

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PHISON ELECTRONICS CORPORATION,
Petitioner

v.

VERVAIN, LLC,
Patent Owner

U.S. Patent No. 11,830,546

Post-Grant Review Case No. PGR2024-00047

DECLARATION OF CARL SECHEN, PH.D.

TABLE OF CONTENTS

I.	Overview:	1
II.	Background and Qualifications:	4
III.	My Understanding of the Applicable Law	8
IV.	Documents Considered in Formulating My Opinions:	11
V.	Relevant State of the Art as of July 19, 2011	14
	A. NAND Flash Memory and Operations	16
	B. Prior Use of SLCs and MLCs in the Same Memory System	26
	C. “Data Integrity Checking” in NAND Flash Memory Systems	40
	D. Strategies for NAND Flash Memory Remapping To Manage System Reliability and Efficiency	48
VI.	Person of Ordinary Skill in the Art (POSITA)	51
VII.	The Common Specification of the Lifetime Patents	52
	A. Lifetime Patents Specification BACKGROUND	52
	B. Proffered Embodiments in the DETAILED DESCRIPTION	55
VIII.	The Evolving Abstraction of the Lifetime Patents Claims	70
	A. Pre-2018 Physical MLC and SLC Modules and Coupled Controller Defined by Functions	71
	B. 2018 Abstraction of the Preferential Mapping to an SLC Module	74
IX.	The ‘546 Claims and their Textual and Conceptual Defects:	83
X.	Patentability of the ‘546 Claims Relative to the Art	106
	A. Claim 1 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA	107
	1. [1Pre] A system for storing data comprising:	107

2. [1a] [1a1] memory space containing volatile memory space and [1a2] nonvolatile memory space, wherein the nonvolatile memory space includes both multi-level cell (MLC) space and single level cell (SLC) space; 108
3. [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 operations from the memory elements; ...110
4. [1c] a bank of nonvolatile memory, including: [1c1] a plurality of MLC memory modules, each including at least one MLC nonvolatile memory element, and [1c2]at least one SLC memory module including at least one SLC nonvolatile memory element; 110
5. [1d] at least one random access volatile memory element;.....114
6. [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; 115
7. [1f] the controller [1f1] having associated controller memory for storing received data therein, the controller [1f2] 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, [1f3] 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, [1f4] 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;.....118
8. [1g] the controller [1g1] 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 [1g2] reading the stored data to the controller memory and [1g3] 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; 122
9. [1h] wherein [1h1] the address table maps logical and physical addresses adaptable to the system, wherein [1h2] the mapping is

performed as necessitated by the system to maximize lifetime, and wherein [1h3] the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and	125
10. [1i] wherein a failure of the data integrity test performed on the stored data by the controller results in [1i1] a remapping of address space to a different physical range of addresses and [1i2] 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.....	128
B. Claim 2 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.....	131
1. [2Pre] The system of claim 1, ...	131
2. [2] wherein the FTL is a software module, or a firmware module containing software updates.....	131
C. Claim 3 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.....	132
1. [3Pre] The system of claim 1, ...	132
2. [3] 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.	132
D. Claim 4 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.....	133
1. [4Pre] The system of claim 1, ...	133
2. [4] wherein the MLC and SLC nonvolatile memory elements comprise flash memory.....	133
E. Claim 5 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.....	134
1. [5Pre] The system of claim 1, ...	134
2. [5] The system of claim 1, wherein the random access volatile memory element is dynamic random access memory.	135
F. Claim 6 Is Obvious Over Gavens, Including Incorporated references, in View of Knowledge of the POSITA.....	135

1. [6Pre] The system of claim 1, ...	135
2. [6] wherein the random access volatile memory element is static random access memory.	135
G. Claim 7 Is Obvious Over Gavens, Including Incorporated references, in View of Knowledge of the POSITA and/or Colligan.....	136
1. [7Pre] The system of claim 1, ...	136
2. [7] wherein the at least one SLC nonvolatile memory element includes a hard disk drive (HDD).	136
XI. Conclusion	137

I, Carl Sechen, Ph.D., declare as follows.

I. Overview:

1. I am over the age of eighteen and otherwise competent to make this Declaration. I have been retained as an expert witness on behalf of petitioner Phison Electronics Corporation (“Petitioner”) for the above-captioned post-grant review (“PGR”). I am being compensated for my time in connection with this review at my standard consulting rate, which is \$450.00 per hour. I have no personal or financial interest in the outcome of this proceeding. I make this Declaration on personal knowledge and on my expertise; where I state understandings with respect to patent law or procedure, the basis generally is discussion with counsel for Petitioner.

2. I have been asked to provide my expert opinions on a petition for post-grant review (“PGR”) of U.S. Patent No. 11,830,546, entitled “LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM,” to Rao (the “‘546 patent”) (Ex. 1001), which issued on Nov. 28, 2023, from U.S. Appl. No. 17/203,385 filed March 16, 2021, as a continuation of U.S. Appl. No. 16/006,299 filed June 12, 2018,¹ which was a continuation of U.S. Appl. No. 14/950,553 filed Nov. 24, 2015,² which was a

¹ Now U.S. Patent No. 10,950,300 (the “‘300 patent” or “Rao ‘300,” Ex. 1011).

² Now U.S. Patent No. 9,997,240 (the “‘240 patent” or “Rao ‘240,” Ex. 1009).

continuation of U.S. Appl. No. 14/525,411 filed Oct. 28, 2014,³ which was a continuation of U.S. Appl. No. 13/455,267 filed Apr. 25, 2012,^{4,5} collectively referred as the “Lifetime Patents.” Each of these incorporated by reference two patent applications for “NONVOLATILE MEMORY SYSTEMS WITH EMBEDDED FAST READ AND WRITE MEMORIES” on a common disclosure, which issued as U.S. Patents Nos. 7,855,916⁶ and 8,194,452 (“Rao ‘452,” Ex. 1004),⁷ collectively referred to as “Embedded RAM Patents.” I further understand that, according to the USPTO records, the ‘546 patent is currently assigned to Vervain, LLC (“Patent Owner”).

3. I understand that the ‘546 patent (as its parent applications, which have identical descriptions of the invention except for their abstracts) claims the priority benefit of U.S. Provisional App. No. 61/509,257, filed on July 19, 2011 (“Provisional,” Ex. 1003). I understand that, based on that filing date, the prior art references and state of the art which I should consider in evaluating the ‘546 patent’s patentable inventiveness should be July 18, 2011, or earlier.

³ Now U.S. Patent No. 9,196,385 (the “‘385 patent” or “Rao ‘385,” Ex. 1007).

⁴ Now U.S. Patent No. 8,891,298 (the “‘298 patent” or “Rao ‘298,” Ex. 1005).

⁵ Three more continuation applications based on the same disclosure as the ‘298 patent have issued as U.S. Patent Nos. 11,854,612, 11,967,369, and 11,967,370.

⁶ U.S. Pat. App. No. 12/256,232, filed Oct. 22, 2008.

⁷ U.S. Pat. App. No. 12/915,177, filed Oct. 29, 2010.

4. In preparing this Declaration, I have reviewed the ‘546 patent and have considered each of the documents cited herein upon which I specifically rely. In formulating my opinions, I have relied upon my experience, education, and knowledge in the relevant art, including other background documents. I have also considered the viewpoint of a person of ordinary skill in the art (“POSITA”) of computer memories (as described below) at the time of the alleged invention of the ‘546 patent, that is, close to, but earlier than July 19, 2011.

5. This Declaration is limited to my opinions on invalidity with respect to claims 1-7 of the ‘546 patent (the “challenged claims”). Those opinions, set forth in detail below, including factual bases and assumptions, are summarized:

A. The challenged claims 1-7 are unenabled under 35 U.S.C. § 112(a) because of (a) lack of detail, accuracy and/or consistency in the specification; and (b) lack of operability of the claims for suggested, but undisclosed, embodiments.

B. The challenged claims 1-7 are indefinite under 35 U.S.C. § 112(b) as (a) failing, in the vagueness, ambiguity and/or contradiction of their terms, to notify the defined POSITA of the scope of the claims; and (b) claiming a generic “controller” by functionality without adequate structure (physical or algorithmic) in conformance with 35 U.S.C. § 112(f).

C. Claims 1-7 are obvious over U.S. Patent No. 8,634,240 (“Gavens,” Ex. 1045), including incorporated references, in view of the knowledge of the POSITA of the extensive background.

6. While the aforementioned references form the primary bases for my opinion, I note that there are many other relevant references which further support a finding of invalidity as discussed herein. I reserve the possibility of amending or supplementing this Declaration should additional information or arguments develop or be presented which affect my opinions.

II. Background and Qualifications:

7. My qualifications and credentials are set forth herein as further evidenced by my curriculum vitae (addendum hereto).

8. I earned a B.E.E. in Electrical Engineering from the University of Minnesota in 1975, followed by a M.S. in Electrical Engineering from the Massachusetts Institute of Technology in 1977. I earned a Ph.D. in Electrical Engineering from the University of California, Berkeley in 1986.

9. I have been a Professor of Electrical Engineering for 38 years. Since August 15, 2005, I have been a Professor of Electrical and Computer Engineering at the University of Texas at Dallas. From July 1992 to August 14, 2005, I served as a Professor of Electrical Engineering at the University of Washington. From July

1986 through June 1992, I served as an Assistant Professor and then Associate Professor of Electrical Engineering at Yale University.

10. Over these years, my research has focused on the design and computer-aided design of digital integrated circuits, including the design of dynamic random-access memory (“DRAM”) and static random-access memory (“SRAM”) modules. I have also designed several chips that included various types of embedded DRAM (eDRAM). In this research work, I have designed multiple sense amplifier circuits for the various DRAMs, embedded DRAMs, and SRAMs. I have authored or coauthored over 200 papers and one book, the majority of which concern digital integrated circuit design and memory design.

11. As a professor, I have developed and taught numerous courses, including several courses that teach digital integrated circuit design and memory design, including extensive coverage of flash memory design. I have taught these courses continuously for the past 29 years. I have taught digital integrated circuit design and memory design, including flash memory design, to undergraduate and graduate students at the University of Washington and at the University of Texas at Dallas. DRAM and embedded DRAM design is covered in these courses, as is the design of sense amplifier circuits for various types of memory. NAND-Flash memory design is taught in detail, including both SLC (single-level cell) and MLC (multi-level cell) variants.

12. I was elected a Fellow of the Institute of Electrical and Electronics Engineers (“IEEE”) in 2002 for contributions to placement and routing of ASICs. IEEE is the leading professional association for electrical engineers. The Board of Directors of the IEEE awards the rank of “Fellow” to individuals with an extraordinary record of accomplishments in any of the IEEE fields of interest. The total number of IEEE members who can be named Fellows in any one year cannot exceed one-tenth of one percent of the total voting IEEE membership.

13. I received several research and teaching awards during my career. I received the Semiconductor Research Corporation’s Inventor’s Recognition Award in 1988 and in 2001. I also received the Technical Excellence Award from the Semiconductor Research Corporation in 1994. While serving as Professor at the University of Washington, I received the Outstanding Research Advisor award from the Department of Electrical Engineering in 2002. In 2008, I received the Distinguished Teacher of the Year Award from the Department of Electrical and Computer Engineering (ECE) at the University of Texas at Dallas. I also received the Distinguished Teaching Award for the Erik Johnson School of Engineering and Computer Science in 2014. In addition, I received the *Graduate Teaching Excellence Award* from the ECE Department at UT Dallas, May 2024.

14. Over the years, I have also received funding to conduct research in computer circuits and memory designs, including area-efficient and reliable

embedded DRAM and SRAM design. Together with my graduate students, I have designed and fabricated various types of computational VLSI chips.

15. I am a co-inventor on four issued patents directed to transistor and computational logic technologies. Three of the patents concern transistor-level programmable logic.

16. In summary, I am an expert in the field of computer architecture, including memory design and management, generally and as applied within the context of the '546 patent, including NAND flash and other non-volatile memories ("NVM"). As set forth above, my experience includes practical product development, including performance considerations and design choices involving the memory technology state of the art at and before the earliest, July 19, 2011, priority date applicable to the challenged claims.

17. As an expert, I am qualified to evaluate, interpret and rely on reports, patents, scholarly papers and professional opinions of others, within my areas of expertise, as to authenticity and accuracy. Moreover, I am qualified to provide an opinion as to what a person of ordinary skill in the art (POSITA) would have understood, known and concluded before July 19, 2011, including the context in which a POSITA would understand and construe teachings and terminology in written publications.

III. My Understanding of the Applicable Law

18. I am not an attorney. However, as outlined in my curriculum vitae addended hereto, I have formed and presented opinions according to my technology expertise and hypotheticals presented to me in other cases involving validity of United States patents, including in *Vervain, LLC v. Western Digital Corp.*, No. 6:21-cv-00488 (W.D. Tex. filed May 10, 2021, settled Aug. 8, 2023), which involved the four other Rao patents identified in paragraph 2 on the same specification as the ‘546 patent. In those cases, as here, I was informed or reminded of the legal standards which are relevant to my analysis and opinions. I summarize here the understandings relevant for this declaration, noting that greater detail may be provided upon any cross-examination, and others may be provided for any amendment or supplementation reserved in paragraph 6.

19. **Written Description and Claiming Requirements.** I understand that, as part of the bargain with the public of providing information to advance “science and the useful arts” in return for patent right claimed and granted with notice to the public, 35 U.S.C. § 112 requires, in subsection (a), an adequate written description that both enables a POSITA to practice, without undue experimentation, the invention as claimed and reasonably conveys to the POSITA that “the inventor had possession of the claimed subject matter as of the filing date” and, in subsection (b), that a patent’s claims be “concise” and, viewed in light of the specification and

prosecution history, inform the POSITA about “the scope of the invention with reasonable certainty.” I also understand that, applied with other statutory requirements, there are instances in which (1) claim elements that are defined by a function may be found to be indefinite if there is inadequate structure disclosed for performing the function, (2) mixed product and process claims may be invalid for inadequate notice to patent-respecting manufacturers or users, and (3) a claim may be invalid for failing to recite a “useful” invention if it is manifestly inoperative.

20. What Constitutes Prior Art that May Invalidate a Patent. I understand that former 35 U.S.C. § 102 listed conditions, including what is considered “prior art,” that may preclude (or invalidate) a patent of the ‘546 patent vintage, most pertinent here being:

(1) under subsection (a), if the invention was known or used by others in this country, or was patented or described in a printed publication in this or a foreign country, before the date of the invention of the asserted patent; or

(2) under subsection (b), if the invention was patented or described in a printed publication in this or a foreign country, or in public use or on sale in this country, more than one year before the effective filing date of the asserted patent; or

(3) under subsection (e), if the invention was described in (a) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (b) a patent granted

on an application for patent by another filed in the United States before the invention by the applicant for patent; or

I understand that one or more of these conditions apply to the art relied upon here for my invalidity analysis.

21. How the Prior Art Is Applied in the Invalidity Analysis:

Anticipation: I understand that a patent claim may be found to be invalid as “anticipated” under a subsection of 35 U.S.C. § 102 if a single prior art document (reference) under that subsection describes every element of the claim explicitly or inherently (necessarily present) as understood by a POSITA to allow making and using the claimed invention.

22. How the Prior Art Is Applied in the Invalidity Analysis:

Obviousness: I understand that a patent claim may be invalid as “obvious” under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the claimed invention, as a whole, would have been obvious to a POSITA before the effective filing date of the claimed invention. The analysis calls for consideration of (a) level of ordinary skill in the pertinent art; (b) the scope and content of the prior art; (c) the differences between the prior art as a whole and the claim at issue; and (d) as appropriate, objective indicia of non-obviousness. The objective indicia, appropriate if there is a nexus with the claimed invention, include: (1) commercial success attributable to the invention; (2) long-felt but unsolved need

for the invention; (3) failure of others; (4) skepticism by experts; (5) praise by others; (6) teaching away by others; (7) recognition of a problem which the claimed invention addresses; (8) copying by others; and (9) unexpected results.

23. I understand that an obviousness may be found on a single reference considered in view of the knowledge of the POSITA. Where multiple references are proffered, I understand that while there is no requirement of an explicit teaching, suggestion or motivation to combine, generally there needs to be put forth some reason why a POSITA (if only by common sense or ordinary creativity) would have made the particular combination and that there is a reasonable expectation of success of the combination.

IV. Documents Considered in Formulating My Opinions:

24. In formulating my opinions, I have considered all documents cited in this Declaration, including, in particular, the documents in the chart below:

Exhibit	Description
1001	U.S. Patent No. 11,830,546, titled “LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM,” to Rao
1003	U.S. Provisional Application No. 61/509,257, titled “IMPROVED LIFETIME MIXED LEVEL NAND FLASH SYSTEM” by Rao
1004	U.S. Patent No. 8,194,452, titled “NONVOLATILE MEMORY SYSTEMS WITH EMBEDDED FAST READ AND WRITE MEMORIES,” to Rao

1005	U.S. Patent No. 8,891,298, titled “LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM,” to Rao
1007	U.S. Patent No. 9,196,385, titled “LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM,” to Rao
1009	U.S. Patent No. 9,997,240, titled “LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM,” to Rao
1011	U.S. Patent No. 10,950,300, titled “LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM,” to Rao
1012	Excerpts from prosecution history of ’300 patent
1014	Excerpts from prosecution history of ’546 patent
1025	Excerpts from Rino Micheloni, et al., INSIDE NAND FLASH MEMORIES (Springer 2010)
1026	Excerpts from NONVOLATILE MEMORY TECHNOLOGIES WITH EMPHASIS ON FLASH: A COMPREHENSIVE GUIDE TO UNDERSTANDING AND USING NVM DEVICES (Joe E. Brewer & Manzur Gill eds. Wiley-IEEE Press 2008)
1027	Excerpts from MICROSOFT COMPUTER DICTIONARY (5 th ed. 2002)
1030	Yu Cai et al., <i>Program Interference in MLC NAND Flash Memory: Characterization, Modeling and Mitigation</i> (IEEE 2013)
1031	Kang-Deog Suh et al., <i>A 3.3 V 32 Mb NAND Flash Memory with Incremental Step Pulse Programming Scheme</i> , 30 IEEE J. Solid State Circuits 1149 (Nov. 1995)
1032	Eitan Yaakobi et al., <i>Error Characterization and Coding Schemes for Flash Memories</i> (2010 IEEE Globecom Workshops, Miami, FL at 1856-60)
1033	Evans, Chris, “Enterprise MLC: How flash vendors are boosting MLC write endurance” (June 3, 2011)

1034	Nelson Duann, SLC & MLC Hybrid (Silicon Motion Flash Memory Summit, Santa Clara, CA, Aug. 12, 2008)
1035	Ashish Jagmohan et. al., <i>Adaptive Endurance Coding for NAND Flash</i> (2010 IEEE Globecom Workshops, Miami, FL at 1841-45)
1036	Sungjin Lee et al., <i>FlexFS: A Flexible Flash File System for MLC NAND Flash Memory</i> (2009 USENIX Annual Technical Conference, San Diego, CA, June 14-19, 2009)
1037	Minyoung Sung & Kanghee Kim, <i>Asymmetric Flash Volume Management</i> , 58 <i>IEEE Transactions on Consumer Electronics</i> 455 (May 2012)
1040	U.S. Patent No. 6,456,528, titled for “SELECTIVE OPERATION OF A MULTI-STATE NON-VOLATILE MEMORY SYSTEM IN A BINARY MODE” to Chen
1041	U.S. Patent No. 8,078,794, titled “HYBRID SSD USING A COMBINATION OF SLC AND MLC FLASH MEMORY ARRAYS,” to Lee et al.
1042	U.S. Patent Appl. Pub. No. 2008/0140918, titled “HYBRID NON-VOLATILE SOLD STATE MEMORY SYSTEM” by Sutardja
1043	U.S. Patent Appl. Pub. No. 2009/0327591, titled “SLC-MLC COMINATION FLASH STORAGE,” by Moshayed
1044	U.S. Patent Appl. Pub. No. 2010/0172179, titled “SPARE BLOCK MANAGEMENT OF NON-VOLATILE MEMORIES,” by Gorobets et al.
1045	U.S. Patent No. 8,634,240, titled “NON-VOLATILE MEMORY AND METHOD WITH ACCELERATED POST-WRITE READ TO MANAGE ERRORS,” to Gavens et al.
1046	U.S. Patent No. 8,806,301, titled “DATA WRITING METHOD FOR A FLASH MEMORY, AND CONTROLLER AND STORAGE SYSTEM USING THE SAME,” to Yu et al.

