in the “Description of the Prior Art” section) para. 7: The flash translation layer (FTL) performs the function of mapping logical pages to physical flash pages.</p>
[1f] the controller having associated controller memory for storing received data therein, the controller controlling access of the MLC and SLC nonvolatile memory elements and the random access	<p>Gavens renders obvious the controller having associated controller memory for storing received data therein, the controller controlling access of the MLC and SLC nonvolatile memory elements and the random access volatile memory elements for storage of the received data therein as stored data, the controller, in at least a Write access operation transferring the stored received data from the controller memory to a given one of the MLC nonvolatile memory elements in an associated MLC memory module, operable to store the received data in the given one of the MLC nonvolatile memory elements as stored data and retain such received data in the random access volatile memory as retained data associated with stored data.</p>

Claim Language	Citations to Prior Art
<p>volatile memory elements for storage of the received data therein as stored data, the controller, in at least a Write access operation transferring the stored received data from the controller memory to a given one of the MLC nonvolatile memory elements in an associated MLC memory module, operable to store the received data in the given one of the MLC nonvolatile memory elements as stored data and retain such received data in the random access volatile memory as retained data associated with stored data;</p>	<p>For example, Gavens discloses that the controller has buffer memory or cache memory for storing data received from a host; the controller controls access of the first portion of SLC memory, the second portion of MLC memory, and the RAM for storing the data; and in a write operation, the controller transfers the data from the buffer memory or the cache memory to the second portion of MLC memory to store the data in the second portion of MLC memory.</p> <p>Fig. 1 ref. 102, col. 8:17-18, 32-35: A controller 102 manages one or more memory chips 100 in the memory device 90. The controller 102 controls and manages higher level memory operations (e.g., write) on the memory array 200.</p> <p>Col. 12:64-67: As data is received from a host, a page of data is staged in the controller 102. The data page is then written to the memory array 200.</p> <p>Col. 16:41-49: The original copy of a page of incoming data is cached. A first copy of the page of incoming data is written to the second portion.</p> <p>Fig. 14A: Write input data page as first copy of data page in second portion 420 of higher density.</p> <p>Fig. 15 ref. 510, col. 17:30-31: Programming a first copy of a group of input data in the second portion.</p> <p>Col. 4:18-19, 15:66-67: Data is written to the second portion</p> <p>Col. 18:1-4: The first portion is divided into a first section and a second section, and the first section of the first portion serves as a cache.</p> <p>Col. 17:43-46: The first portion serves as a cache for incoming data, so a cache copy of the input data is programmed into the cache, then a first copy of the data is programmed into the second portion.</p> <p>Fig. 17 refs. 602, 610: Programming a cached copy of a group of input data in a first section of the first portion. Programming a first copy of the group of input data in the second portion.</p>

Claim Language	Citations to Prior Art
	<p data-bbox="514 264 1648 297">Fig. 16A, col. 18:9-12: Cached copy of data page in first section 411 of first portion 410.</p> <p data-bbox="514 337 1686 370">Col. 5:53-54: Input data is first staged in D1 (SLC) and subsequently folded into D3 (TLC).</p> <p data-bbox="514 410 1885 516">Col. 16:29-40: The entire disclosure of Chen US6,456,528 is incorporated by reference. Chen US6,456,528 col. 4:39-41: The controller 20 contains buffer memory for the user data being written to or read from the memory array.</p> <p data-bbox="514 557 1719 630">Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference:</p> <ul data-bbox="562 651 1881 1011" style="list-style-type: none"> <li data-bbox="562 651 1881 792">• Gorobets US2011/0153912 paras. 130, 136, 143-44: The data is transferred from the host 501 onto the volatile buffer memory RAM 511 inside the controller, then transferred to the binary section D1 515 of the non-volatile memory 513, and then transferred to the MLC section D3 517 of the non-volatile memory 513. <li data-bbox="562 813 1881 919">• Gorobets US2011/0153912 abs., para. 95: The memory manager 300 is implemented in the controller 100, manages the storage and retrieval of data in the flash memory 200, and maintains system control data and directory data for the flash memory 200 and the controller RAM 130. <li data-bbox="562 940 1881 1011">• Gorobets US2011/0153912 para. 104: Flash memory includes a first portion, which is the main memory, and a second portion, which is a cache of data to be written to the main memory. <p data-bbox="514 1068 1875 1209">Col. 17:57-67: The entire disclosure of Lee US5,930,167 is incorporated by reference. Lee US5,930,167 abs., fig. 7, claims 1, 4: EEPROM flash memory includes an MLC portion and an SLC portion that is used as a write cache. Incoming data is first written to the SLC write cache, then later, the data is read from the SLC write cache and rewritten to MLC.</p> <p data-bbox="514 1284 1835 1349">It would have been obvious to retain the data stored in the RAM so that the data stored in the RAM can serve as cache for faster access of the data in the future.</p>