1047	U.S. Patent Appl. Pub. No. 2010/0115192, titled “WEAR LEVELING METHOD FOR NON-VOLATILE MEMORY DEVICE HAVNG SINGLE AND MULTI LEVEL MEMORY CELL BLOCKS,” by Lee
1048	U.S. Patent Appl. Pub. No. 2011/0271043, titled “SYSTEM AND METHOD FOR ALLOCATING AND USING SPARE BLOCKS IN A FLASH MEMORY,” to Segal et al.
1049	U.S. Patent Appl. Pub. No. 2011/0153912, titled “MAINTAINING UPDATES OF MULTI-LEVEL NON-VOLATILE MEMORY IN BINARY NON-VOLATILE MEMORY,” by Gorobets et al.
1050	U.S. Patent No. 5,930,167, titled “MULTI-STATE NON-VOLATILE FLASH MEMORY CAPABLE OF BEING ITS OWN TWO STATE WRITE CACHE,” to Lee et al.
1051	U.S. Patent Appl. Pub. No. 2010/0172180, titled “NON-VOLATILE MEMORY AND METHOD WITH WRITE CACHE PARTITIONING,” by Paley et al.
1052	U.S. Patent No. 8,225,035, titled “SYSTEMS AND METHODS FOR OPERATING A DISK DRIVE,” to Colligan

V. Relevant State of the Art as of July 19, 2011

25. I describe here the state of the art of NAND flash memory technology prior to July 19, 2011, that much relied upon, but only touched upon, in some cases incorrectly, in the common specification for the ’546 patent and each of the seven other patents listed in paragraph 2 (collectively “Lifetime Patents”). As I analyze at Section VII below, all that was disclosed in that specification is a NAND flash memory system (such as a solid state drive) of three components: (1) a bank of multiple physical modules of single-level (1-bit) cell “SLC NAND Flash,” (2) a bank

of multiple physical modules of multi-level (multi-bit) cell “MLC NAND Flash,” and (3) an abstract “controller” that is not specified structurally as hardware such as a common NAND flash controller module or even as software. Instead, the controller is defined in the claims by functions variably corresponding to the only embodiments disclosed in the specification, namely (a) *controlling* NAND flash memory read/write/erase functions using NAND-characteristic logical (host-visible)-to-physical (“L2P”) *mapping* (including by a flash translation layer or “FTL”), (b) as in Figs. 3A and 3B, *performing* a “data integrity verification check” on data to be written to a mapped physical location by *comparing* what is read from that location after the writing and, upon a failure to match, *remapping* to an available location in the SLC memory, (c) *allocating* “hot” blocks (that receive frequent writes) “into the SLC NAND flash module” and “cold” blocks (that only receive infrequent writes) “into the MLC NAND flash module” and (d) on a periodic basis, *transferring* the contents of those blocks written to most frequently “into the SLC NAND flash module” (with ambiguity as to whether the referenced blocks are the NAND flash-defining physical, erasable blocks, or “logical” blocks mapped to them). These functions were proposed in the specification to take advantage of the well-known superior endurance of SLC memory over MLC memory in schemes to avoid premature end to reliable use of a memory system composed of multiple MLC and SLC NAND flash modules. Although the specification proposed the

substitution for the SLC module of technologies such as PCM, RRAM and MAGRAM, it only explained that those technologies operated essentially differently from NAND flash in that they did not require erasure before writing – and thus failed to explain how they could be substituted operably. I review here relevant terms and technical details that a POSITA would have understood to underly the NAND flash embodiments sketched in the specification.

A. NAND Flash Memory and Operations

26. NAND flash memories operate with connected cells (field-effect transistors, as the example shows below) in a characteristic block. A NAND flash block is erased (all cells are set to the logical “1” state) as a unit (typically 64 or 128 pages of 4K or 2K data as well as 128 or 64 spare bytes [8 bits each]) used for other functions, such as error correction code, “ECC”). When a cell is erased, meaning there is essentially no trapped charge in the floating gate, that cell will be read by detection circuitry in the module (chip) as having current readily conducting from drain to source for a nominally applied control gate voltage. Also, the smallest programmable unit of NAND flash is a “page,” or row of cells, where the programming of the cells in a page is such that a cell may be left in the logical 1 state from erasure or a cell may be programmed to logical 0 (trapped charge is added to the floating gate), where the detection circuitry in the module will find largely an

absence of current conducting from drain to source for the nominally applied control gate voltage.

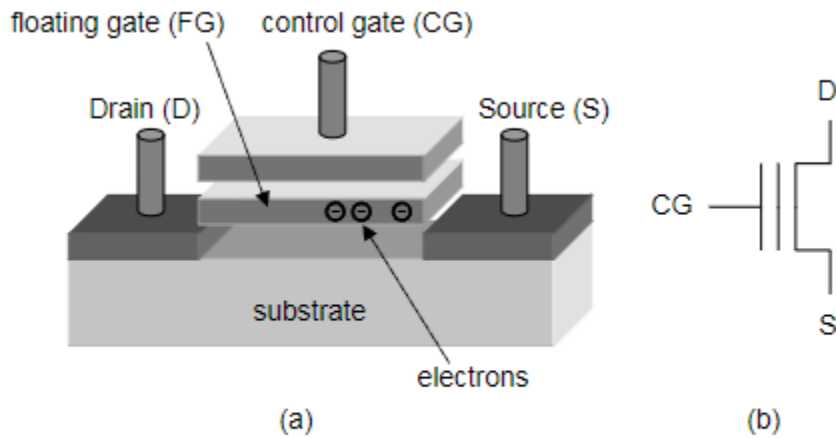


Fig. 2.1. (a) Floating gate memory cell and (b) its schematic symbol

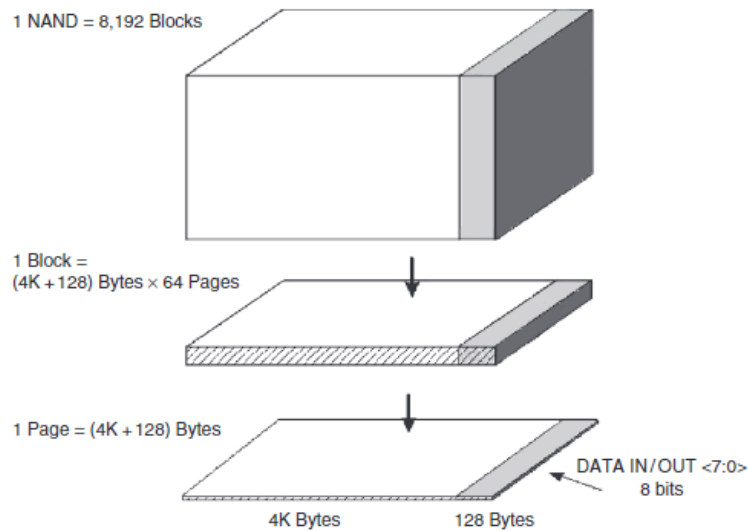


Fig. 2.7. NAND memory logic organization

Rino Micheloni, et al., INSIDE NAND FLASH MEMORIES (2010) (“Micheloni,” Ex. 1025) at 20, 27.

27. Typically, a NAND flash memory card or SSD (solid state drive) includes one or more banks of flash memory modules, such as chips, in an array and at least a device-level controller coupled to those modules:

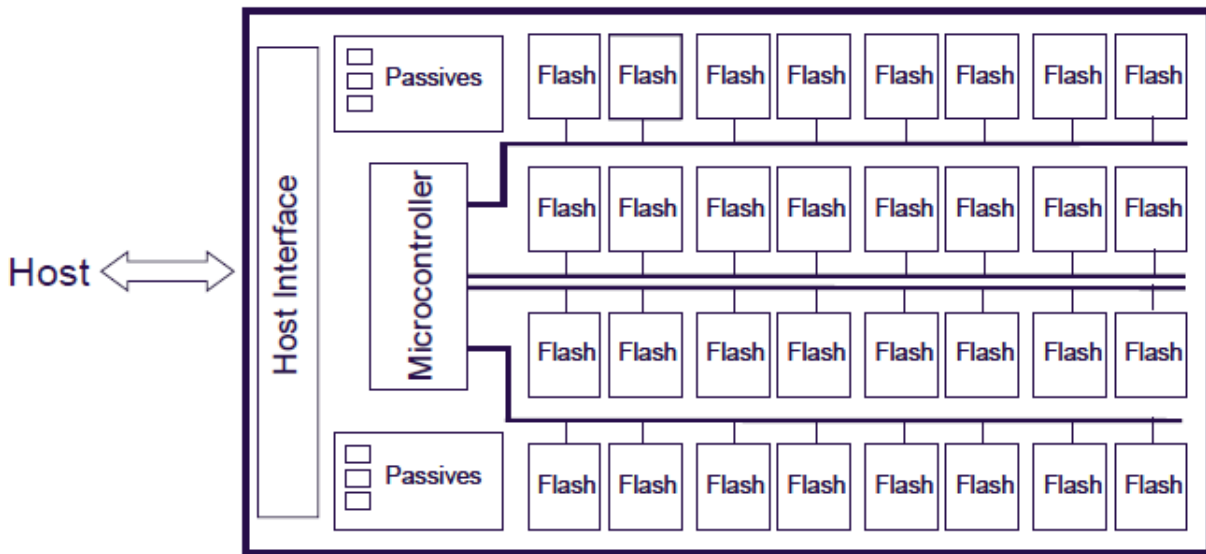
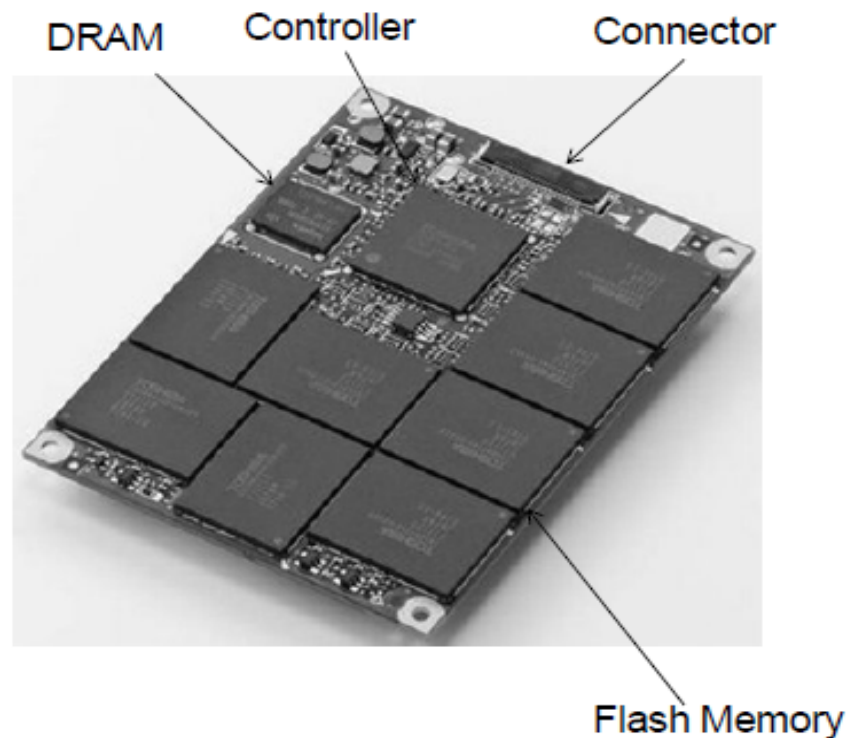


Fig. 2.32. Block diagram of a SSD



Id. at 39, 14 (Fig. 1.16). The flash memory module typically has the following components in addition to the transistors (cells) in the memory array, generally for input/output from the array, including buffers and addressing logic (*e.g.*, word lines and bit lines), detection circuitry and generation and application of voltages for reading, writing and erasing:

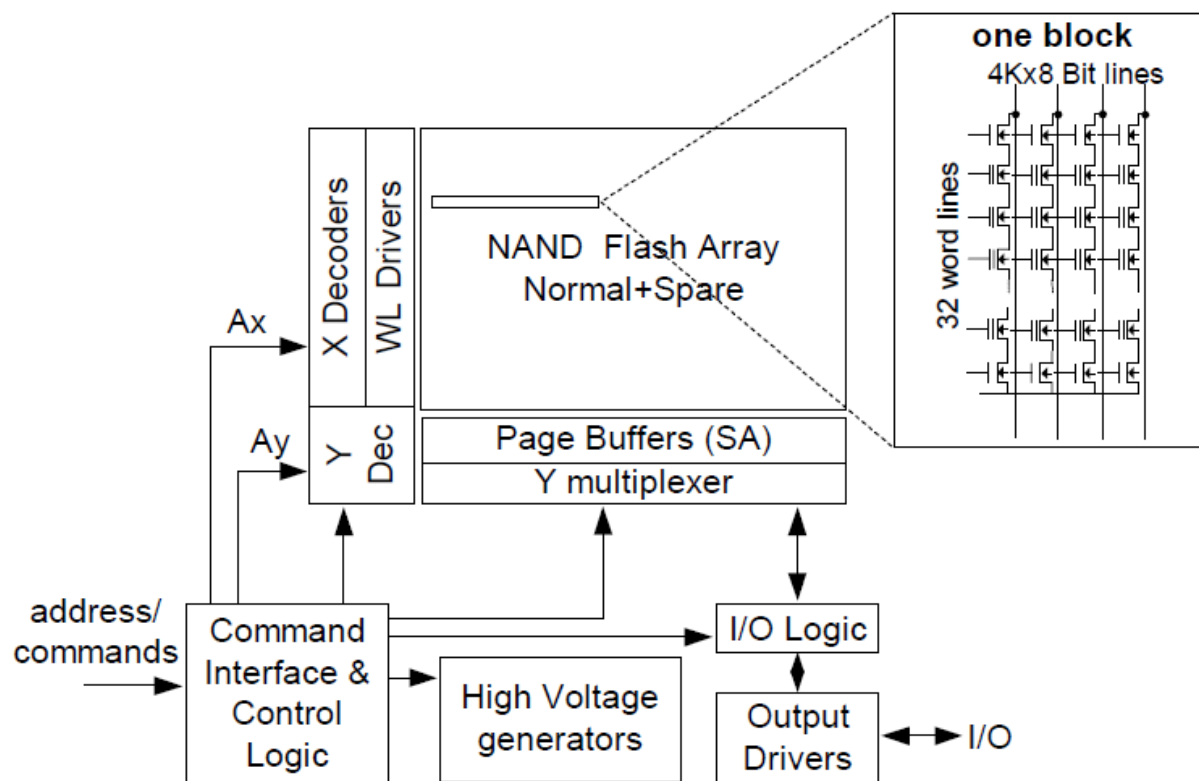


Fig. 15.1. NAND Flash memory block diagram

Id. at 424. A POSITA would understand this as a NAND flash module, with circuitry necessary for physical NAND flash cell organization, reading, writing and erasing. The memory device controller typically has a microprocessor and various memory to support its functions of interfacing with a host computer and to maximize transfer

speed, data integrity and information retention. *Id.* at 40. In addition to ECC processing, a flash file system (FFS) is implemented as firmware operating in different functional sublayers for wear leveling, garbage collection and bad block management. *Id.*

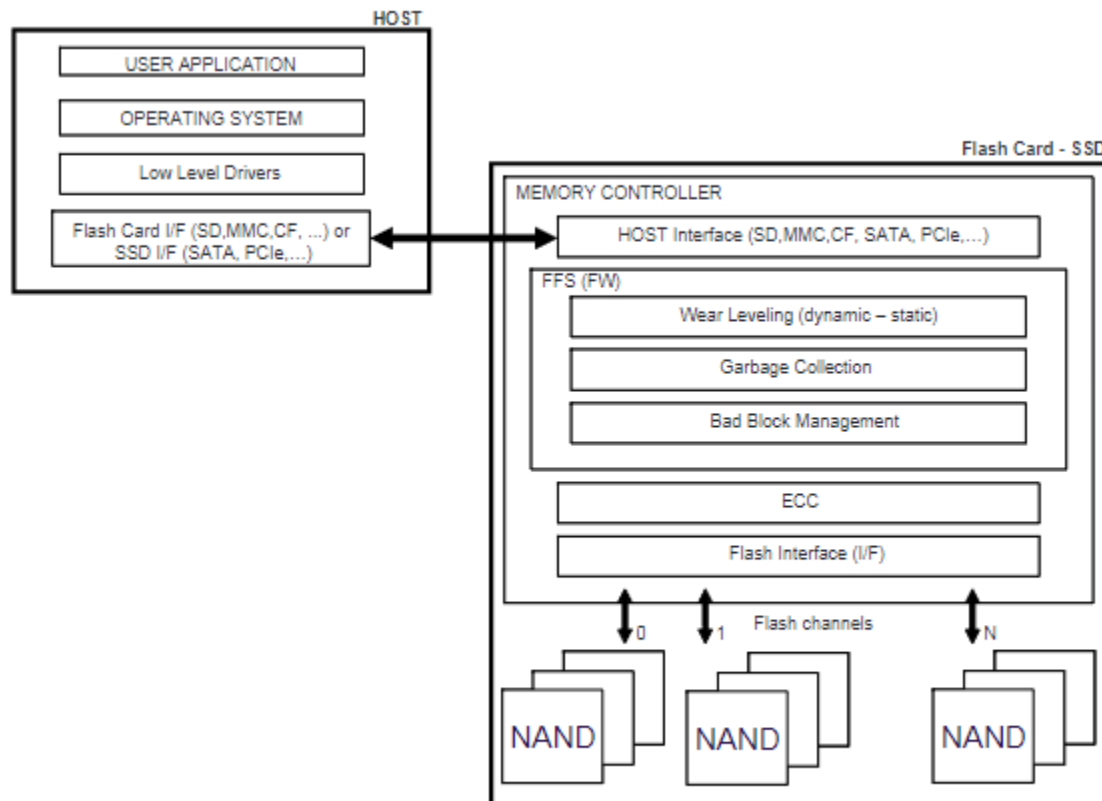


Fig. 2.33. Functional representation of a Flash card (or SSD)

Id. To perform these functions, tables, implemented as a software Flash Translation Layer or FTL, are widely used in order to map sectors (traditional organization for hard disk drives) and pages from logical (host-visible) locations (addresses) to physical locations (addresses) in the flash memory. *Id.* at 40. This is illustrated as

follows, with the upper row being the host-logical view of the memory and the lower row being the physical flash memory:

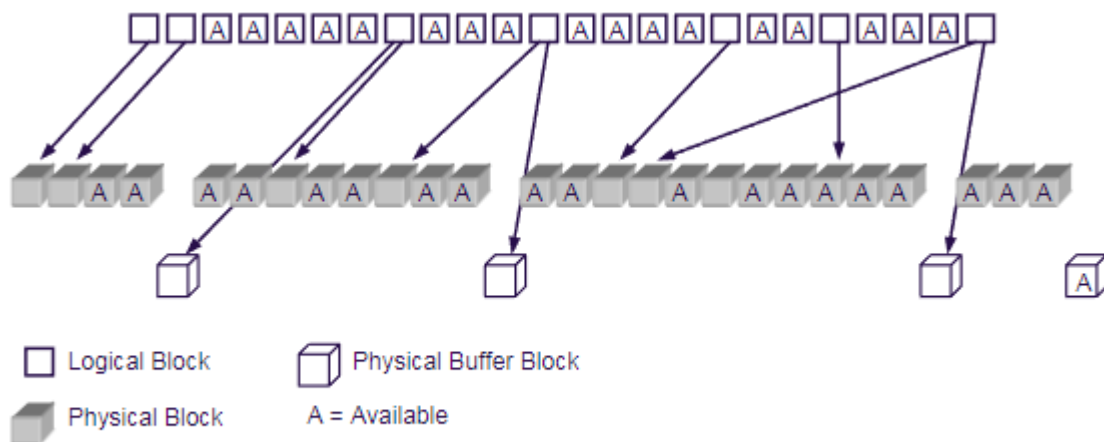


Fig. 2.34. Logical to physical block management

Id. at 41. Meeting the problem that flash memories lack the “update in place” capability of other memories such as magnetic disk, because entire blocks must be erased before even a single page can be rewritten, the FTL allows emulation of independent modification of logical pages through “logical to physical remapping.” *Id.* at 492; also “L2P”). When the file system sends a command to write or update a certain logical page, the FTL writes to a different available physical page than currently mapped and updates the table storing the logical to physical mapping. *Id.* The physical page containing the old data is marked as invalid. *Id.* Thus, in NAND flash operation, every update or write to a previously mapped physical location that has content must be remapped to another empty or erased location.

Through the process of garbage collection, valid data are copied to new locations and the old location erased to free up memory for new writes. *Id.* at 493. Although not shown in the diagrams above, to perform these functions, including the interfaces and processing for ECC and FFS processing, there is necessarily memory inside or otherwise efficiently available to the memory controller to accommodate interfaced and otherwise processed data (user and system, including ECC as appropriate). That memory size must be at least as large as a page, since this is the smallest unit read or written from the flash memory.

28. **Wear leveling and FTL operation.** It is considered important to keep flash memory aging (the number of program/erase cycles) of each block in a NAND flash memory system to a minimum and as uniform as possible, considering that the expected (in industry specifications) life of SLC NAND memories is 100K cycles and that of MLC NAND memories is 10K program/erase cycles. Micheloni at 492. These metrics often are set forth as “*Endurance*[,] the term used to describe the ability of a device to withstand [stresses of normal processes of writing and/or erasing cells that will eventually degrade the properties of the memory or disturb the contents of the memory] and is quantified as a minimum number of erase-write cycles or write-read cycles that the chip can be expected to survive.” (NONVOLATILE MEMORY TECHNOLOGIES WITH EMPHASIS ON FLASH: A COMPREHENSIVE GUIDE TO UNDERSTANDING AND USING NVM DEVICES (Joe E. Brewer & Manzur Gill eds.

Wiley-IEEE Press 2008) [“Brewer,” Ex. 1026] at 13.) This is a rating guarantee of the manufacturer; most often, “the device will continue operating well after the max cycling spec has been exceeded.” *Id.* at 30.

Flash memories’ blocks can only be programmed and erased a limited number of times. This is the reason why a good FTL should also implement a *wear leveling* algorithm. A wear leveling software module makes sure all NAND blocks [of the same endurance expectation] are cycled an identical number of times, avoiding some blocks ending their cycle life [and compromising the useful life of the system] while others being cycled only a few times. Were not a wear leveling algorithm present, blocks with frequently changed data would endure much more cycles than blocks containing long-lived data. (Michelsoni at 494.)

Wear Leveling techniques rely on the concept of logical to physical translation: that is, each time the host application requires updates to the same (logical) sector, the memory controller dynamically maps the sector onto a different (physical) sector, keeping track of the mapping either in a specific table or with pointers. The out-of-date copy of the sector is tagged as both invalid and eligible for erase. In this way, all the physical sectors are evenly used, thus keeping the aging under a reasonable value. (*Id.* at 41.)

There are two levels of wear leveling. In the *first level wear leveling* [also called dynamic] new data [directed by the host to a logical block] is programmed into a free [physical] block with the fewest write/erase cycles. Two techniques to achieve this are the chain method and the array method. (*Id.* at 494.)

In the *chain method*, the pool of available blocks is organized in a chain A simple round robin algorithm selects one block after another. When a block is erased, it is added to the chain. (*Id.*)

In the *array method* an age vector stores the number of each block's program/erase cycles. This information can be used by the wear leveling algorithm to ensure a uniform exploitation of all the blocks. (*Id.*)

In the *second level wear leveling* [also called static] from time to time blocks containing long-lived data are moved (copied) into other blocks even though they never received an update command. The original block can then be used to store new incoming data. The second level wear leveling is triggered when the difference between the maximum and the minimum number of cycles per block reaches a predetermined threshold. (*Id.*) A consequence of the erase-before-write nature of NAND flash in which a logical page may be written only to an empty physical page of a NAND block (or the block must be erased first), updates to [logical] data are remapped (possibly including non-updated or "valid data") to physical pages in other physical blocks (which may need to be erased), leaving the data in the previously mapped location obsolete or "invalid." "Garbage collection" function of FTL is the consolidation of valid data in a new physical block while the prior physical block is erased to open up space for new data:

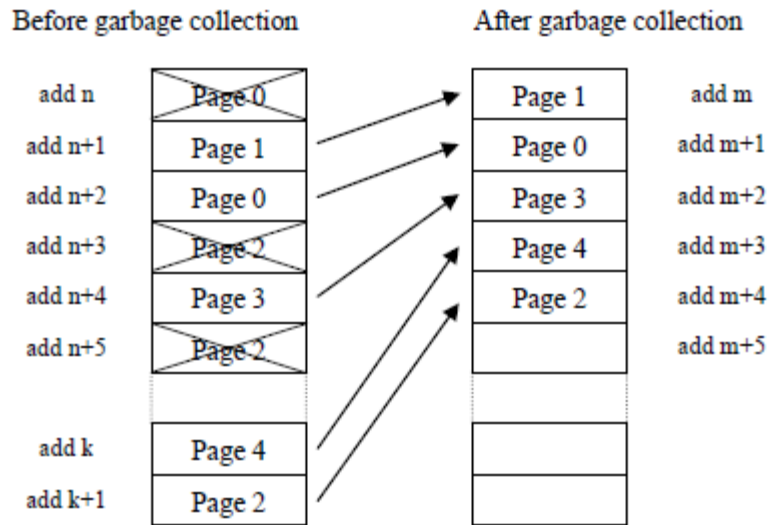


Fig. 17.7. Garbage collection

(Michelsoni at 493.) Higher garbage collection efficiency, measured by the ratio of number of invalid pages to the total number of pages in a physical block, is desirable in order to reduce the number of rewrites of valid data while erasing and opening the block to new data. (*Id.*) To improve such efficiency across a memory system, various strategies may be adopted to cluster all “hot data” (frequently updated data, that is, logical addresses) that will soon be erased and all “cold data” (non-hot data). (*Id.* at 494-97.) As described below, where a NAND flash memory system includes different types of memory operating with different expected endurance, there are many strategies to avoid premature failure (or rated end-of-life) of physical blocks that could degrade system operation.

29. **Bad block management.** “Bad blocks” are “blocks which contain one or more locations whose reliability is not guaranteed.” Thus, a “*Bad Block*

Management (BBM) module creates and maintains a map of bad blocks.” “[T]his map is created during factory initialization of the memory card, thus containing the list of bad blocks already present during the factory testing of the NAND Flash memory modules.” *Id.* at 42. The map is updated as the memory device is used when a block is determined to be bad (*id.*) and thereby retired.

B. Prior Use of SLCs and MLCs in the Same Memory System

30. SLCs are “single level,” which allows storage of one bit of information in a cell by allowing a single, higher threshold voltage associated with greater trapped floating-gate charge associated with a logical “0” state. This is one of the two states for SLCs; the other state is the erased state characterized by a very low threshold voltage (largely absent of any trapped floating-gate charge) and represents the logical “1” state. MLCs, first developed in 1992-97 (Brewer [Ex. 1026] at 595-96), allow storage of multiple (m) bits of information in a cell by allowing charging of the floating gate to 2^m threshold voltages, each threshold voltage corresponding to a logical m-bit binary number. (The assignment may be by Gray code [adjacent threshold voltage representing only one digit transition] to minimize overlap of corresponding threshold voltage curves. In Figure 12.6 below, the bit-pair transitions would instead be, for example, in left to right order: 11 10 00 01.)

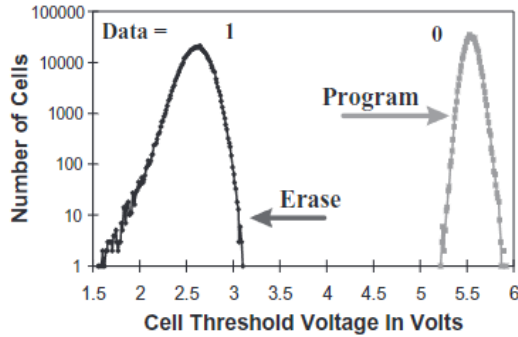


Figure 12.5. Single-bit/cell array threshold voltage histogram.

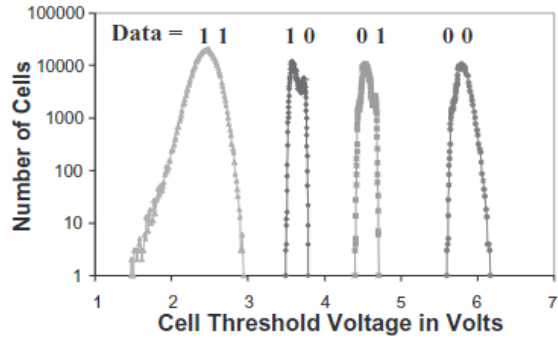


Figure 12.6. Two-bit/cell array threshold voltage histogram.

Id. The distributions do not exist simultaneously for any single cell but show for a particular logical (binary) value the number of cells in a real array (batches of chips) that would have a specific threshold voltage. In Figure 12.6 above, the most significant bit (MSB) for a cell is part of the so-called upper page and the least significant bit (LSB) for a cell is part of the so-called lower page. For the first two possible codes (11 and 10), the MSB is 1 and for the latter two codes (01 and 00) the MSB is 0. For the Gray code approach, for the first two possible codes (11 and 10), the MSB is also 1, and for the latter two codes (00 and 01) the MSB is 0. Thus, the writing of MLC is executed for two “levels” of pages. This distinguishes multi-level cell (MLC) NAND flash from single level (SLC) NAND flash. (The distinction does not apply to other non-volatile memory, and I have never heard of “SLC” applied to other than NAND flash memories.)

31. Obviously, the greater m is, the more crowded the assigned voltage thresholds are in the threshold “window” (divided into m regions), and the greater the opportunity for error (non-uniformity) in writing the cells.

32. The figure below is from Yu Cai et al., *Program Interference in MLC NAND Flash Memory: Characterization, Modeling and Mitigation* (IEEE 2013) (“Cai,” Ex. 1030) at 124:

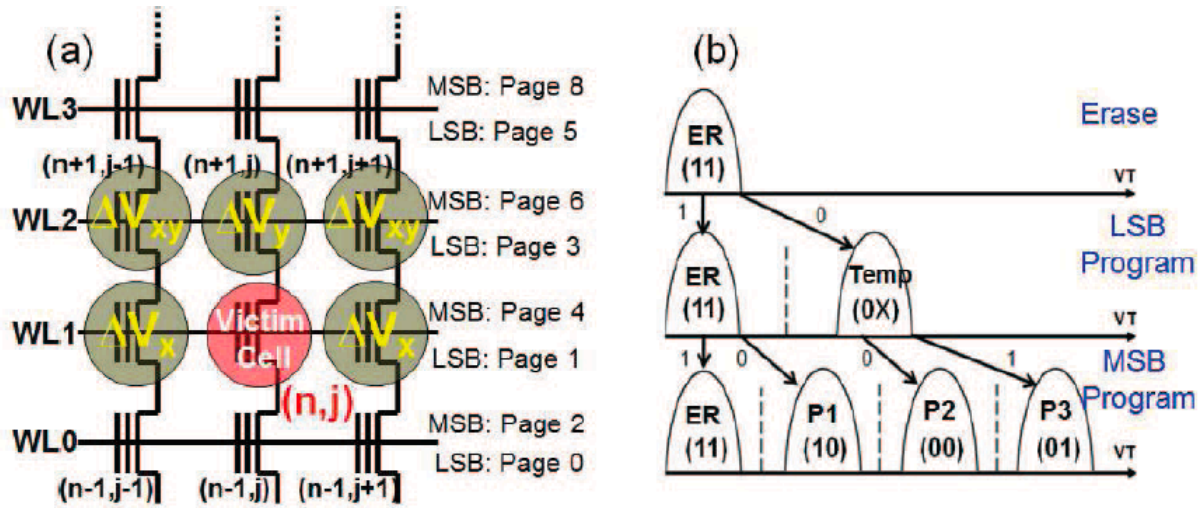


Fig. 1. (a) All-bit-line NAND flash block architecture with victim cell circled in red and aggressor cells circled in grey; (b) Two-bit MLC flash programming scheme. Cell states are encoded in format (LSB, MSB)

33. In the above figure of a flash *block*, each row (word lines WL0, WL1, etc.) contains two pages, an MSB page and an LSB page, for this MLC architecture. Pages can be 4kB in length, or longer. The bit-line is at the top of each column of flash transistors.

34. “A contemporary two-bit-per-cell MLC flash cell is programmed to a desired value in two stages, as shown in Fig. 1(b). After an erase operation (applied to the page the cell resides in), the cell starts out at the erased state (ER). If the LSB of the cell is programmed as 0 (during the programming of the corresponding LSB page), the flash cell moves into a temporary program state (Temp). Otherwise, it

remains in the ER state. During the programming of the corresponding MSB page, if bit value 1 is programmed into the cell's MSB, the flash cell either remains in the ER state or moves from the Temp state into the P2 state.” (Cai at 124.)

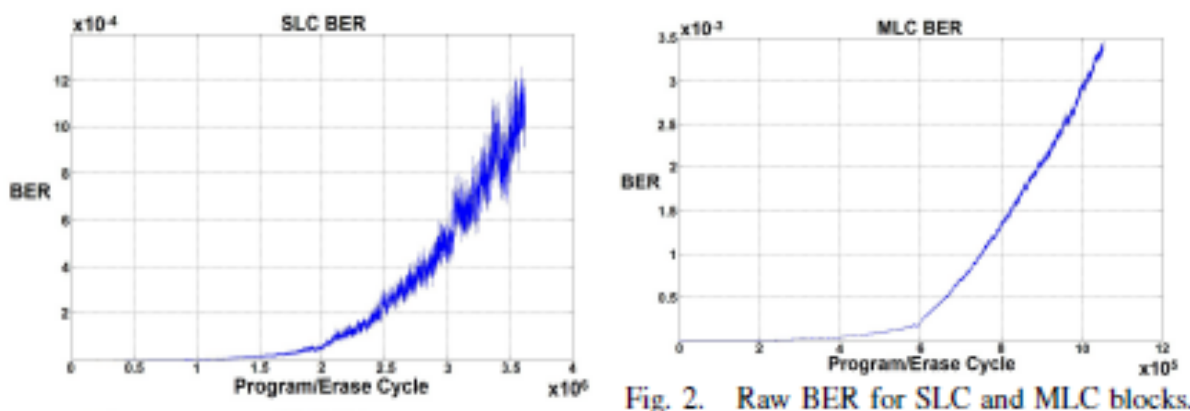
35. “In MLC NAND flash memory, the logical value stored in a memory cell is determined by the threshold voltage range into which the cell's actual threshold voltage falls. MLC NAND flash chips use incremental step pulse programming (ISPP) to set the state of the cell's threshold voltage to a value within the voltage range corresponding to the logical value to be stored. However, due to coupling capacitance between neighboring floating gates, the programmed threshold voltage of a cell may change when neighbor cells are programmed later. Program interference is the phenomenon in which the threshold voltage of a flash cell, called the victim cell, unintentionally changes (*i.e.*, gets disturbed) while another cell, called the aggressor cell, is being programmed. If the change in threshold voltage due to program interference causes the victim cell's voltage to shift to a different threshold voltage range, then the victim cell's value becomes incorrect, leading to an error when the cell is read. An example victim cell and its aggressor cells are shown in Fig 1(a).” (Cai at 125.)

36. “Incremental step pulse programming (ISPP) achieves fast program performance under process and environmental variations while still keeping a tight programmed cell V_t distribution. ISPP effectively covers process variations by

allowing “easily” programmed cells to be programmed with a lower program voltage and difficult-to-program cells to be programmed with a higher program voltage. After an initial 15.5 V program pulse, each subsequent pulse (if required) is incremented in 0.5 V steps up to 20 V. Since sufficiently programmed cells are automatically switched to the program inhibit state in the verification step, easily programmed cells are not affected by the higher program voltages.” (Kang-Deog Suh et al., *A 3.3 V 32 Mb NAND Flash Memory with Incremental Step Pulse Programming Scheme*, 30 IEEE J. Solid State Circuits 1149, 1154 (Nov. 1995) [Ex. 1031].) This pulsing, in addition to the phasing of multi-page writes to MLC blocks, required different circuitry than the pre-existing “SLC” NAND flash, so that a POSITA would understand that an SLC module is different from an MLC module not just in capacity, but in circuitry necessary for the different reading, writing and erasing.

37. A high voltage applied to the bit-line causes no additional charge to be added to the floating gate when a programming pulse is applied to the word line (control gates). This is done if the programmed bit is supposed to remain 1, or if the cell is already programmed correctly after a certain step in the ISSP process. On the other hand, a low voltage applied to the bit-line enables the programming pulse to cause an increase in charge in the floating gate, lowering the threshold voltage. In single-level cells (SLC), this is how the logic 0 is written into the cell.

38. The ability of a floating gate to retain a specific amount of charge degrades slightly after each block erase operation. After, for example, 10,000 such erase cycles, it can happen that certain MLC cells cannot retain the proper threshold voltage in their floating gates for sufficiently long. Obviously with SLC, the number of erase cycles can be much larger since the difference between the high threshold and low threshold is at least half of the voltage supply, and hence the floating gate can still distinguish the proper threshold voltage for much longer. In one empirical study published in 2010, Eitan Yaakobi et al., *Error Characterization and Coding Schemes for Flash Memories* (2010 IEEE Globecom Workshops, Miami, FL at 1856-60) (“Yaakobi,” Ex. 1032), it was remarked that “the observed lifetimes of the tested flash blocks were much longer than the lifetimes specified by the manufacturer . . . [a]n SLC block can typically tolerate 10^5 to 10^6 erasures whereas an MLC block 10^4 . . .” Raw bit error rates were presented:



These are factors towards the greater error probability and wear for MLCs compared to SLCs, leading to the known trade-off between shorter erase-write cycle life of MLCs but their lower cost per bit of memory.