Claim Language	Citations to Prior Art
	Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference. Gorobets US2011/0153912 paras. 24, 109: Use RAM as a controller cache of the memory controller.
<p>[1g] the controller performing a data integrity test on stored data in the given one of the MLC nonvolatile memory elements in the associated one of the MLC memory modules after the at least a Write access operation performed thereon by reading the stored data to the controller memory and comparing the stored data in the controller memory in the given one of the MLC nonvolatile memory elements to the retained data that was associated with the stored data in the random access volatile memory by the controller during</p>	<p>Gavens renders obvious the controller performing a data integrity test on stored data in the given one of the MLC nonvolatile memory elements in the associated one of the MLC memory modules after the at least a Write access operation performed thereon by reading the stored data to the controller memory and comparing the stored data in the controller memory in the given one of the MLC nonvolatile memory elements to the retained data that was associated with the stored data in the random access volatile memory by the controller during the Write access operation.</p> <p>For example, Gavens discloses that the controller performs a post-write read check on the data stored in the second portion of MLC memory after the write operation by reading the data from the second portion of MLC memory to data latches or the controller and comparing the read data to a cached copy of the data.</p> <p>Abs., col. 6:22-23: After a copy of data is written to the second portion, the data is read from the second portion and checked for errors.</p> <p>Col. 4:19-21, 15:67-16:1: After the data is written to the second portion, the data is read back to check of excessive error bits.</p> <p>Col. 5:1-3: After writing to a high density storage portion, a post-write read checks if the error rate is within a prescribed limit.</p> <p>Col. 6:4-6, 24:45-50: Post-write reading every memory cell to check what has been written.</p> <p>Title: Post-write read to manage errors.</p> <p>Fig. 15 ref. 520, col. 17:32-33: Reading the first copy from the second portion.</p>

Claim Language	Citations to Prior Art
the Write access operation;	<p>Col. 4:55-60: The first copy in the second portion and the cached copy in the first portion are loaded into a set of data latches on an integrated circuit chip and compared to check for error bits.</p> <p>Col. 16:46-49: The first copy of the data page is read back in a post-write read to determine if there are any errors by comparing with the original copy which was cached.</p> <p>Col. 18:30-44: The memory array includes data latches onto which the first copy in the second portion and the cached copy in the first portion can be loaded and compared. The comparison is not made at the controller.</p> <p>Col. 12:67-13:4, 29:57-59: Data is read from the second portion, latched in the data latches 430, and shifted out to the controller 102. The controller 102 compares the ECC in the read data to a second version of ECC computed from the read data.</p> <p>It would have been obvious to compare the data read from the second portion of MLC memory to a cached copy of the data retained in the RAM.</p> <p>Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference:</p> <ul style="list-style-type: none"> • Gorobets US2011/0153912 paras. 24, 109: Use RAM as a controller cache of the memory controller. • Gorobets US2011/0153912 para. 104: Flash memory includes a first portion, which is the main memory, and a second portion, which is a cache of data to be written to the main memory. • Gorobets US2011/0153912 paras. 130, 136, 143-44: The data is transferred from the host 501 onto the volatile buffer memory RAM 511 inside the controller, then transferred to the binary section D1 515 of the non-volatile memory 513, and then transferred to the MLC section D3 517 of the non-volatile memory 513.

Claim Language	Citations to Prior Art
	<p>Col. 8:40-43: The entire disclosure of Paley US2010/0172180 is incorporated by reference. Paley US2010/0172180 para. 155: The memory manager 300 in the controller 100 maintains system control data and directory data associated with its operations among the flash memory 200 and the controller RAM 130.</p>
<p>[1h] wherein the address table maps logical and physical addresses adaptable to the system, wherein the mapping is performed as necessitated by the system to maximize lifetime, and wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and</p>	<p>Gavens discloses the address table maps logical and physical addresses adaptable to the system, wherein the mapping is performed as necessitated by the system to maximize lifetime, and wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories.</p> <p>For example, Gavens discloses that the directory maps logical addresses and physical addresses adaptable to the flash memory device, the mapping is performed as necessitated by the flash memory device to maximize the lifetime of the flash memory device, and the mapping maps a block of data in the non-volatile memory array.</p> <p>Col. 8:36-39: A memory block management system implemented in the controller maps the logical sectors to the physical structure of the memory array.</p> <p>Col. 15:2-6: An important consideration for flash memory is that it has an endurance problem as it ages with use. The error rate increases with the number of program-erase cycling.</p> <p>Col. 16:34-37: It is desirable to have longer operational life of memory cells.</p> <p>Fig. 15 ref. 550, col. 17:39-40: Identifying the last written copy as valid data for subsequent read.</p> <p>Col. 11:19-22: The group of cells erasable together consists of one or more addressable erase units. The erase unit or a block stores one or more pages of data. A page is a minimum unit of programming and reading.</p> <p>Col. 21:21-23: Each block contains data from a well-defined group of logical addresses.</p>