39. By 2010, the comparative characteristics of SLCs and MLCs (often understood to be 2 bits/cell, compared to TLC [3 b/c] and QLC [4 b/c] were known and summarized in the following graph (Micheloni at 9):

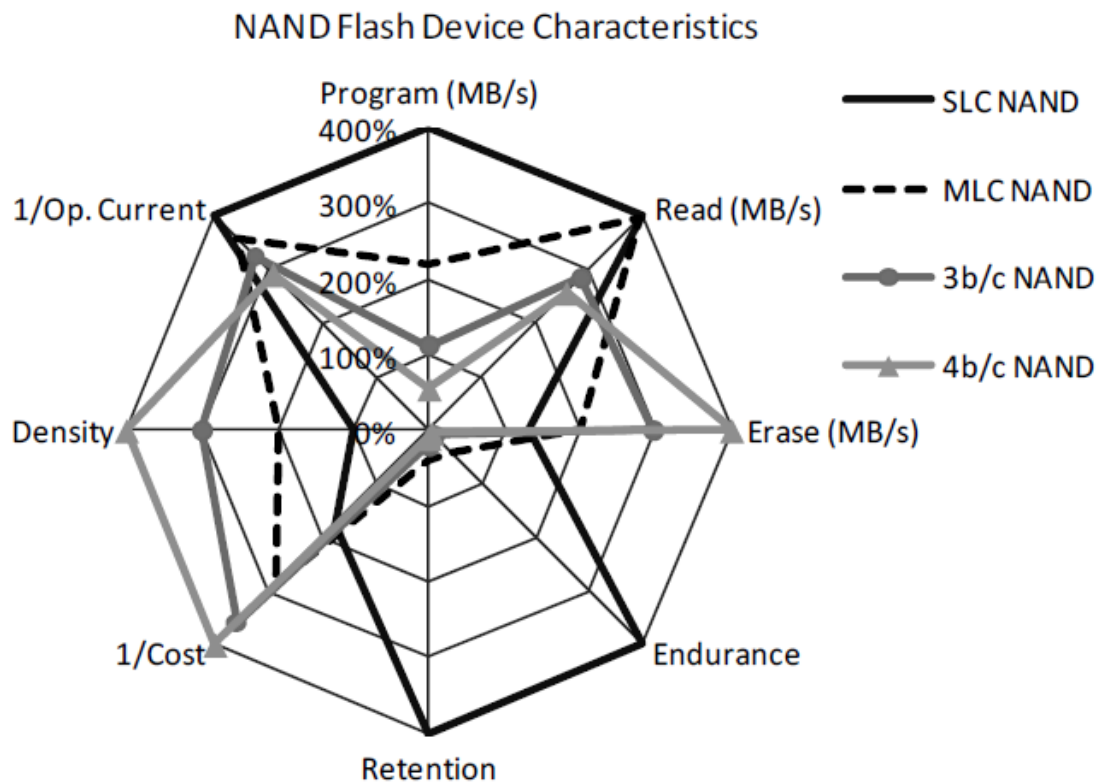
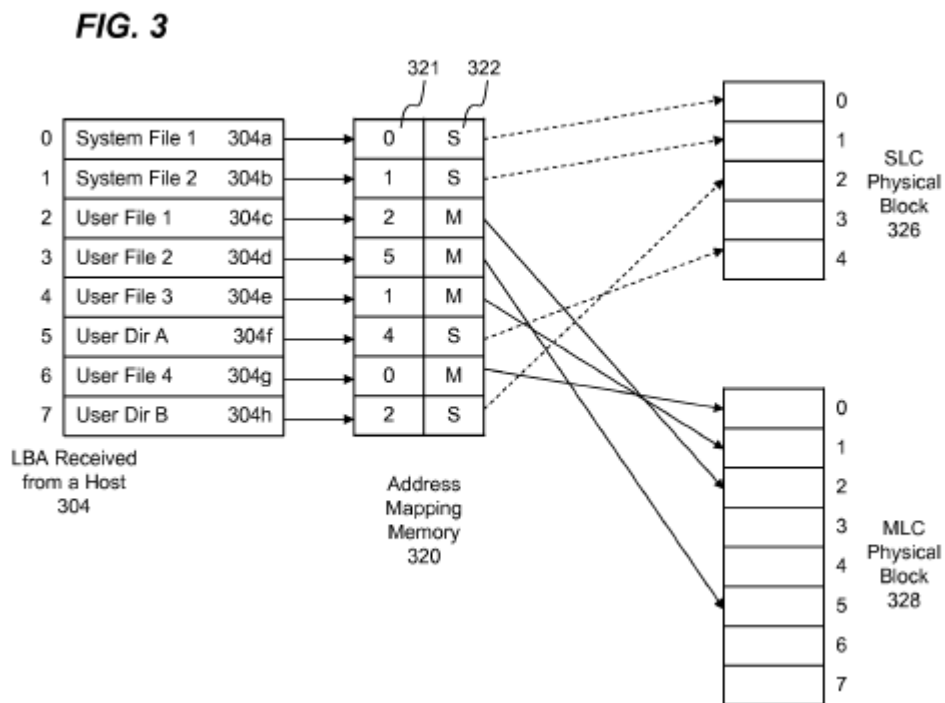


Fig. 1.10. NAND device characteristics (Source: Forward Insights)

Notably, the rated “endurance” for MLCs is an order of magnitude less than for SLCs.

40. This trade-off favored the use of MLCs for infrequent write applications such as memory cards for consumer digital cameras. Enterprise use demanding endurance and quality were often met with SLC NAND. (Michelsoni at 169.) However, it was recognized that, for enterprise use, frequently programmed or critical data (such as chip or memory system information) might be advantageously held in an SLC memory module and less frequently programmed data (such as user data) held in an MLC system. *See* Chris Evans, *Enterprise MLC: How flash vendors are boosting MLC write endurance*, ComputerWeekly.com (June 3, 2011) (Ex. 1033). The hybrid SLC and MLC modules approach had been proposed by 2008. *See* Nelson Duann, SLC & MLC Hybrid (Silicon Motion Flash Memory Summit, Santa Clara, CA, Aug. 12, 2008) (“Duann,” Ex. 1034). U.S. Patent No. 8,078,794 to Lee et al., published Sept. 4, 2008 (“Lee ‘794,” Ex. 1041) disclosed the use of distinct “SLC flash memory array 226 [that] may include at least one flash memory chip” and “MLC flash memory array 228 [that] may include at least one MLC flash memory chip” (5:10-13, Fig. 2 [the “hybrid solid state device (SSD) 200 also includes a controller 210, a data read cache 203 and a data write cache 204, and a logical-to-physical address mapping memory 220]”, 4:11-20). Fig. 3 shows mapping by logical block address (LBA) of system files and user directories (both frequently written) to SLC physical blocks 326 and user files to MLC physical

blocks 328, where the physical blocks are physically part of their respective memory arrays (chips).



Figs. 5A-5B show decisions to remap system files and user directories to the SLC blocks.

41. The hybrid use of SLC and MLC memory chips/modules to take advantage of their respective advantages was also applied to wear leveling prior to June 19, 2011. Sutardja, U.S. Patent Appl. Pub. No. 2008/0140918 “Hybrid Non-Volatile Solid State Memory System” (“Sutardja,” Ex. 1042). The Sutardja architecture included a first solid state nonvolatile memory 204 with relatively large capacity (such as MLC with a device usable lifetime of 10K write cycles) and a second solid state nonvolatile memory 206 with lower density and greater lifetime

(such as SLC with a device usable lifetime of 100K write cycles). *Id.* ¶¶ [0006], [0106]-[0108]. Multiple, detailed criteria are provided for mapping data to one or the other memory, ranging from write cycle lifetimes to ratios of use to frequencies of writes, using a write leveling software module and a write monitoring software module. *Id.* ¶¶ [0011]-[0083]. Detailed algorithms are disclosed for application of these criteria. *Id.* Figs. 7A-7E, 8 and explanations.

42. Similarly, Moshayedi U.S. Patent Appl. No. 2009/00327591 (“Moshayedi,” Ex. 1043), specifically showed four channels (banks) Fig. 1 (108) each connecting “five chips of flash (e.g., the first chip can be a K9K8G08 SLC flash, and the rest can be K9G8G08 MLC flash)” (¶ [0038]) where the drive

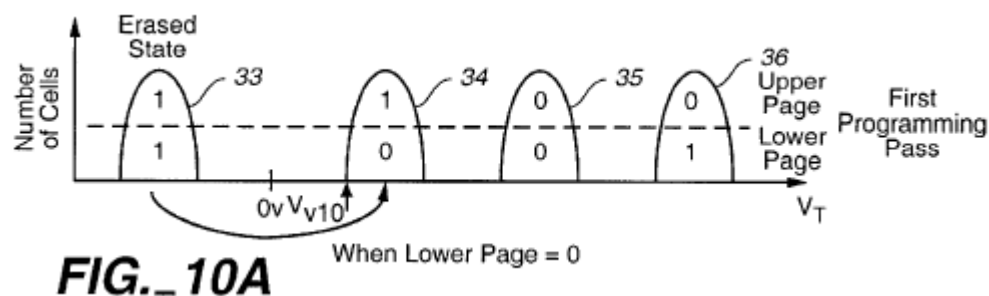
keeps track of the number of times that data for each logical block address (LBA) has been written to the flash memory, and determines whether to store newly received data associated with a particular LBA in SLC flash or in MLC flash, depending on the number of writes that have occurred for that particular LBA

(¶ [0024].) In an example, the drive may use an erase count list:

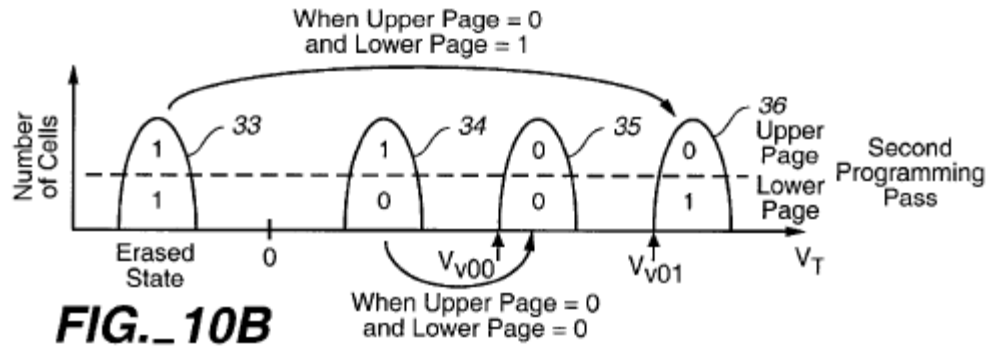
once a block in MLC flash reaches a threshold erase count (e.g., 500), the next write operation to that block triggers a swap where the data from the MLC flash block is written to a block in SLC flash. In this manner, data for an LBA that is frequently written and causes frequent erasures is moved to SLC flash which can perform more erase cycles than MLC flash.

(¶ [0032].) There is distinguished “‘dynamic data’ . . . which is subject to more frequent write or rewrite operations (e.g., operations which may reduce the reliability of the associated data segments and/or data blocks) . . . and can be relocated to data blocks with less wear [“static data”] (e.g., as determined by the number of data read errors associated therewith).” (¶ [0034].)

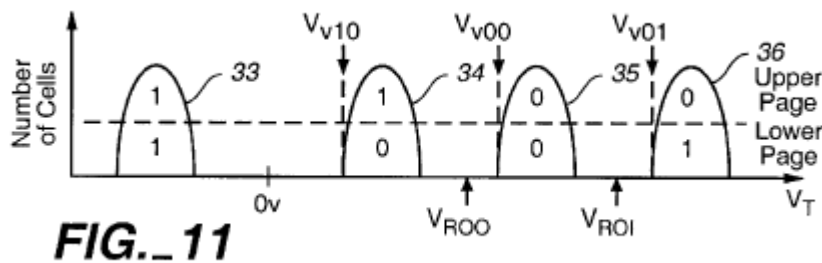
43. Meanwhile, memory system manufacturers such as SanDisk provided memory systems including MLC blocks that could be configured to function in one-bit (two-state) mode, later sometimes called “pseudo-SLC” or “pSLC,” this was disclosed as early as September 24, 2002, in the issuance of U.S. Patent No. 6,456,528 to Chen for “Selective Operation of a Multi-State Non-Volatile Memory System in a Binary Mode” (“Chen ‘528,” Ex. 1040), showing Figs. 10A and 10B for respectively in two passes programming a “lower page” (least significant bit)



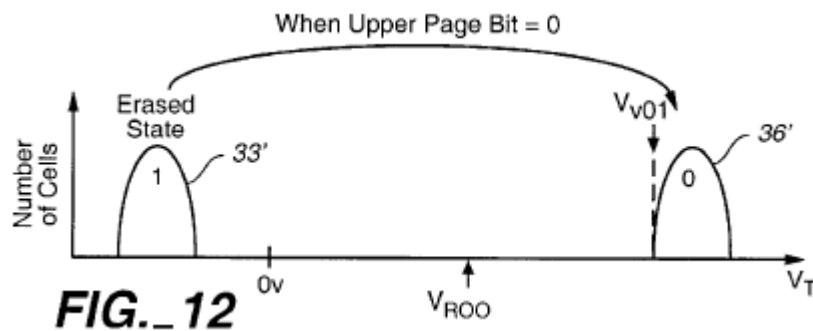
and an “upper page” (most significant bit)



resulting in the two-bit (MLC) distribution of Fig. 11:



To program for a one-bit (two-state) mode, only one programming pass is used, in this case, the “upper page” resulting in greater separation of the voltage threshold distributions as shown in Fig. 12:



Recognizing the retention advantage of the two-state mode because of the increased margin and fewer programming passes (writes, wear) (9:52-10:30), Chen’s

suggested uses included operating frequently rewritten logical blocks (user data, directories, system overhead data as block cycle counts) in two-state mode and less frequently in multi-state mode (10:41-52). In U.S. Patent Appl. Pub. No. 2010/0172179 to Gorobets (“Gorobets ‘179,” Ex. 1044), disclosing adaptive use of spare blocks in binary-operated and MLC pools, it was observed of binary operation (lower page/bit): “While this approach does not fully optimize the range of the threshold window as in the case of a single-level cell (“SLC”) memory, it has the advantage of using the same demarcation or sensing level as in the operations of the lower bit of the MLC memory” (¶ [0083]). Such configuration was disclosed in U.S. Patent No. 8,634,240 to Gavens et al., published April 28, 2011 (“Gavens,” Ex. 1045, 8:19-21 [“[a] memory chip 100 includes a memory array 200 of memory cells with each cell capable of being configured as a multi-level cell (“MLC”)]”) and the multiple patents it incorporated by reference. The benefits of split mode operation include similar trade-offs as between “actual” SLC and MLC modules or chips as in Lee ‘794, although the program cycle and processing overhead may be the same between an MLC operating in MLC mode and one operating in SLC mode, and the per-cell cost may be the same. In an early appraisal of relative (Samsung) performance, where “pseudo-SLC” is “MLC_{LSB}”:

Table 1: Performance comparison of different types of cell programming (us)

Operation	SLC	MLC_{LSB}	MLC_{BOTH}
Read (page)	399	409	403
Write (page)	417	431	994
Erase (block)	860	872	872

Sungjin Lee et al., *FlexFS: A Flexible Flash File System for MLC NAND Flash Memory* (2009 USENIX Annual Technical Conference, San Diego, CA, June 14-19, 2009) (Ex. 1036). Yaakobi at 1859-60 (Ex. 1032) remarked that the 2-state MLC technique can be made “adaptive by initially using the MLC device in its native mode, and then switching to SLC mode after a certain number of iterations, when the BER has become larger due to device wear.” Another paper at the same conference, Ashish Jagmohan et. al., *Adaptive Endurance Coding for NAND Flash* (2010 IEEE Globecom Workshops, Miami, FL at 1841-45) (Ex. 1035), shows that 2-state operation of MLC has a BER after 20K p/e cycles an order of magnitude better than native-mode operation of MLC, but definitely not as good as SLC. This is confirmed in current commercial offerings that compare SLC as rated for 50-100K p/e cycles, pSLC for 20-30K and TLC for 3K. *E.g.*, Smart Modules Technologies, *Pseudo-SLC For Flash Storage*, <https://www.smartm.com/technology/pseudo-slc-pslc> (visited Aug. 18, 2024); Cactus Technologies, *Industrial pSLC Flash Storage*, <https://cactus-tech.com/products/industrial-pslc> (visited Aug. 18, 2024).

44. Prior to June 19, 2011, and since, system design considerations included different deployment of SLC or pSLC versus MLC memory, such as is exemplified in Segal et al. U.S. Patent Appl. Publ. No. 2011/0271043 Fig. 3 (Ex. 1048), for bad block replacement by adjusting MLC and SLC spare pools, including optimizing the diversion of MLC spares for SLC-mode spare usage, and similarly in Gorobets '179 for MLC and pSLC spares.

C. “Data Integrity Checking” in NAND Flash Memory Systems

45. In addition to the threshold voltage programming check mentioned above for readable writing (at the memory module level), there may be some kind of “integrity check” of whether the data written into a NAND flash word (the smallest group for writing, connected by a word line), is correctly written. Different actions may follow, including retries (although it would not be efficient to erase and retry the same physical location). Excessive errors may lead to flagging a block as a “bad block” with respect to bad block management (BBM). In most commercialized NAND flash memory systems, checks of user data written are performed indirectly (if at all until end-of-rated-life) as part of the processing of error correction code (“ECC”) schemes. ECC schemes have long been used for automated correction of digital communications, partly because such communications may suffer intervening interruptions such as radio frequency interference. Such interference also occurs in NAND flash memory due to the effects of writing into

adjacent cells. Also, a limited number of bad cells in a page can be tolerated if ECC is used, thereby lengthening the lifetime of the flash memory by not erasing and writing elsewhere. In a typical NAND flash memory system, an ECC is generated for the data to be written, which may be stored with that data as written (as shown) or in a separate memory associated with that data. When the written data is read, a new ECC may be generated and compared to the stored ECC to detect and correct erroneous data bits. Following is a diagram of an on-chip ECC from Brewer (Ex. 1026) at 88 showing the writing of user data D into the memory with generation by an ECC encoder of ECC correction bits C, stored in the memory, and when the written data WD is read with the original C, a new code S is generated to indicate

which bits of WD are different from D to correct those erroneous bits to output D:

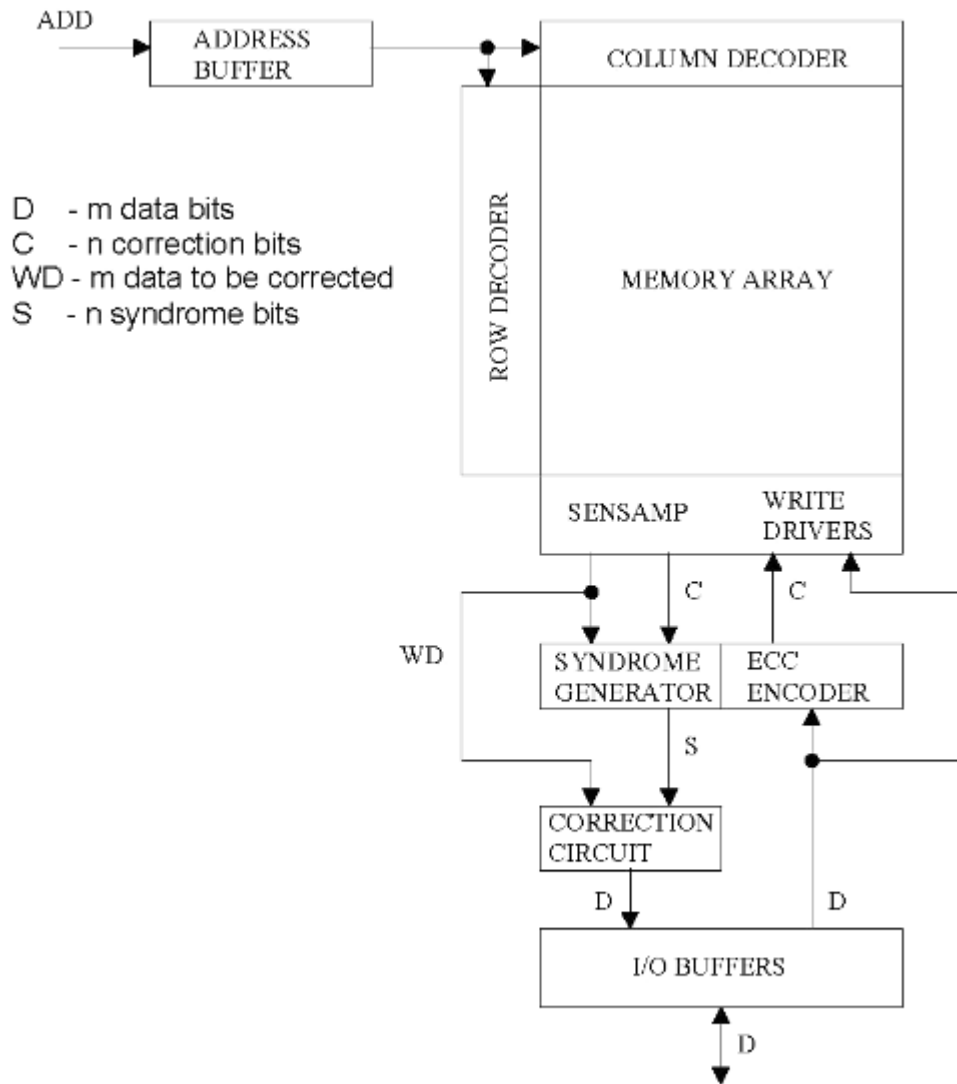


Figure 3.20. On-chip ECC for Flash memory.

With enough processing resources and ECC space needed (which may be external to the chip) to correct multiple error bits, nothing needs to be done with the data (WD) with the error bits in the memory system, as what is read from that data will be corrected to what was sent to be written (D). However, as a flash memory ages, the number of required error correction bits for a block may exceed the storage

available for error correction bits and that block will have to be retired. As more and more blocks are retired, at some point the flash memory is no longer useful.

46. Gavens (Ex. 1045), published April 28, 2011, is one of a portfolio of SanDisk patents, many incorporated by reference in the context of a NAND flash memory system including “[a] memory chip 100 [that] includes a memory array 200 of memory cells with each cell capable of being configured as a multi-level cell.” It discloses the use of such ECC similarly to that shown in the preceding paragraph:

In the types of memory systems described herein . . . the integrity of the data being stored is maintained by use of an error correction technique. Most commonly, an error correction code (ECC) is calculated for each sector or other unit of data that is being stored at one time, and that ECC is stored along with the data. The ECC is most commonly stored together with a unit group of user data from which the ECC has been calculated. The unit group of user data may be a sector or a multi-sector page. When this data is read from the memory, the ECC is used to determine the integrity of the user data being read. Erroneous bits of data within the unit group of data can often be corrected by use of the ECC. (3:27-39.)

As data is received from a host, a page of data is staged in the controller 102 and its ECC 76' is computed by the ECC processor 62. The data page incorporating the ECC is then written to the memory array 200. Typically, when the data page is read, the data page is latched in the data latches 430 and shifted out of the I/O circuits 440 to the controller 102. At the controller 102, the data page's existing ECC is compared to

a second version of the ECC computed on the read data. The ECC typically includes an error detection code ("EDC") for rapid detection of any error in the data page. If the EDC indicates the existence of any error in the read data page, the ECC is invoked to correct erroneous bits in the read data page. (12:64-13:8; *also* 29:33-62.)

Although this is the preferred approach for checking and correcting for errors (when possible, given the ECC overhead choice⁸), Gavens also discloses an alternative of direct comparison of the data written into memory (and then read back in “post-write

⁸ “[F]or most of the lifetime of the device, the ECC is only marginally utilized, resulting in its large overheads being wasted and realizing no real benefits.” (Gavens 3:67-4;2.) Thus, alternatives are disclosed (a) to “adaptively rewrite data from a higher density memory portion [“each memory cell storing more than one bit of data”] to a lower error rate memory portion [“each memory cell storing one bit of data”] to control error rate (4:9-5-30); (b) “enhanced post-write-read error management” where the management of (a) is not applied until a “hot count” tracking “the endurance or the number of times the erase block has been cycled through erase and program operations” has passed a predetermined threshold (5:31-6:2) and (c) “accelerated post-write-read” on a small sample related to ECC statistics (6:3-38). These design alternatives are supported by statistics and sources of error set forth graphically at Figs. 10A-13 and explained at 13:17-16:13.

read”) with a cached copy of what was to be written, among embodiments under the rubric “Adaptively Rewrite Data from a Higher Density Memory Portion to a Lower Error Rate Memory Portion to Control Error Rate” (15:59-61).

47. In one group of adaptive rewrite embodiments (Figs. 14 and 15 and associated text), user data is written to the higher density (*e.g.*, MLC) portion as a “first copy.” Then, in a “post write read,” there is a check for errors “either by comparison with the original copy which may be cached **or** by checking the EDC portion of the ECC” (16:47-49 [emphasis added]). If the number of error bits is less than a predetermined amount, the first copy is deemed valid and subsequent reads of the data page will be from that copy with any errors corrected by ECC. (16:50-56.) If the number is greater, then a copy is written to the lower error rate (more robust, *e.g.*, SLC) portion; this takes place in one embodiment from a cached copy or in another embodiment from application of ECC to the first copy. (17:9-14.)

48. In a second group of adaptive rewrite embodiments (Figs. 16 and 17 and associated text), the more robust portion is further divided into a first section for caching incoming data (simultaneously with a “first copy” written into the higher density portion) and a second section for storing rewrites from the higher density portion. (18:21-29.) In one of these embodiments,

the memory array is provided with a set of data latches on an integrated circuit chip, the checking of the error bits in the first copy is

accomplished by loading the first copy and the cached copy into the set of data latches and making a comparison at the set of data latches.

(18:30-35 [emphasis added]; 18:40-44 [details].) If the number of error bits is less than a predetermined amount, the first copy is deemed valid and subsequent reads of the data page will be from that copy with any errors corrected by ECC. (18:45-52.) If the number is greater, then a copy is rewritten from the cached copy to the rewrite section of the more robust portion. (18:53-66.) This then shows that when the number of errors in a copy written in MLC (for example) exceeds a certain amount, then that data is moved (rewritten) to SLC.

49. U.S. Patent No. 8,806,301 to Yu, published April 14, 2011 (“Yu ‘301,” Ex. 1046) uses an error checking and correcting unit 210 to determine the number of error bits in the read data:

. . . when the memory management unit 204 receives a host write command from the host system 1000, the error checking and correcting unit 210 generates an error checking and correcting code (ECC code) corresponding to write dat[a] of the host write command, and the memory management unit 204 writes the write data and the ECC code into the flash memory chip 106. And, when the memory management unit 204 reads data from the flash memory chip 106, the memory management unit 204 simultaneously reads the ECC code corresponding the read data, and the error checking and correcting unit 210 performs the error checking and correcting process to the read data based on the ECC code corresponding to the read data. In the present

exemplary embodiment, the maximum number of error bits which can be corrected by the error checking and correcting unit 210 is 26. . . .

(6:15-29.) However, “in another exemplary embodiment of the present invention, the memory management unit 204 may compare the read data and the writing data from the host system 1000 bit by bit to determine whether the read data has any error bit and the number of error bits thereof.” (11:49-53 [emphasis added]).

50. A bit-by-bit comparison of user data to be written (held in some temporary memory, *e.g.*, via Gavens’ on-chip latches) with the data read from the memory where it is written requires longer processing time than typical ECC schemes using a controller for comparison (18:36-39). While there are the alternative bit-by-bit embodiments recited in Gavens and Yu – which necessarily (inherently) would require adequate memory available to the controller to hold a to-be-written page to compare with a written-and-read page – I am unaware of any commercial use of bit-by-bit “comparison” for read-after-write “integrity checking” for flash memory management. In my view, it would be an inefficient choice. Keeping in mind that a page of data is on the order of 8kB or 64k bits, it is a daunting task to compare two 64k-bit data items bit-by-bit. Typically, a subtractor is used to do the comparison, and this takes place using an adder, since subtracting two vectors ($A - B$) is equivalent to adding A to the complement of B . Now the largest adders handle about 64 bits (8 bytes). That means the adder would have to be applied 1000

times! Or you would need 1000 such adders to accomplish the task in the time it takes for one 64-bit adder to execute. A combination of the two approaches (serial and parallel) could also be attempted. It is also necessary to accumulate the total number of errors, so each 64-bit addition result must be accumulated (all 64-bit adder results must be added together) in a memory or register. Overall, this takes an enormous amount of resources and time, and a POSITA would not expect this method to be used in industry, even though it is theoretically a possible approach.

D. Strategies for NAND Flash Memory Remapping To Manage System Reliability and Efficiency

51. Prior to July 19, 2011, there are many related strategies for avoiding premature failure (or falling below rated endurance) of a memory component of a NAND flash memory system thereby degrading system reliability, including availability of adequate memory space; as reviewed in Gavens (note 8 above), selection of a strategy (such as application of ECC) may be informed by statistics developed from testing. To some extent, simple rotation of writing of new or updated data among physical blocks provides some wear leveling, even among components of the same kind. Micheloni at 494 (dynamic, simple round robin algorithm selects one block after another in a chain of available blocks, where erased blocks are added to the chain). As further reviewed at paragraph 28 above, wear leveling, aside from attempting to achieve uniform wear on NAND flash blocks, also

involves attempting to achieve (garbage collection) efficiency by clustering “hot data” and “cold data” for efficient writing and erasing.

52. U.S. Patent Appl. Pub. No. 2010/0115192 to Lee (“Lee ‘192,” Ex. 1047, Samsung assignee) disclosed wear leveling in a memory system with both SLC and MLC blocks (Fig. 2), such that if the sum of “SLC mode usage” plus “MLC mode usage” minus an average usage of all blocks is greater than a threshold set for the mix of applications for the SLC (faster and more frequent writes) and MLC blocks, wear leveling is performed (Fig. 4), in which the LBA is changed, defining a new logical/physical memory block relationship (§ [0023]).

53. As reviewed at paragraphs 41 and 42 above, Sutardja (Ex. 1042) and Moshayedi (Ex. 1043), respectively detailed multiple embodiments for preferential writing into SLC or MLC modules. As explained in a contemporaneous article, Minyoung Sung & Kanghee Kim, *Asymmetric Flash Volume Management*, 58 *IEEE Transactions on Consumer Electronics* 455 (May 2012) (Ex. 1037), the asymmetry between SLC and MLC is not just endurance, but in the organization of page and block capacity to be considered relative to user data. Thus, an aspect of preferential writing is “to maximize the overall read/write performance of the logical volume by (1) hot data identification and (2) periodic data migration between SLC and MLC [with the objective of providing] the highest overall read/write throughput for the next period.” *Id.* at 455. Thus, “write operations from a process with a high storage

update frequency are marked ‘hot’ before forwarded to the flash storage system.”
Id. at 456.

54. SLC/MLC flash memory remapping based on detected errors (“bad blocks”) was also established in the state of the art.

55. Yu ‘301 (Ex. 1046), *e.g.*, Fig. 5, disclosed a scheme of reading after writing, resulting, as quoted in paragraph 50 above, in a number of erroneous bits. If the number of erroneous bits exceeds a maximum that can be corrected in the ECC implementation for the system (26 in the example), the remaining “valid” data in the physical block is copied to another physical block, the original block is marked “bad,” and a system warning is sent to the host system. If the number is less than the maximum but more than a set threshold, the valid data is copied to another block, and the original block is marked “bad.” If the number is greater than zero, but less than the threshold, the block is marked “unreliable.”

56. Gavens (Ex. 1045) disclosed a similar scheme of “post-write read” error-management decisions based on an ECC determination of a threshold number of erroneous bits applied to multiple strategies of writing or temporarily writing to a memory configured into a “first portion having memory cells operating with a margin of error larger than that of a second portion” – such as between SLC and MLC operation (*e.g.*, Figs. 15, 17). Two groups of these embodiments were described in paragraphs 47 and 48 above. In another embodiment (Fig. 19), the

error-management scheme is not enabled until a “hot count” (number of erases) of a block reaches a threshold.

VI. Person of Ordinary Skill in the Art (POSITA)

57. I am informed by counsel that a “person of ordinary skill in the art at the time of the inventions” (“POSITA”) is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the inventions. This hypothetical person would also be a person of ordinary creativity, capable of understanding the scientific principles applicable to the relevant field.

58. In view of this instruction, my experience with the issues, and the ‘546 specification, in my opinion, a person of ordinary skill in the art at the time of the claimed inventions would have a bachelor’s degree in computer engineering, electrical engineering, computer science, or a closely related field, along with at least two years of experience in the design, development, implementation, or management of memory devices and systems. A person with an advanced degree in a relevant field, such as computer or electrical engineering, would require less experience in the development and use of memory devices and systems. As is common, one could obtain equivalent knowledge and perspective from other life experiences as well.

VII. The Common Specification of the Lifetime Patents

59. Having read many patents, the common specification of Lifetime Patents, which incorporate the Embedded RAM Patents (identical specifications) struck me as disclosing little that is even suggested to be new. The embodiments disclosed merely skeletally avail of recited, known superior endurance of SLC NAND flash memory over MLC NAND flash memory to improve expected useful lifetime of a system with understood, physical modules of each in appropriate number. The embodiments also use an otherwise unspecified “controller” that, using well-known NAND flash L2P/FTL mapping, (1) performs the flowchart of Figs. 3A and 3B to remap into (physical) SLC memory data for a logical address previously mapped and written to a physical memory address that fails a comparison of the data to be written and the data read after actual writing; and/or (2) even more skeletally, “allocates” “‘hot’ (NAND flash) blocks” into “the SLC NAND flash memory module.” My interpretation is that the basic function disclosed (against which claims may be measured) is the preferential mapping into physical SLC NAND flash modules to improve NAND flash system life (see note 9 *infra*).

A. Lifetime Patents Specification BACKGROUND

60. The ‘546 patent specification, as that of all the Lifetime Patents, admits a “BACKGROUND” state of the art that includes parts of the NAND flash background reviewed above, particularly that it “is well known in the art, any ‘write’

or ‘program’ to a block in NAND flash (floating gate) requires an ‘erase’ of a block before ‘write.’” (3:41-44.) As part of that NAND flash technology, logical block to physical block (L2P) mapping (implemented with an FTL) was described for “erase-before-write” NAND flash operation to emulate host-computer logical address operations associated with “write in place” (sectors) hard disk drives, mapping (and impliedly remapping in updates) to NAND flash blocks of 32 to 64 pages, each with 2-4 Kbits (2:21-60). The POSITA would understand that the lower numbers are associated with one-bit SLC modules and the higher ones with 2-bit MLC modules. Described in the BACKGROUND is the use of the FTL to manage wear leveling “to attempt to ensure that ‘hot’ blocks, *i.e.*, blocks that are frequently written, are not rendered unusable much faster than other blocks.” (3:4-10.) This is characterized as “wear leveling” (3:14-26), but it is not the POSITA-known NAND flash wear leveling. The BACKGROUND confuses its notion of “hot blocks” with “hot data,” the clustering of the latter and “cold data” (associated with logical blocks) explained at paragraph 28 for NAND process-efficient garbage collection and, in the Sung article (Ex. 1037) quoted in paragraph 53, for efficient throughput. The BACKGROUND characterizes wear leveling as “typically” writing to the (physical) block with the lowest block count and “ensures that blocks of static data are moved when their block erase count is below a certain threshold” (3:19-26 [emphasis added]) – both incorrectly calling this “static wear leveling” (compare ¶ 28 above)

and presenting a pointless floor as erases only increase. Without explanation, the BACKGROUND states that the prior art wear leveling, by determining the physical block to be written, “prolong[s] the service life of the flash memory.” (3:1-6.)⁹ The BACKGROUND states that “various embodiments of a NAND flash storage system that provides long life time (endurance) storage at low cost are described herein.”

⁹ The specification, without explanation of actual causation, recites a need for “writing data optimally for improved life time (endurance) of flash memory” (3:56-57), embodiments of NAND flash “that provides long lifetime (endurance) storage” (3:58-60), the use of SLC “to boost the lifetime (endurance) of the storage system (5:7-9), and of a system and method of “increasing the reliability and lifetime of a NAND flash storage system, module, or chip” (7:8-9). I understand from my experience and the literature, such as Gavens (note 8 above), and believe that a POSITA in computer memory would understand, that the improvement sought in the field and referenced in the Lifetime Patents specification is to avoid premature failure (actual or rated) of enough memory components to result in degradation of a hybrid SLC-MLC NAND flash system making it no longer reliable for its intended use. This “improvement” is relative to a general hybrid NAND system configured for projected uses, not “maximized” for a given actual system which may be used differently from such projection.

(3:43-45.) However, the only embodiment described in the DETAILED DESCRIPTION is a system of SLC memory modules, MLC memory modules (their relative expected endurances given in the BACKGROUND at 2:18-20 and 3:31-41) and an abstract controller without structure as a hardware or software module, explained only by a two-paragraph explanation of the NAND flash memory steps of Figs. 3A and 3B and “additional” embodiments of restated FTL and two sentences that further confuse the BACKGROUND’s incorrectly stated “wear leveling.”

B. Proffered Embodiments in the DETAILED DESCRIPTION

61. The only embodiment described in the Lifetime Patents specification “DETAILED DESCRIPTION” with any structure is for the method steps of Figs. 3A and 3B for the operation of a mixed SLC and MLC module system, with a theme of preferential NAND mapping and writing to an SLC module. Introduced as “other” embodiments were the known NAND-flash characteristic FTL and a two-sentence restatement of the incorrect BACKGROUND “wear leveling” reviewed in the preceding paragraph.

62. **SLC and MLC modules’ physical and operational foundation:** The proffered innovative system is stated in the “DETAILED DESCRIPTION” as the combination of POSITA-understood physical SLC and MLC NAND flash memory modules and an abstract controller:

Generally, and in particular regard to NAND flash memory, two separate banks of NAND flash are maintained by a controller. One bank contains economical MLC NAND flash, while a second bank contains high endurance SLC NAND flash. The controller conducts a data integrity test after every write. If a particular address range fails a data integrity test, the address range is remapped from MLC NAND flash to SLC NAND flash.

(4:67-5:4.) The only alternative embodiment is shown in Fig. 4 and described as

two NAND flash memory banks **56** and **58**, whereby memory bank **56** comprises a plurality of MLC NAND flash memory modules 60a and a plurality of SLC NAND flash memory modules 62a. Memory bank **58** comprises a plurality of MLC NAND flash memory modules 60b and a plurality of SLC NAND flash memory modules 62b.

(7:61-67 [emphasis added].)¹⁰ In language reflecting the first, ‘298 patent claim 1

(7:8-8:9), the “SUMMARY” states,

¹⁰ The term “module” is used throughout the common specification, including in the context of alternative physical items: “NAND flash storage system, module or chip” (‘546 1:39 [emphasis added], *also* Fig. 7 in Rao ‘452 (Ex. 1004) 1:40-41 [“ICs mounted on a memory card or module”]) and “a device or a module containing devices” (‘546 52:48 [emphasis added]).

there is provided a system for storing data which comprises at least one MLC nonvolatile memory module (hereinafter referred to as “MLC module”) and at least one SLC non-volatile memory module (hereinafter referred to as “SLC module”), each module comprises a plurality of individually erasable blocks. The data storage system according to one embodiment of the present disclosure further comprises a controller for controlling both the at least one MLC module and the at least one SLC module. In particular, the controller maintains an address map comprising a list of individual logical address ranges each of which maps to a similar range of physical addresses within either the at least one MLC module or the at least one SLC module. After each write to (flash) memory, the controller conducts a data integrity check to ensure that the data was written correctly. When the data was not written correctly, the controller modifies the table so that the range of addresses on which the write failed is remapped to the next available range of physical addresses within the at least one SLC module.

(4:8-26 [emphasis added].) A POSITA would understand “module” in this context to be a NAND flash memory chip or package such as labeled “NAND” or “FLASH” in the block diagrams and the picture in background paragraph 27 above and in Figs. 6a and 7 of the incorporated Embedded RAM patents¹¹ (*e.g.*, Rao ‘452, Ex. 1004).

¹¹ Figs. 6-8 show a “current” Samsung NAND flash chip K9F108XOA with a “1024M + 32M BIT NAND FLASH ARRAY,” clearly a “module” of 1-bit cells,

It comprises the NAND flash memory array and input/output (including addressing, writing and sensing) circuitry. This understanding of “module” is compelled by computer electronics usage¹² and the inclusion of multiple individually erasable blocks, understood by a POSITA as a defining structure of NAND flash memory.

i.e., SLCs. *Compare* Fig. 4 with Moshayedi (Ex. 1043) specifically showing four channels (banks) Fig. 1 (108) each connecting “five chips of flash (e.g., the first chip can be a K9K8G08 SLC flash, and the rest can be K9G8G08 MLC flash)” (¶ [0038]).

¹² The MICROSOFT COMPUTER DICTIONARY 346 (5th ed. 2002) (Ex. 1027) provides the following definition:

In hardware, a self-contained component that can provide a complete function to a system and can be interchanged with other modules that provide similar functions. *See also* memory card, SIMM.

Similarly, from the Merriam-Webster online dictionary, <https://www.merriam-webster.com/dictionary/module:>

3b: a usually packaged functional assembly of electronic components for use with other such assemblies

A “flash memory module,” as used for embedded applications, “can readily be integrated into the host logic device” and includes a bus interface unit distinguished from the flash memory core, which includes a memory control as well as a flash memory array. Brewer [Ex. 1026] at 395-96, Fig. 9.13.