Claim Language	Citations to Prior Art
	<p>Col. 5:41-45: A hot count is tracks the number of times each erase block has been cycled through erase and program operations.</p> <p>Col. 8:40-43: The entire disclosure of Paley US2010/0172180 is incorporated by reference. Paley US2010/0172180 figs. 8-9, paras. 155-159, 201-202: The memory-side memory manager 300 is implemented in the controller 100 to manage the storage and retrieval of the data of host logical sectors among metablocks of the flash memory. The memory manager 300 maintains system control data and directory data associated with its operations. The memory manager 300 comprises a back-end system 320, which includes a number of software modules including a media management layer 330 and a dataflow and sequencing layer 340. The media management layer 330 organizes logical data storage within a flash memory meta-block structure. The dataflow and sequencing layer 340 sequences and transfers sectors of data between a front-end system and a flash memory. The dataflow and sequencing layer 340 includes a flash control layer 346. The memory manager 300 is preferably implemented in the controller 100 and translates logical addresses received from the host into physical addresses within the memory array.</p> <p>Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference. Gorobets US2011/0153912 figs. 8-9, paras. 95-99: The memory-side memory manager 300 is implemented in the controller 100 to manage the storage and retrieval of the data of host logical sectors among metablocks of the flash memory. The memory manager 300 maintains system control data and directory data associated with its operations. The memory manager 300 comprises a back-end system 320, which includes a number of software modules including a media management layer 330 and a dataflow and sequencing layer 340. The media management layer 330 organizes logical data storage within a flash memory meta-block structure. The dataflow and sequencing layer 340 sequences and transfers sectors of data between a front-end system and a flash memory. The dataflow and sequencing layer 340 includes a flash control layer 346. The memory manager 300 is preferably implemented in the controller 100 and translates logical addresses received from the host into physical addresses within the memory array.</p> <p>Col. 17:57-67: The entire disclosure of Lee US5,930,167 is incorporated by reference. Lee US5,930,167 col. 2:54-3:7: The usual desire to evenly wear the memory. Avoid using one portion of memory much more than other portions. Spread out heavy use function across the memory. Maintain separate counts of the</p>

Claim Language	Citations to Prior Art
	<p>number of times that each sector has been erased and programmed to even the wear among the sectors.</p> <p>Col. 16:29-40: The entire disclosure of Chen US6,456,528 is incorporated by reference. Chen US6,456,528 col. 10:41-60: Storing the most frequently written data in SLC and less frequently written data in MLC avoids the need to replace blocks that reach their endurance limit.</p> <p>U.S. Prov. Pat. Appl. No. 61/509,257 (admitted prior art as stated in the “Description of the Prior Art” section) para. 7: The flash translation layer (FTL) performs the function of mapping logical pages to physical flash pages.</p>
<p>[1i] wherein a failure of the data integrity test performed on the stored data by the controller results in a remapping of address space to a different physical range of addresses and transfer of data corresponding to the retained data to those remapped physical addresses from those physical addresses determined to have failed the data integrity test.</p>	<p>Gavens discloses a failure of the data integrity test performed on the stored data by the controller results in a remapping of address space to a different physical range of addresses and transfer of data corresponding to the retained data to those remapped physical addresses from those physical addresses determined to have failed the data integrity test.</p> <p>For example, Gavens discloses that a failure of the post-write read check performed on the data stored in the second portion of MLC memory by the controller results in a remapping of the logical address of the data from the physical address in the second portion of MLC memory to a different physical address in the first portion of SLC memory and a rewriting of the cached data to the first portion of SLC memory.</p> <p>Fig. 14B: If post-write read failed, then rewrite input data page as a second copy of data page in first portion 410 of lower density.</p> <p>Col. 5:3-7: If post-write read checks results in an error rate above the prescribed limit, then the data is rewritten to another, less error-prone portion that has each memory cell storing less data bits per cell.</p> <p>Col. 17:9-12: After the post-write read detects that the number of error bits in the data page has exceeded the predetermined amount, a second copy of the data page is made from the original data which was cached, and the second copy is rewritten to the first portion.</p>