There is no subdivision of memory cells that could be coupled to the controller without the input/output circuitry. A POSITA would also understand not only from their separate recitation, but from essentially different capacity, organization and incremental writing in passes (*see* ¶¶ 30-39 above), that an SLC module is different from an MLC module. Operation of part of an MLC module in 2-state (although sometimes loosely called “SLC”) mode (¶ 43 above) does not transform it into an SLC module both because of the understood separate usage and the substantial operational, performance and endurance differences between “pseudo-SLC” and real SLC (*id.*). In the SUMMARY language, “at least one” each of SLC and MLC modules – if only one – appears too limited to achieve the design of the Lifetime Patents to avail of the superior endurance of SLC: the only suggestion of a beneficial mix is “SLC NAND flash equal to 12.5% or 25% of MLC NAND flash” (5:10-11), which is consistent with Fig. 4 showing two banks each of 2+ SLC modules compared to 8+ MLC modules (which with MLCs capable of storing twice the amount of data as SLCs, fitting the suggested 12.5% ratio).¹³

¹³ *See also* Moshayedi ratio at note 11 above. This also corroborates the POSITA’s understanding that SLC modules comprise all SLCs as distinguished from MLC modules, rather than any partition for 2-state operation of MLCs: the specification for the first, ‘298 patent was filed on April 25, 2012, nearly ten years after issuance

63. **The controller and defining, SLC remapping function:** Tying the subject SLC and MLC foundational modules together in the NAND flash memory system is a “controller.” The term by itself is abstract, merely suggesting some

of Chen ‘528 (Ex. 1040), which disclosed the 2-state operation of MLC. However, the Lifetime Patents specification mentioned nothing about such operation, much less a partition of MLC modules for such operation. The specification Summary statement (4:26-27) that “SLC module can be” PCM, RRAM, or MAGRAM offends a POSITA’s understanding of “SLC module.” These alternative technologies are explained in the specification BACKGROUND as having fundamentally different operation from erase-before-write NAND flash operation (2:50-60; ¶ 27 above) by “1) allow[ing] data to be written over existing data (without prior erase of existing data) [and] 2) allow for an erase of individual bytes or pages (instead of having to erase an entire block)” (2:15-18). These other technologies do not have page and block operations as does a NAND flash. A POSITA immediately understands the term “SLC memory” refer to NAND flash and the term “SLC module” to refer to collection of many NAND flash blocks of SLCs with supporting circuitry. CPM, RRAM, and MAGRAM would not be considered as having “SLC” or “single level cells” because that description of two-state NAND operation is used only to distinguish it from multi-level NAND flash cells.

“control” of the physical components, namely SLC and MLC NAND flash modules. The Lifetime Patents specification might have recited a POSITA-understood NAND flash memory controller with the functions outlined at paragraph 27, but instead maintained abstraction, stating in the SUMMARY reproduced above that in “one embodiment” there is a controller for “controlling” the SLC and MLC modules (4:14-16), particularly “maintain[ing]” a logical to physical address map (4:16-20). The function further defining this embodiment is “conduct[ing] a data integrity check” after “each write” “to ensure that data was written correctly,” and if not, “modif[ying]” the L2P address map to remap into the SLC module (4:20-26).

64. There is a paraphrase of this “read-modify-write” function at 5:49-57 (“as time permits”), but the only structure for the function is set forth in method steps of Figs. 3A and 3B and the description at 6:5-44.¹⁴ According to that description, in Fig. 3A (100) a command is received (presumably from the host) to write a “quantum

¹⁴ Figs. 3A and 3B of the five patents are identical to the hand-drawn Figs. 3A and 3B in the Provisional (Ex. 1003, sheets 2 and 4). The descriptions are virtually identical except for a few transition words. Compare ‘546 5:58-6:43 with Ex. 1003 at 7-8, ¶¶ [0022] and [0023].

of data stored in DRAM”¹⁵ to a particular location in NAND flash (not limited to the MLC module or the SLC module). In step 102, that data is “read from DRAM into memory within the device controller”¹⁶ (6:10-12 [emphasis added]). In step 106 that

¹⁵ A POSITA would understand that (1) the abstract “quantum” refers to data from the host directed to a logical NAND block address (with sub-addresses) for that data (which may have been previously established in update cases), and (2) a NAND flash device controller would assign (or have assigned) a physical NAND block address and issue appropriate memory commands to be executed at the SLC NAND flash and/or MLC NAND flash module level according to their distinct physical structures and operations. None of this is disclosed for alternative technologies. The specification also does not disclose how the quantum of data gets to DRAM in the disclosed system, which might be assumed to be DRAM 20. However, in my view, the referenced DRAM should be that of (or directly connected to) processor/host 12, rather than DRAM 20 shown as directly connected to device controller 14 in Fig. 1.

¹⁶ The “controller” of the DETAILED DESCRIPTION is abstractly connected in Fig. 1 (described at 5:13-32) as device controller 14 by lines (“coupl[ing]”) to processor 12 (host computer), DRAM 20, MLC flash 26, SLC flash 28 (such as “NAND flash memory modules, confirming their physical distinction) and other discrete devices such as disks 24. Other than the quoted description of “memory,”

data (in controller memory) is “combined with the contents of the NAND flash memory”¹⁷ (6:16-18). In step 108, that “NAND flash physical address to be written is erased.”¹⁸ (6:18-19). Then, in step 110, “the combined data is written to the

there is no description of the structure of a controller having internal memory into which contents of “DRAM” may be “read” (not how I would describe “writing”). A POSITA would understand a NAND device controller to be a separate chip (see photograph at ¶ 27 above, Rao ‘452 [Ex. 1014] Fig. 3) in which chip a DRAM device could not be embedded.

¹⁷ It is indefinite what “content” is referenced. Step 104 suggests that it would be the physical address to which “the NAND flash physical address to which the quantum of data is to be written,” also “read” to unspecified controller memory (6:13-15). But that would inexplicably mean the existing data at that address would be combined with the data to be written to that address. It may be that this was intended to address the situation in which the targeted address may include valid data (in part of a physical page) to be combined with the new data (to be updated to the corresponding logical page) to be written as to another physical address. But that is not what the specification says.

¹⁸ This is highly inefficient. To erase the address targeted, the entire block must be erased, and other valid data rewritten to other physical blocks – disrupting, adding

appropriate NAND flash physical address range.” (6:19-21.) In step 112, the combined data that was written in step 110 into NAND “is read into device controller memory”¹⁹ (6:21-22 [emphasis added]) – which is not otherwise identified in the specification. Then, in step 114 the “the NAND flash physical range that was read into device controller memory is compared to the retained data representing the combination of the previous contents of the physical address range and the quantum of data to be written”²⁰ (6:24-29). If at branch 116 there is no “match” in the step

latency to memory system operations. Normally the erasure would be part of an optimized garbage collection process that could be performed in the background.

¹⁹ It is indefinite whether “device controller memory” is the previously referenced “memory within the device controller,” which is not explained as part of the device controller physical structure (note 16), or DRAM 20 which is shown as coupled to the device controller, which thus might be considered “device controller memory.”

²⁰ This step is incorrectly described. “[T]he NAND flash physical range that was read into device controller memory” was part of step 104 (6:13-15) which apparently established an L2P mapping with the “logical address range” also “read” into the device controller memory in that step. It makes no sense to compare that physical address range to “retained data” – which itself has no specified location in “representing” the combination apparently of step 106 (note 17 above).

114 comparison,²¹ and if there is available SLC memory, “the failed NAND flash physical address range is remapped²² to the next available quantum of SLC NAND flash memory” and the write process from step 110 repeated. (6:29-43.) Nothing is said about flagging or retiring the physical address range in which data was written and found not to match the data that was mapped to be written there: if that range is not retired, there may be problems in the future resulting in wasteful checking, remapping and rewriting, adding to wear. On the other hand, if the range is retired early in device lifetime, this is also wasteful, taking memory out of service upon a

²¹ This “match” language makes clear that the Lifetime Patents’ read-after-write integrity check is disclosed as a bit-by-bit comparison between data. This was corroborated in the ‘300 patent prosecution express surrender of coverage of ECC error detection (Oct. 8, 2020, Amendment at 8-9 [Ex. 1012 at 36-37]), which was repeated in ‘546 prosecution argument (Sept. 22, 2023, Amendment at 13) [Ex. 1014 at 13] [requiring comparison in “two data stores”]). As I explained at paragraph 50 above, bit-by-bit comparison is inefficient. It is particularly inefficient if made after each write, severely contributing to latency.

²² It doesn’t make sense that the physical address range is remapped to another physical location, since mapping is between logical addresses and physical addresses.

single error. That would lead to earlier exhaustion of system capacity (and thus life) compared to prior art ECC schemes, statistically deployed later in device life, which still allow use (corrected reading) of data found incorrect until error rates exceed chosen thresholds of correctability leading to bad block identification and retirement (§§ 47-49 above). Contradicting the SUMMARY that the disclosed data integrity check is performed after every write to flash memory (4:20-21), the DETAILED DESCRIPTION qualifies that function: “when time permits” (5:51-54). These two substantial ambiguities, in the only specification embodiment with any structure, appear to me to fail the disclosure requirements of enabling written description showing “possession of the claimed invention.” These failures were amplified in the indefiniteness of the increasingly abstract claims of the Lifetime Patents as reviewed in section VIII below.

65. **FTL feature as additional “embodiment”:** Fig. 4 is described as “another embodiment” where “FTL” logic is “entirely resident” in a NAND flash module that includes two banks of NAND flash memory banks of SLC and MLC NAND flash modules. (6:56-67.) There is nothing new about FTL. I explained at paragraph 28 above the prior art FTL’s role in NAND flash-required L2P mapping, wear leveling and garbage collection. FTL is admitted prior art in the specification background, mixed with reference to L2P mapping implemented other than in a software layer (2:61-3:26). Nothing is added by the alternative FTL embodiment

other that it exists in some “logic” in a physical module that includes MLC and SLC modules. Such a module typically is a NAND flash memory card such as depicted at paragraph 27 (Micheloni Fig. 2.33) to include a flash file system “implemented as firmware operating in different functional sublayers for wear leveling, garbage collection and bad block management” such as a Flash Translation Layer.

66. **Wear leveling features as additional “embodiments”:** Another “embodiment” or two are proposed in a two-sentence DETAILED DESCRIPTION restatement of the BACKGROUND “wear leveling” (3:14-26), shown at paragraph 60 to be an incorrect sketch of POSITA-understood NAND flash static wear leveling relative to “hot” and “cold” data. The first sentence states:

Another application of one embodiment of the present disclosure, not depicted in any of the drawings, is to allocate “hot” blocks;²³ i.e., those blocks that receive frequent writes, into the SLC NAND flash memory module **28**, while allocating “cold” blocks; i.e., those blocks that only

²³ As reviewed at paragraph 28 above, the Micheloni textbook concept of “hot data” relates to clustering for garbage collection efficiency often-updated data associated with logical blocks. At paragraph 41, Lee ‘794 (Ex. 1041) is shown to write frequently written data such as system data and user directories (with associated logical blocks) to physical SLC blocks and less frequently written data such as user data to physical MLC blocks. *Also* Moshayedi as explained at para. 43.

receive infrequent writes, into the MLC NAND flash memory module **26**.

(6:44-50.) It is not explained in the DETAILED DESCRIPTION whether the referenced blocks are the logical blocks to which a host directs data frequently or infrequently (“hot” or “cold” data as explained in paragraph 28 above) or physical blocks that may be worn by frequent or infrequent writing of different data (since updating data generally means writing to a different physical block). Nor is it explained what allocation “into” SLC or MLC modules means and when and how it is performed. The next sentence appears to be a sub-embodiment:

This could be accomplished within the device controller **14** described above, which could simply maintain a count of those blocks that are accessed (written to) most frequently, and, on a periodic basis, such as, for example, every 1000 writes, or every 10,000 writes, transfer the contents of those blocks into the SLC NAND flash memory module **28**.

(6:50-56.) Contrary to the BACKGROUND’s incorrect description of “static wear leveling” as “static data” moved when the erase count of its physical block falls below a threshold (para. 60 above), the DETAILED DESCRIPTION sentence maintains ambiguous reference to logical or physical blocks and adds further indefiniteness as to whether “maintain[ing] a count of those blocks” means counting blocks that are written most frequently or maintaining a count for each block. The first clause is consistent with selective writing of “hot data” (frequently written

logical blocks) to SLC as in Lee '794 and Moshayedi (note 23 above). In further indefiniteness, the second clause, of “periodic” transfer (presumptively a time periodicity), is inconsistent with its recitation of thresholds of numbers of “writes” and remains indeterminate as to its reference to logical or physical blocks whose “contents” are transferred to the SLC module. If a physical block, then infrequently updated data can push the physical block over the threshold write count, the embodiment would wastefully map infrequently written data to more expensive (and less numerous – such as the 12.5% recommendation, 5:10-12) SLC memory. The “wear leveling” embodiments of the DETAILED DESCRIPTION make no sense, being inoperative to enhance system lifetime and non-enabling of rationalization in the claims of the Lifetime Patents, as reviewed in Sections VIII and IX.

67. **Non-Enablement of Other than SLC and MLC Technologies:** A POSITA would understand that the Lifetime Patents specification at most discloses the NAND flash operation of a mixed system of physical MLC and SLC modules favoring mapping and remapping to SLC modules under two circumstances: failure of a read-after-write comparison of data to be written with data as written and some kind of wear leveling, for the unexplained improvement of system lifetime (endurance). It is stated in the SUMMARY that the “SLC module” can be something other than a NAND module, such as “PCM, RRAM, MAGRAM or any other solid-state non-volatile memory as long as it has endurance that is superior to that of MLC

flash” (4:26-29). But there is nothing in the specification to explain how the embodiments reviewed at paragraphs 62-67 would operate (and enable the Lifetime Patents claims) by simple substitution for an SLC module, which is understood by a POSITA to incorporate physical input/output (charging, sensing, erasing) circuitry for SLC NAND flash operation, such as shown at paragraph 27 and in the Embedded RAM Patents. Unlike the fundamental NAND flash operation of page-write-after-block-erase, “PCM is a ‘direct write’ technology. There is no need to erase a block of data and then reprogram it.” (Brewer [Ex. 1026] at 707-08, 716.) Indeed, in the only explanation of the alternative technologies, the BACKGROUND focuses on their operational differences from NAND flash (2:15-18; note 13 above). The ambiguities, inaccuracies and inconsistencies in the DETAILED DESCRIPTION confirm to me that none of specification embodiments (save FTL) were implemented in the real world but merely speculative, not showing possession of any invention.

VIII. The Evolving Abstraction of the Lifetime Patents Claims

68. This Section reviews the evolving abstraction in the Lifetime Patents from a claimed memory system of physical, understood MLC and SLC modules “coupled to” a “controller” limited by functions recited in the claims and supported at most by the “embodiments” reviewed at paragraphs 62-67. with preferential mapping to the SLC module to improve useful NAND flash system lifetime. The evolution was to claims unanchored in physical or even algorithmic structure to

attempt to cover the technologies referred to in paragraph 67 without any enablement of those technologies in the specification. That evolution involved the abstraction (1) of physical NAND flash modules and blocks to “memory spaces” and “memory elements” which have no support in the specification and (2) of “maximiz[ing] lifetime” not limited by the DETAILED DESCRIPTION preferences for mapping into SLC modules. My description of this evolution should not be considered an exhaustive critique of the claims of the Lifetime Patents preceding the ‘546 patent (claims which I critique in detail in Sections IX and X).

A. Pre-2018 Physical MLC and SLC Modules and Coupled Controller Defined by Functions.

69. The first, ‘298 patent was issued on an independent claim that corresponds to the “embodiments” reviewed in Section VII.B:

A system for storing data comprising:

at least one MLC non-volatile memory module comprising a plurality of individually erasable blocks;

at least one SLC non-volatile memory module comprising a plurality of individually erasable blocks; and

a controller coupled to the at least one MLC non-volatile memory module and the at least one SLC non-volatile memory module wherein the controller is adapted to:

a) maintain an address map of at least one of the MLC and SLC non-volatile memory modules, the address map comprising a

list of logical address ranges accessible by a computer system, the list of logical address ranges having a minimum quanta of addresses, wherein each entry in the list of logical address ranges maps to a similar range of physical addresses within either the at least one SLC non-volatile memory module or within the at least one MLC non-volatile memory module;

- b) determine if a range of addresses listed by an entry and mapped to a similar range of physical addresses within the at least one MLC non-volatile memory module, fails a data integrity test, and, in the event of such a failure, the controller remaps the entry to the next available equivalent range of physical addresses within the at least one SLC non-volatile memory module;
- c) determine which of the blocks of the plurality of the blocks in the MLC and SLC non-volatile memory modules are accessed most frequently by maintaining a count of the number of times each one of the blocks is accessed; and
- d) allocate those blocks that receive the most frequent writes by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module.

(‘298 7:6-8:9). The first two elements of physical SLC and MLC modules are physically coupled to a third element of a controller defined by four functions that roughly correspond to the DETAILED DESCRIPTION disclosures reviewed at paragraphs 63-66 above. Controller function a) roughly corresponds to the FTL/L2P

functions admitted in the BACKGROUND (for convenience, ‘546 2:61-3:26). Controller function b) roughly corresponds to the DETAILED DESCRIPTION Figs. 3A and 3B SLC remapping upon failure of the comparison test reviewed at paragraph 64, but with the test applied only to initial writes to MLC. Controller functions c) and d) restate the two sentences of the DETAILED DESCRIPTION shown at paragraph 66 to describe an even more indefinite and inoperative “wear leveling” than that incorrectly described for wear leveling as shown at paragraph 60. This core physical memory system of SLC and MLC memory modules (understood to include read/write/erase circuitry) coupled with a controller defined by the recited functions is recited virtually identically in the only independent claim 1 of the next patent (‘385 7:11-8:4, explicitly adding FTL) and, with some elaboration, independent claims 1 and 6 of the third patent (‘240 7:30-8:7 and 8:20-59). Thus, the ‘298, ‘385 and ‘240 patent claims conformed to the Lifetime Patents specification disclosure of NAND flash operation of a mixed system of physical MLC and SLC modules favoring mapping and remapping to SLC modules after failure of a read-after-write test and upon some schedule of wear-leveling. Although the claims explicitly recited mapping into SLC modules, they were only by specification background – and not by express claim limitation – directed to improving mixed NAND system lifetime based on the well-known superior endurance of SLCs over MLCs. In further loosening of claims, the third continuation

application, nearly seven years after the July 19, 2011, priority date, abstracted its claims to the point where they no longer recited a preference for mapping or remapping into the SLC module.

B. 2018 Abstraction of the Preferential Mapping to an SLC Module.

70. **Abstraction of the base terms:** The opening independent claim of the application resulting in the fourth patent, the ‘300 patent, reads:

A system for storing data comprising:

memory space containing volatile memory space and nonvolatile memory space, wherein the nonvolatile memory space includes both multilevel cell (MLC) space and single level cell (SLC) space;

at least one controller to operate memory elements and associated memory space;

at least one MLC nonvolatile memory element;

at least one SLC nonvolatile memory element;

at least one random access volatile memory element;

an FTL flash translation layer, wherein the at least one controller, or FTL, or a combination of both maintain an address table in one or more of the memory elements; and

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.

(June 12, 2018, Application at 14 [Ex. 1012 at 19] [emphasis added]).

71. **Abstracting out physical MLC and SLC modules:** Despite the base structure of the Lifetime patents specification of physically defined MLC and SLC modules, the ‘300 patent application substituted a recitation of abstract “memory space” and “memory element,” apparently to allow claim to other types of technology and distribution of memory. That erasing of the physical boundaries established in the specification had no basis. The word “space” had been used in the specification only in the context of the market space for hard disk drives, “HDD space” (‘546 2:40), and a quantity of “total non-volatile memory space” (5:12; *also* incorporated ‘912 4:50-51 and ‘452 4:48-49 [“available Storage space”]). In this context, a POSITA would understand “memory space” to be the storage capacity property of physical memory. “Element” is only mentioned in the incorporated patents describing Fig. 2 as an “exemplary NAND memory integrated circuit as one element of the NVMS” (‘Rao ‘452 [Ex. 1004] 2:52-53). Given the broad

understanding of word “element,”²⁴ a POSITA would not understand whether it refers to a cell (perhaps most elemental), a writable word or page or an erasable block (as in the original recitation) or even “bytes” recited in the mapping limitation – and whether it is different for the recited MLC, SLC and random access volatile memory²⁵ (“RAVM”). The abstraction facially allowed a reach to other-than NAND flash non-volatile memory technology, such as PCM, RRAM, MAGRAM or other relatively robust technology (‘546 4:26-31). But the abstraction, while facially allowing claim to such technologies, introduced uncertainty across other axes of choice: location, continuousness or discreteness in abstract “space,” physical or logical role, and different physical operations of the different technologies. No operations were identified, much less enabled, for other than NAND flash writing,

²⁴ *Microsoft Computer Dictionary* [Ex. 1027] at 189: “**1.** Any stand-alone item within a broader context.” Merriam-Webster online: “**f:** a distinct part of a composite device.” <https://merriam-webster.com/dictionary/element>.

²⁵ That addition of “random-access volatile memory” might appear supported by “DRAM 20” shown in Fig. 1, but as explained in note 19 above, the only example in the specification is that new data is first held in the “external” DRAM 20, then combined in “memory in the device controller” (existence and nature of which is not disclosed).

itself only barely reviewed in the specification. Unlike NAND flash “modules,” which a POSITA would understand to include known read/write/erase circuitry, technology-agnostic “element” includes no structure to define, much less enable, the claimed “system for storing data.” On an amendment, more new matter was added: the MLC and SLC elements were qualified respectively as those “that can be mapped into the [MLC/SLC] memory space”; the new RAVM element was not further qualified (Oct. 8, 2020, Amendment at 2 [Ex. 1012 at 30] [emphasis added]). The newly recited mapping didn’t even make sense for NAND flash, where the logical (host-visible) data address is mapped into a physical address, the latter updated with each updating of data; it is an irrelevant limitation that a physical “element” can be mapped into abstract “memory space.” More basically, technology-agnostic “element” still carries with it no understood physical structure including read/write/erase operations or effects for which “mapping” or “lifetime” has any relevance.

72. **Abstracting out the endurance advantage of SLC:** The ‘300 patent application not only abstracted the disclosed physical MLC and SLC modules and blocks to “spaces” and “elements,” it abstracted out the disclosed and claimed ‘298 mapping or remapping to the SLC modules, the alleged innovation of the Lifetime Patents. As a result, the filed claim 1, aside from reciting the inclusion of the abstract “memory spaces,” “controller” and SLC, MLC and RAVM “elements,” merely

claimed the abstract notion of “mapping logical to physical addresses of a system of abstract volatile and nonvolatile memory spaces and elements to maximize lifetime.” As quoted in paragraph 70, the only limitation on mapping among the recited memory spaces was that mapping is to be “performed as necessitated by the system to maximize lifetime” on “blocks, pages, or bytes of data in either volatile or nonvolatile or both, memories.” There is no mention or availing of any endurance advantage of SLC memory. There is no recitation, much less supporting disclosure of when and how “the system” “necessitates” “maximiz[ation of] lifetime” and of what. There is no indication whether what is modified by this recitation is (1) each mapping, where ordinary mapping (and writing) to MLC modules that itself would incrementally lessen lifetime, at least of the physical cell or block written, multiplied in the erase-before-write NAND flash premise (unlike the alternative “direct write” technologies, para. 67 above); or (2) mapping applied after some event and continuously (or not) thereafter. In stark contrast are the known phased strategies to avoid premature reliability failure of NAND flash system components based on statistics (note 8 above). In my view, it is impossible to maximize the lifetime of a particular complex memory system with random stresses without knowing *a priori* its potential lifetime. As reviewed at note 9 above, the lifetime (endurance) improvement of the Lifetime Patents specification, postulated, without further explanation, to particularly favor superior-endurance SLC modules, would be

understood by a POSITA to be some sort of attempt to avoid early unacceptable lowering of expected reliability of a hybrid SLC-MLC NAND system configured for a projected use. That does not define “maximization” of some “lifetime” as “necessitated” by a particular, recited hybrid NAND system. Not only may a claimed NAND flash system depart from its generally projected use of SLC and MLC modules relative to the Lifetime Patents’ disclosure of circumstances for preferential mapping to SLC modules, but there are other “lifetime” factors such as retention (of charge) which are subject to mitigation by refreshing by different processes (*e.g.*, Gavens 15:39-41). The ‘300 filed claim was directed to an abstract maximization of lifetime of abstract memory spaces and elements, with likely different “lifetime” factors for technologies other than NAND flash. An amendment to the maximizing lifetime mapping limitation added another limitation that abstracted steps 114 to 126 of Fig. 3B (recited in NAND flash physical terms as ‘298 controller function b) for failure of a post-write check – with remapping of “address space” only to a “different range of physical addresses from those determined to have failed the data integrity test to achieve endurance” (June 26, 2019 Amendment at 2 [Ex. 1012 at 23] [emphasis added]) – not to the SLC module that provided physicality and a superior endurance rationale in the specification and the ‘298

patent.²⁶ The amendment introduced the ambiguity of whether the maximizing lifetime mapping limitation applied to all mappings differently from the new remapping for failure of a post-write check, which itself was limited by the vague qualifier, “to achieve endurance,” modified in issued ‘300 patent claims 1 and 12, “to achieve enhanced endurance” (‘300 8:5-11, 9:3-9).

73. **Inconsistent restoration in ‘546 of “memory module”:** In the ‘546 patent, the terms “MLC memory module” and “SLC memory module” returned as part of one “bank of nonvolatile memory” (7:37-41).²⁷ This departed from the

²⁶ A new claim 10 limited the remapping of data “to the other of the MLC and SLC memory elements” (June 26, 2019, Amendment at 3 [Ex. 1012 at 24]), contrary to the alleged innovation, that a failed SLC address would be remapped to a less robust MLC location. After further amendments, the claim was issued as claim 7 with remapping “to the SLC nonvolatile memory element (‘300 patent 8:26-28). A new independent claim 19 recited the maximizing lifetime mapping limitation and SLC-agnostic “different address range” remapping plus transfer. (June 26, 2019, Amendment at 4-5 [Ex. 1012 at 25-26].) These limitations remain in the issued claim 12 (‘300 patent 8:39-9:9).

²⁷ The applicant stated in prosecution of the ‘546 patent, “The claim requires that there be a plurality of separate MLC memory modules and at least one SLC memory

DETAILED DESCRIPTION primary embodiment of two separate banks, one with MLC modules and one with SLC modules (4:64-5:4 [emphasis added]) and also from an alternative embodiment (Fig. 4, 6:57-67) showing two banks, each including 8+ MLC modules and 2+ SLC modules.²⁸ The latter depiction is consistent with the Lifetime Patents specification's only quantification of a memory system in which the alleged innovation would be beneficial: the recommendation that SLC memory would be 12.5-25% of MLC capacity (4:61-63) of depicted (Fig. 4) multiple NAND flash modules each with 1K NAND flash blocks (Embedded RAM Patents Fig. 6). However, as reviewed below for element [1c], '546 adopted for its reintroduction of "module" one that is not physically anchored to the POSITA-understood "plurality of [NAND flash] blocks" but one abstractly qualified as including "at least one element" of unspecified technology. This does not make sense for understood,

module in the bank of nonvolatile memory. This relates to Figure 4 of current application." (Sept. 22, 2023, Amendment at 9 [Ex. 1014 at 9] [emphasis added].)

²⁸ It would be understood by a POSITA that the memory module depicted in Fig. 6a of the incorporated Embedded RAM Patents (*e.g.*, Ex. 1004) was a Samsung NAND flash chip of 1,024 blocks each of 64 pages (each of 2K bytes plus 64 bytes for spares or system) or 1056 Mbits. (See note 11 *supra* [also showing Moshayed's use of five MLC modules for each SLC module].)

practical NAND flash operation. Nowhere, except in the claim itself, is it suggested that one “element,” even one NAND flash block of 64 pages, much less one page or some undisclosed alternative technology, could enable the claimed system for any usefulness. ‘546 retained the ‘300 patent abstracted “space–element” framework (element [1c]) and “maximizing lifetime mapping” (element [1h]), with a nod to the ‘298 physicality of the POSITA-understood NAND flash modules. The ‘546 backbone of ‘300 abstraction with a flash of understood ‘298 structure exacerbates the failings of both. As shown in Section IX, each of the nine labeled elements of the ‘546 patent claim 1, which at the claim level remains abstract with a controller defined by functions for which there is no structure provided in the specification, are defective as abstract, inoperative, unsupported (unenabled) and/or indefinite, invalidating (particularly the abstract maximizing lifetime mapping limitation) individually or in aggregate the ‘546 patent. Crucially, none of the ‘546 patent claims are limited to the mapping or remapping to SLC flash memory for wear leveling or failure of integrity, the only alleged innovation disclosed (if defectively) in the Lifetime Patents specification.

IX. The '546 Claims and their Textual and Conceptual Defects:

74. I understand that there is one independent claim 1 of the '546 patent from which the dependent claims 2-7 depend and incorporate the claim 1 limitations. Claim 1 is reproduced here with my labeling in brackets elements and sub-elements.

1. [Pre] A system for storing data comprising:
 - [1a] [1a1] memory space containing volatile memory space and [1a2] nonvolatile memory space, wherein the nonvolatile memory space includes both multi-level cell (MLC) space and single level cell (SLC) space;
 - [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 operations from the memory elements;
 - [1c] a bank of nonvolatile memory, including:
 - [1c1] a plurality of MLC memory modules, each including at least one MLC nonvolatile memory element, and
 - [1c2] at least one SLC memory module including at least one SLC nonvolatile memory element;
 - [1d] at least one random access volatile memory element;
 - [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;
 - [1f] the controller [1f1] having associated controller memory for storing received data therein, the controller [1f2] 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, [1f3] 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, [1f4] 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;

[1g] the controller [1g1] 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 [1g2] reading the stored data to the controller memory and [1g3] 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;

[1h] wherein [1h1] the address table maps logical and physical addresses adaptable to the system, wherein [1h2] the mapping is performed as necessitated by the system to maximize lifetime, and wherein [1h3] the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and

[1i] wherein a failure of the data integrity test performed on the stored data by the controller results in [1i1] a remapping of address space to a different physical range of addresses and [1i2]

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.

(7:28-8:34.)

75. I understand that terms of a patent claim are to be construed in accordance with the ordinary and customary meaning of such terms as understood by a POSITA at the time of the invention, considering the context of the claim, the specification, and the prosecution history pertaining to the patent. I also understand that statements made by the applicant during prosecution of a parent application are relevant to construing terms in a patent resulting from a continuation application if such statements relate to the subject matter of the claims being construed.

76. I review here the defects of the ‘546 patent claims from the viewpoint of a POSITA in electronic computer memory devices (Section VI) regarding the ‘546 base claim of a “system for storing data.” Such a POSITA would understand the claim text itself to fail to define a non-abstract, useful system (see paragraph 74 above) and that resort needs to be made to at least the “system for storing data” of the specification, a non-volatile memory system as explained only as NAND flash, as reviewed at Section VII. I critique here the compliance with the written description and claiming requirements (as I understand at paragraph 19 above) element by element as I have labeled them at paragraph 74 above. But I note that in this case, because of the abstraction of the claims I reviewed in Section VIII, I will first present

my views that (1) claim 1 is undeserving of patenting because it remains a claim to an abstract goal to “maximize lifetime” of an abstract memory system of abstract “memory spaces” of abstract “memory elements,” without any limitation to the advantages of the physical SLC module that is the disclosed rationale for the Lifetime Patents; and (2) the “controller” of claim 1 remains abstract, with stated and assumed functionality that are not provided structure (physical or algorithmic) in the specification, especially for the technologies implicated by the abstract “memory spaces” and abstract “memory elements.”

77. **Claim 1 claims an abstract goal to “maximize lifetime.”** As explained at paragraphs 72 and 73 above, the recitation of “mapping” in an abstract system of abstract “memory spaces” and “memory elements” (as in element [1a]) to “maximize lifetime” of something (as in element [1h]) is merely stating an abstract goal of “mapping to maximize lifetime.” Such mapping, not tethered to actual technology, is conceptual and conceptually could be performed mentally with pen and paper. I don’t believe “maximiz[ing] lifetime” can be defined or achieved even for an actual physical system of SLC and MLC modules (see note 9 above), much less an abstract system. And it is different from NAND flash wear leveling, defectively described in the specification (see ¶ 66 above), but as explained at paragraphs 28 and 53 above, is directed to uniformly spread the writes/erases of NAND flash blocks, reduce inefficient garbage collection by clustering “hot” and

“cold” logical data, and effectuate desired use of asymmetric capacity. Simply reciting the presence of an abstract controller [1b] and FTL [1e], [1c] SLC and MLC modules, and an RAVM element [1d], without some manifest connection to the abstract goal doesn’t transform the abstraction of the claim. The addition of controller functions for [1f], [1g] and [1i] together [as shown below, defectively] to remap to “a different range of physical addresses” as part of as their [1i] “result,” is not connected to the maximization goal [1h] or the preferential mapping to an SLC module disclosed as the only embodiments in the specification. The claim as a whole is inoperative to achieve the recited abstract maximization of lifetime or the specification goal of improving lifetime/endurance by use of SLC.²⁹

78. **Claim 1 recites a controller defined by functions for which no structure (physical or algorithmic) is disclosed in the specification.** A key system claim component (other than the [1c] SLC and MLC “modules” and [1d] RAVM “element”), the “controller”, is explained in the specification only by its role

²⁹ See note 9 above (SLC to “boost lifetime (endurance)” 5:8). Notably, element [1i] omitted from the otherwise nearly verbatim remapping and transfer clause of ‘300 patent claims 1 and 12, with its limitation-by-result, “to achieve enhanced endurance” (‘300 8:5-11, 9:3-9). Interestingly also, the “Improved” of the title of the Provisional [Ex. 1003] was dropped from that of the Lifetime Patents.

in Fig. 1, described as “controlling” various peripheral memory devices (note 16 above). No structure is provided except the functional block diagram in Fig. 3 of the incorporated Embedded RAM patents [e.g., Ex. 1004]. I understand that general terms such as “controller” which are recited in a patent claim to perform a function require corresponding structure (physical or algorithmic) disclosed in the specification. As just reviewed, there is no such disclosure for the “controller” of claim 1 at all, much less for recited or implied functions:³⁰ [1b] “to operate memory elements and associated memory space for associated Write access operations . . . and Read access operations” (7:33-36 [emphasis added]); [1e] maintain (with FTL) an address table (7:43-46); [1f2] “controlling access” of “memory elements” for “storage of the received data as stored data” (7:49-8:1) (without antecedent basis, ¶

³⁰ I am advised that hybrid product and process claims such as the claimed “system for storing data” limited by process steps such as “controlling,” “mapping,” “performing,” etc., are suspect. I understand that these steps are particularly suspect if they are not descriptive of the structure of the product but are steps that a user takes such that a manufacturer would not be put on notice of user infringement. As explained at note 9 above, user choice in configuration of the claimed memory system affects what improves “lifetime” in, for example, “mapping,” which itself is a user choice in what data is mapped and written.

74 above, and ambiguous as to which element stores the received data); [1f3] “transferring the store received data” (8:1-5) (without antecedent basis, ¶ 74 above); [1f4] “operable to store the received data” in an MLC “memory element” “as stored data” and “retain such received data” in the “random access volatile memory [not element]” as “retained data associated with stored data” (8:5-9) (indefinite, ¶ 84 below); [1g1] “performing a data integrity test . . . by reading . . . and comparing” (8:10-16 [emphasis added]) and [1i] “failure of the data integrity test performed on the stored data by the controller results in a [1i1] remapping” and [1i2] transfer of data (8:28-34 [emphasis added]). In addition to the lack of clarity and antecedents, further detailed below, there is no specific algorithm disclosed for any of these. The closest is the explanation of the flowchart of Figs. 3A and 3B, which calls for a series of operations that might be understood for NAND flash, but not at all for the abstract “memory elements” of claim 1. As an example, a “Write access operation” for NAND flash is generally observed as requiring “an ‘erase’ (of a block) before ‘write’” (3:43-44), but there is no algorithm provided for NAND flash “access,” “erase” or “write” (as explained, for example, at ¶¶ 30-37 above) and no description of NAND flash Read access operation or any of this for the other technologies sought to be covered by the post-2018 abstraction.

79. Claim 1 is indefinite as to when the [1e]-[1f] and [1i] write-read-compare-remap-transfer operation is performed. In another cross-element

defect, it is indefinite as to when the [1e]-[1f] write-read-compare operation is performed, and when the [1i1] remapping and [1i2] transfer operations are performed.³¹ In the embodiment described in the specification with two separate banks of MLC and SLC, “the controller conducts a data integrity test after every write.” (5:1-7 [emphasis added].) However, in the same column description of the (perhaps misnamed) “read-modify-write” scheme, the read-after-every-write execution is qualified by “if time permits.” (5:49-54.) The claim itself is silent. Is a single read-after-write-and-compare initiated by a user an infringement charged to the manufacturer? (See note 30 above.) A POSITA would not necessarily read claim 1 to require [1g1] performing after every “at least a Write operation” for the entire lifetime of the memory system. (See ¶ 66 above.) As I observed at paragraph 51 above, the bit-by-bit comparison of [1g3] is very inefficient; with reading the stored data into controller memory [1g2] to be compared to the retained data left there by

³¹ If “retained data” is in controller memory – which itself is ambiguously claimed and which would be limited in capacity – as it is controller memory recited in [1f] as where data is to be received (from the host), the transfer would need to be done promptly to make room for new incoming data. If retained data is in some external DRAM that has much more capacity, that transfer might be delayed until the system is idle or less busy or when the logical address is “accessed.”

[1f1] severely affects data throughput. This is in stark contrast to the prior art SanDisk systems that keep even incorrectly written data if they can be corrected by ECC upon reading and even defer deploying ECC until a statistically appropriate time. (See note 9 above.) It makes substantial difference which reading is chosen.

80. **Element [1a] is abstract, unsupported and indefinite.** The unsupported “memory space” construct [1a] (7:29-31), carried over from the ‘300 patent, is abstract, unsupported and indefinite as explained at paragraph 71 above. This abstraction and logical-physical ambivalence of “memory space” and “memory element” confuses the physical-versus-logical boundaries in the other claim elements, for example, detracting from the specificity required of the functionality of the controller recited in element [1b]. As explained at paragraphs 72, 73 and 77 above, where recitation of NAND flash modules in the first three patents provided the POSITA with some understanding how the recited components of the claimed systems worked together for preferential mapping into an SLC module, the technology-agnostic untethering of “memory space” (in the specification it’s a *capacity*) and “memory elements” leaves gaps in how data is read, written or erased and thus it is not clear how the recited claim components could work together as a useful system.

81. **Element [1b] is indefinite and unsupported.** As reviewed at paragraph 64 above, the “controller” in the Lifetime Patents specification is abstract,

not described structurally, even as a typical NAND flash controller such as described at paragraph 27 above. The specification simply shows in Fig. 1 a “device controller 14” schematically connected to a processor 12 (host), DRAM 20, disks 24, MLC flash 26 and SLC flash and IO 16 and 18. Where, as here, such a generic term as “controller” is used in the patent claim, I am advised that the functions that it performs as recited by the claims must be matched to structure or a software algorithm disclosed in the specification. In paragraph 78, I particularly applied that principle to identify stated and implied functions throughout the elements of claim 1 and concluded that the functions have no specifically disclosed structure (physical or algorithmic) in the specification. In element [1b] particularly, the controller is recited as “operat[ing] memory elements and associated memory space for associated Write access operations . . . and Read access operations from the memory elements.” The references to “memory elements” (physical?) “associated” with “memory space” (abstract, logical?) and the tautological “operat[ion]” of “associated” operations generalize away from structure. Although there is very general explanation in the specification as to writing in NAND flash (background on the need to erase before write, 2:38-58) there is nothing about a “Write access operation” for the abstracted memory elements that may include other technologies and nothing about a “Read access operation” at all. Thus, element [1b] is indefinite and unsupported.