Claim Language	Citations to Prior Art
	<p>Fig. 15 refs. 530, 540, col. 17:34-38: Does the error exceed a predetermined number of error bits? If yes, then programming a second copy of the group of input data in the first portion.</p> <p>Abs., col. 6:23-25: If the copy in the second portion has excessive error bits, then a copy is written to the first portion.</p> <p>Col. 4:21-22, 16:1-3: If error bits exceeds a predetermined amount, then the data is rewritten at the less error-prone first portion.</p> <p>Col. 18:53-67: After a post-write read detects excessive number of error bits, a second copy of the data page is taken from the cached copy and rewritten to the second section of the first portion. The directory in the block management system embodied in the firmware of the controller is updated to direct subsequent access to the second copy.</p> <p>Col. 4:62-67: A directory in a block management system embodied in the firmware of the controller is updated to render the first copy in the second portion obsolete and to direct subsequent access to the cached copy in the first portion. The cached copy in the first portion replaces the first copy in the second portion as the valid data.</p> <p>Col. 17:15-20: A directory in a block management system embodied in the firmware of the controller is updated to render the first copy in the second portion obsolete and to direct subsequent access to the second copy in the first portion. The second copy in the first portion replaces the first copy in the second portion as the valid data.</p>
<p>[2] The system of claim 1, wherein the FTL is a software module, or a firmware module containing software updates.</p>	<p>Gavens discloses the FTL is a software module, or a firmware module containing software updates.</p> <p>For example, Gavens discloses that the memory block management system implemented in the controller is software.</p>

Claim Language	Citations to Prior Art
	<p>Fig. 1 ref. 102, col. 8:36-39: A memory block management system implemented in the controller 102 maps the logical sectors to the physical structure of the memory array.</p> <p>Fig. 1 ref. 60, col. 8:44-45: The firmware 60 provides codes to implement the functions of the controller 102.</p> <p>Col. 18:64-67: A directory in the block management system embodied in the firmware of the controller is updated to direct subsequent access to the second copy.</p> <p>Col. 4:62-67: A directory in a block management system embodied in the firmware of the controller is updated to render the first copy in the second portion obsolete and to direct subsequent access to the cached copy in the first portion. The cached copy in the first portion replaces the first copy in the second portion as the valid data.</p> <p>Col. 17:15-20: A directory in a block management system embodied in the firmware of the controller is updated to render the first copy in the second portion obsolete and to direct subsequent access to the second copy in the first portion. The second copy in the first portion replaces the first copy in the second portion as the valid copy.</p>
<p>[3] The system of claim 1, wherein at least one of the random access volatile memory or the MLC and SLC nonvolatile memory elements are embedded in the at least one controller.</p>	<p>Gavens discloses at least one of the random access volatile memory or the MLC and SLC nonvolatile memory elements are embedded in the at least one controller.</p> <p>For example, Gavens discloses that the RAM is embedded in the controller.</p> <p>Col. 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as US2011/0153912, is incorporated by reference:</p> <ul style="list-style-type: none"> • Gorobets US2011/0153912 fig. 1, para. 56: RAM 130 in the controller 100. • Gorobets US2011/0153912 fig. 11, para. 109: Cache RAM 102 in the controller.

Claim Language	Citations to Prior Art
	<ul style="list-style-type: none"> • Gorobets US2011/0153912 fig. 20, para. 130: RAM 511 in the controller. • Gorobets US2011/0153912 para. 18: Controller RAM. <p>Col. 8:40-43: The entire disclosure of Paley US2010/0172180 is incorporated by reference:</p> <ul style="list-style-type: none"> • Paley US2010/0172180 fig. 1, para. 119: RAM 130 in the controller 100. • Paley US2010/0172180 fig. 15, para. 197: Cache RAM 102 in the controller.
<p>[4] The system of claim 1, wherein the MLC and SLC nonvolatile memory elements comprise flash memory.</p>	<p>See Exhibit A1 ('298 patent) claims 4 and 5.</p>
<p>[5] The system of claim 1, wherein the random access volatile memory element is dynamic random access memory.</p>	<p>Gavens, in combination with other reference(s), renders obvious the random access volatile memory element is dynamic random access memory.</p> <p>For example, US2009/0327591 ("Moshayedi") discloses dynamic random access memory.</p> <p>Moshayedi fig. 1 ref. 106, paras. 3-4, 28, 53-55, 67: DRAM.</p> <p>For example, Duann, Nelson, "SLC & MLC Hybrid," Silicon Motion Flash Memory Summit, 2008 ("Duann") discloses dynamic random access memory.</p> <p>Duann page 6: Utilize DRAM as cache.</p>

Claim Language	Citations to Prior Art
<p>[6] The system of claim 1, wherein the random access volatile memory element is static random access memory.</p>	<p>Gavens, alone or in combination with other reference(s), renders obvious the random access volatile memory element is static random access memory.</p> <p>Col. 8:40-43: The entire disclosure of Paley US2010/0172180 is incorporated by reference. Paley US2010/0172180 paras. 423-425, 502-503, 506: SRAM.</p> <p>Duann page 6: SRAM as cache.</p>