82. **Element [1c] is unsupported and inoperative if “module” is not the POSITA-understood “chip” with understood number of “blocks.”** Element [1c] recites “at least one MLC nonvolatile memory element” and “at least one SLC nonvolatile element” within, respectively, the new formulations of “plurality of MLC modules” and “at least one SLC module” (7:38-41). This is not supported by the embodiment of “at least one” each of POSITA-understood physical MLC and SLC NAND modules each with a “plurality of individually erasable blocks” as recited in the specification SUMMARY (4:7-13) and in the ‘298 patent (7:9-12). As explained at paragraph 71 above, the abstraction to “element”³² is unsupported by any specification explanation of how such an element other than a NAND flash page or block is operated in a system described only as a NAND flash system with integrity-failing or wear-leveling mapping of received data into a physical SLC module. Thus, element [1c], as element [1a], is unsupported. If the recited “modules” were construed to be other than POSITA-understood, specification-enabled SLC and MLC NAND modules with their respective

³² As an example of the strain caused by the abstraction on the claims, the definition in [1c1] of MLC modules each including “at least one MLC nonvolatile memory element” fails to provide antecedent basis for [1f3] recitation of “nonvolatile [plural] memory elements in an associated MLC memory module.”

reading/writing/erasing circuitry (§ 62 above), not only would the claim element be unsupported, but the claim as a whole, without other provision for that circuitry, would be inoperative, failing as a useful system. As explained at paragraphs 62 and 73 above, the Lifetime Patents specification assumes a size of NAND memory in the recited modules on the order of 1K blocks for the alleged innovation to operate to improve lifetime of the system. Element [1c] calls for a plurality of MLC modules and at least one SLC module, each module only including “at least one [MLC/SLC] nonvolatile memory element. If “module” in element [1c] could mean inclusion of input/output circuitry and of the requisite “elements” only adequate to meet the “at least one” limitation, the claimed system would not be a useful one. The remapping and transfer recited in element [1i] could be performed only a few times, inoperative to improve, much less “maximize” lifetime (if element [1h] applies, see discussion below). It would be much less than the system disclosed by the specification as having multiple SLC and MLC modules each of a thousand blocks, useful for the disclosed and claimed preferential writing to an SLC module. In a minimal, single-blocks system as claimed, there could be no multiple writes to count given the NAND flash requirement of writing each update to another physical address, much less to enhance or maximize lifetime of the block or the system. And there is nothing in the specification describing a sufficient number of blocks or “memory elements”

needed for the claimed inventions to enhance or maximize lifetime, or any way to determine a sufficient number.

83. **Element [1d] is indefinite and unsupported, particularly for its recited function.** Element [1d] calls for “at least one random access volatile memory [RAVM] element.” RAVM is disclosed only as DRAM 20 in the specification. Generalizing that memory as an “element” makes claim element [1d]³³ indefinite by failing to inform a POSITA what it means and whether it has any structural commonality with the “elements” of claim element [1c], particularly of unexplained technologies. Even in the context of the NAND flash operation outlined in the specification, there is ambiguity in the term RAVM element and lack of support for its function in the rest of the claim. As explained at paragraph 71 above, the “RAVM element” was added as new matter in the ‘300 patent prosecution. If DRAM 20 is the RAVM referred to at step 102 of Fig. 3A, it is not disclosed as the RAVM recited in elements [1f4] and [1g3] as storing “retained data,” as the specification states that at step 102 (6:11-12), the data in DRAM is “read into

³³ In yet another example of abstract terminology mismatch, “at least one random access memory element” does not provide proper antecedent for [1f2] “the random access volatile memory [plural] elements for storage of the received data” recited for the operation of the claimed system (7:50-51 [emphasis added]).

memory within the device controller (which acts as the memory controller)” (6:12-13) and at step 106 combined (presumably there) with (valid) data in the targeted NAND flash memory (6:16-18), which appears to be the “retained data” referred to and compared at steps 114 through 120 (6:24-33, although step 114 seems to have the wrong referent for comparison [note 20 above]). That is, the RAVM for “retained data” in elements [1f4] and [1g3] are supported not by DRAM like DRAM 20 external to the controller, but the “memory within the device controller.” It is almost impossible that DRAM memory (generally a chip) could be internal to an SSD device controller (also generally a chip as shown in paragraph 27), due to the fact that DRAM fabrication processes are not compatible with those needed for an SSD device. It may be possible to have some RAM in the controller, but it is unclear whether it would be large enough for an entire block of 64 pages to be combined, compared and written to a different block. The specification does not describe any on-controller memory except by the quoted language, “memory within the device controller” (6:12-13). The ‘546 applicant argued that the controller memory and the RAVM are two distinct data stores (Sept. 22, 2023, Amendment at 11-17 [Ex. 1014 at 11-17]). Thus, element [1d] is indefinite and unsupported.

84. Element [1e] is indefinite and unsupported for other than NAND flash operation. A POSITA would understand “FTL” to be particular to NAND flash operation as described at paragraphs 27 and 28 above and sketched in the

specification BACKGROUND as typically firmware (location unspecified) for ordinary NAND flash operation including L2P mapping, wear leveling and garbage collection (2:61-3:26). The DETAILED DESCRIPTION proposes as an embodiment Fig. 4, which incorporates FTL in a physical module including banks of physical MLC and SLC modules as “another embodiment,” confirming the POSITA’s understanding of “modules.” While this understanding applies to claim element [1e] for NAND flash operation, the claim element is indefinite and unsupported if the recited memory elements are construed to include technologies such as CPM, RRAM or MAGRAM that operate under critically different principles from NAND flash (§ 71 above).

85. **Element [1f] is indefinite, unsupported and inoperative.** Element [1f] recites controller characteristics and functions in abstract terms setting up the compare test of element [1g]. Even reciting at [1f1] “having associated controller memory” (which may or may not be part of the controller), there is provided no structure for the controller component of the claimed system. The qualification [1f4] (operable to store data) does not provide structure other than for understood NAND flash module operations (where read and write [store] operations are performed at the memory chip level). Functional limitations [1f2] (controlling access of “memory elements”), [1f3] (transferring data to a “memory element”), as explained in paragraph 78 for all the sub-elements of [1f], find no structure (physical or

algorithmic) in the specification other than as assumed for NAND flash operation. The term “memory elements” has already been shown, at least in paragraphs 71 and 73 above, to be indefinite and lacking support. As explained in paragraph 81 regarding element [1b], the “Write access operation” recited there was only roughly described as NAND flash writing, which requires an empty or erased destination, and not at all for other technologies implicated by “memory elements”; it is therefore indefinite and lacking support. Element [1f] creates confusion as to where data for comparison in element [1g] is stored. As explained at paragraph 83 for element [1d], no controller memory referenced at [1f2] is disclosed in the specification other than by reference to “memory within the device controller” in explanation of Figs. 3A and 3B. This controller memory is referenced in [1f1] as “associated” with the controller for “storing received information therein.” However, [1f2] recites the controller’s function of controlling access of the MLC and SLC elements recited in [1c] and the RAVM element(s) recited in [1d] – not the controller memory – for “storage of the received data therein as stored data.” Element [1f3] calls for “transferring the stored received data” from the controller memory to MLC. Element [1f4] calls for the controller to be “operable to store” the received data [apparently in the controller memory] in MLC and “retain such received data” [apparently in the controller memory] in the RAVM as “retained data associated with the stored data. As explained in paragraph 83, the specification doesn’t specify

but appears to assume that data to be written is combined with valid data in the originally targeted NAND flash location in the internal controller memory and left there as retained data. Element [1f4], inconsistently with both the recited [1f3] action and the DETAILED DESCRIPTION, locates “retained data” in RAVM (“retain such received data in the random access volatile memory as retained data”). In element [1f1] the received data is in the controller memory. In element [1f2], the received data is “transferred” from controller memory to MLC, but it is not stated that it is erased rather than retained. In contrast, there is no step in [1f] or the specification that moves data to RAVM to be written from there to MLC but also retained there. As explained in paragraph 84, in the specification the retained data is the data received from DRAM, combined in controller memory with existing valid data at the original target address, to be written to a new address, but apparently left in controller memory. This inconsistency renders element [1f] indefinite and unsupported. A member of the public would not know whether to avoid infringing claim 1 by placing retained data in one or the other of controller memory or RAVM (they have not been shown or claimed to be the same). Because the received data is not in RAVM to be retained, element [1f] is also inoperative.

86. **Element [1g] is indefinite, unsupported.** Element [1g] is an abstraction (first added in ‘300 as discussed at ¶ 73 above) of the NAND flash operation comparison check disclosed through step 114 of Figs. 3A and 3B (and

claimed as ‘278 patent controller function b)). Again, the term “memory elements” has already been shown, at least in paragraphs 72 and 74 above, to be indefinite and lacking support, along with “Write access operation” for non-NAND flash implementations, as explained at paragraphs 79 and 82 above. Although the claim element [1g2] corrects (as it arguably may not) the specification’s error of what is compared (§ 64 above), the inconsistency remains as to where the retained data recited at [1g3] is located. As reviewed in paragraph 84 above, both the specification and the [1f] steps suggest the “retained data” is in “the controller memory”; however, [1f4] labels “as retained data” the “received data” in RAVM, where it was never placed in [1f]. This inconsistency is not fixed but added to in [1g3] ungrammatically “comparing the stored data in the controller memory in the given one of the MLC nonvolatile memory elements” with the retained data in RAVM, where it never was. A member of the public would not know whether to avoid infringing claim 1 by placing retained data in one or the other of controller memory or RAVM or by avoiding comparison in the controller memory the stored data in MLC. As written, elements [1f] and [1g] are indefinite, unsupported and inoperative.

87. **Element [1h] is abstract, grossly indefinite, unsupported and inoperative to achieve the benefits of the Lifetime patents based on superior SLC endurance.** Element [1h], carried over verbatim from the only limitation of claim 1 of the ‘300 patent application as filed in 2018, nearly seven years after the

priority date, is criticized at paragraphs 72 and 73 above as unworkably abstract and as abandoning the clear objective of the Lifetime Patents specification to avoid premature system degradation by availing of superior SLC endurance. The specification itself makes no claim to maximize lifetime, but only “to boost the lifetime” or “provide long lifetime” for a NAND flash storage system. In the context of ‘546 patent claim 1, even the preliminary element [1h1] calling for L2P mapping “adaptable to the system” has no meaning except possibly to support the proposed reach of claim 1 to non-NAND flash systems that may need adaptation of the claim. For that reason alone, the element fails because there is no support in the specification for a non-NAND flash system. In addition to the criticisms previously made particularly about the gross abstraction of [1h2] where “mapping is performed as necessitated by the system to maximize lifetime” (8:22-24), I note that it is unsupported where the DETAILED DESCRIPTION (1) compares a range of up to 50,000 to 100,000 writes for SLC and up to 3,000 to 10,000 for MLC (3:35-41), (2) states a requirement “to manage the methods of writing data optimally for improved life-time (endurance) of flash memory, and (3) states that it describes “various embodiments of a NAND flash storage system that provides long lifetime (endurance) storage at low costs.” It would be absurd to apply [1h2] to every L2P mapping in the operation of the system, including [1h3] mapping to imagined non-NAND flash memory system “elements,” even down to the byte level in volatile

memories. Is each mapping on any operation to be measured against a maximization goal? Even in the context of NAND flash, does each mapping to NAND flash maximize lifetime? If followed by a stress such as a write or erase, the mapping would at least marginally degrade the lifetime of the written block, SLC or MLC. In the context of ‘546 claim 1, does [1h2] apply to all instances of [1e] FTL and [1f] necessary mapping, or only to [1i] remapping after [1g] compare? Without the wear-leveling limitations, for example, the ‘298 controller functions c) and d), the only life-enhancing mapping in claim 1 is the one invoked by element [1i]. But that element, as discussed in the next paragraph, does not favor remapping into an SLC module, as the supposed life-extending innovation of the Lifetime Patents. In view of its context in ‘546 claim 1, it can be said that element [1h], like the entire claim 1, is inoperative to extend lifetime by preferential mapping and writing to SLC. At very minimum, a POSITA would be left with no clear idea how to *maximize* lifetime after reading the ‘546 disclosure and claims.

88. **Element [1i] is indefinite and inoperative to achieve the benefits of the Lifetime patents based on superior SLC endurance.** In addition to claiming a result rather than the steps to reach that result, element [1i] has a serious ambiguity: the source of transfer in element [1i2] is “from those physical addresses determined to have failed the data integrity test” – ridiculously resulting in propagation of failed data. Even if [1i2] is read as transferring the “retained data,” as explained at

paragraphs 85 and 86 above, elements [1f] and [1g] incorrectly place the “retained data” in the RAVM rather than in the controller memory where it is left both in the specification and [1f] – and not in RAVM. Element [1i] requires remapping not into an SLC module as disclosed in the specification, but “to a different physical range,” that in most hybrid SLC-MLC systems in the proportion .125-.25:1 as suggested at 5:10-12, would favor MLC rather than SLC according to the alleged innovation of the Lifetime Patents. Moreover, element [1i] did not repeat the “to achieve enhanced endurance” limitation of the comparable elements in the ‘300 patent (8:10-11, 9:8-9), changing any coupling to [1h] mapping “to maximize lifetime” and eliminating the specific objective of endurance. Thus, element [1i] is indefinite at least in its reference to in incorrectly placed “retained data” and inoperative for that reason and its abandonment of the approach for preferential mapping or remapping to superior endurance SLC.

89. At least for the above vagaries, contradictions and departures from the disclosure, **claim 1** would not be understood by a POSITA to definitely claim what it purports to cover. The abstractions in the claim elements to “space” and “elements,” apparently for non-NAND flash implementations, and other departures from the specification and rationalizing of specification ambiguity, are not supported by the disclosure, and as a whole, the claim fails to operate to achieve the benefits

of preferential mapping to an SLC module. Dependent claims 2-7 also fail for these reasons, but claims 3 and 7 further fail independently.

90. **Claim 3**, “wherein at least one of the random access volatile memory or the MLC and SLC nonvolatile memory elements are embedded in at least one controller” (8:37-40) is unsupported insofar as there is no disclosure of any of these in the device controller (or of any physical content of the controller). The incorporated Embedded RAM Patents, *e.g.*, Rao ‘452 (Ex. 1004) relate to use of volatile memory on the flash chip, although they show at Fig. 3 a controller with registers presumably using temporary memory. In my experience, only an SRAM could possibly be embedded in the controller (typically a discrete chip as shown at paragraph 27 above), because the fabrication process for DRAM, such as suggested in Lifetime Patents Fig. 1 (20) and Fig. 3A (102), is not compatible with the fabrication process for a CMOS controller chip. Obviously, MLC and SLC, in the quantity for which the ‘546 patent would be operative, could not be embedded in the controller. The [1d] antecedent of “at least one [RAVM] element” and the inconsistent placement in [1f] and [1g] of “retained data” in discretely referenced “controller memory” and “RAVM” could only be reconciled (but unsupported) as the “RAVM” and “controller memory” being the same. And yet, as explained at paragraph 83, the ‘546 applicant argued that the controller memory and the RAVM

are two distinct data stores (Sept. 22, 2023, Amendment at 11-17 [Ex. 1014 at 17-19]). Claim 3 is independently indefinite and unsupported.

91. **Claim 7**, “wherein the at least one SLC nonvolatile memory element includes a hard disk drive (HDD)” (8:47-49) is on its face contradictory if “element” is understood to be some physical component of an SLC memory module (as in claim 1, 7:41-42), which is clearly understood at least to involve NAND flash technology described in the specification. Besides, a POSITA would understand that it is physically impossible to install or somehow include a hard disk drive (HDD) on an SLC chip since the HDD is many orders of magnitude larger. Claim 7 is independently indefinite and unsupported.

92. **Claim Construction:** The fundamental changes from physical SLC and MLC modules to abstract “memory space” and “memory element” as well as from preferential NAND flash mapping into an SLC module to “maximiz[ing] lifetime” should invalidate the claims for the grounds reviewed above. If for some reason the claims survive, they may need construction for infringement analysis. However, other than a POSITA’s ordinary and customary understanding of the specifically recited “SLC memory modules” and “MLC memory modules” as specific types of NAND flash modules, construction is not required for invalidity analysis based on art as set forth in the next section. The NAND flash technology background reviewed in Section V included the principles and features restated in

the Lifetime Patents specifications, with the POSITA's understanding of SLC and MLC NAND flash modules filling the gaps in the specification for NAND flash operation that support the controller functions recited in the claims. The background art discloses remapping to SLC to enhance hybrid SLC-MLC system lifetime in invalidating instances of the '546 patent claim "mapping to maximize lifetime" of broader technology.

X. Patentability of the '546 Claims Relative to the Art

93. The independent claim 1 of the '546 is inclusive of NAND flash post-write read bad-block management and wear leveling detailed in multiple prior art textbooks, articles and patents reviewed in Section V(C). Gavens (Ex. 1045), among the SanDisk portfolio and incorporating by reference others of that portfolio, currently appears to the Petitioner as the one location in which the recited limitations of '546 claim 1 are most clearly disclosed. However, Gavens, as in most patents of the 2010 timeframe found by Petitioner and the '546 Examiner relevant to hybrid one- or multi-level NAND flash operation, involve the partitioning and configuration of some MLC blocks for one-level (two states) or pseudo-SLC (pSLC) operation. This does not define the physical SLC module disclosed in the specification, claimed in the pre-'300 Lifetime Patents, and reintroduced into '546. However, as reviewed at paragraph 43 above and in Gavens itself (*e.g.*, 12:6-10) the pSLC mode of operations enjoys superior endurance compared to MLC in multi-

level mode, and it would be obvious to a POSITA to have one MLC module operating entirely in pSLC mode with expectation of success for the relative superiority to MLC in multi-level mode³⁴ or use a “real” SLC module as understood by a POSITA from the embodiments actually disclosed in the specification.

A. Claim 1 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.

94. In case claim 1 is not invalidated for the reasons detailed at paragraphs 79-89 above, I address the limitations recited in claim 1 as set forth in the groupings as printed in the ‘546 patent, further labeled as in paragraph 75 above:

1. [1Pre] A system for storing data comprising:

95. I understand the question of whether a preamble is limiting is a question of claim construction. I do not provide an opinion as to whether the preamble of claim 1 is limiting.

96. To the extent the preamble is limiting, Gavens discloses “a system for storing data,” namely, the “flash memory device 90” that includes memory array 200. (Gavens, Fig. 1, ref. 90 and 16:41-17:20).

³⁴ This would meet the Lifetime Patents specification reservation that “SLC module” may be “any other solid-state non-volatile memory as long as it has endurance that is superior to that of MLC flash” (4:26-29).

2. [1a] [1a1] memory space containing volatile memory space and [1a2] nonvolatile memory space, wherein the nonvolatile memory space includes both multi-level cell (MLC) space and single level cell (SLC) space;

97. [1a1] To the extent that this clause is not invalid as explained at least in paragraph 80 above, and “memory space” is understood to include all memory in the claimed system, Gavens discloses, or at least renders obvious, “memory space containing volatile memory space.”

98. Gavens discloses memory containing volatile RAM and non-volatile memory array. (1:26: RAM; 1:18: Non-volatile memory; Fig. 1: Memory array 200 in memory chip 100.)

99. [1a2] Gavens’ non-volatile memory array includes a first portion of pseudo-SLC memory and a second portion of MLC memory. Specifically, Gavens states that 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. (4:32-34; 17:21-23.) 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. (5:15-17, 50-53, 16:22-24, 20:31-34.) As disclosed in 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. For further reference, see: 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; Fig. 8, col. 12:11-12, 31-33: A multi-state memory, for example, a triple-level cell or multi-level cell (TLC) or D3 memory has a population of cells with each cell in one of eight possible states, whereas a multi-level cell (MLC) or D2 memory has a population of cells with each cell in one of four possible states; Abstract, 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; 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; 20:53-59: The entire disclosure of Gorobets U.S. Appl. Ser. No. 12/642,584, published as U.S. Patent Appl. Pub. No. 2011/0153912 (“Gorobets ‘912,” Ex. 1049) is incorporated into Gavens by reference:

- Gorobets ‘912 fig. 1, ¶ 56: RAM 130 in the controller 100;
- Gorobets ‘912 fig. 11, ¶ 109: Cache RAM 102 in the controller;
- Gorobets ‘912 fig. 20, ¶ 130: RAM 511 in the controller and in the memory 503;
- Gorobets ‘912 ¶ 18: Controller RAM.

These configuration details are understood by a POSITA to be consistent with the system disclosed in Gavens, which would be understood, as incorporated disclosures below, to focus on different features of the more inclusive SanDisk NAND flash

memory systems. Not only are the features disclosed, but in my analysis here, they are already combined with more than just a reasonable expectation of success. Thus, Gavens and incorporated references disclose claim element [1a].

3. [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 operations from the memory elements;

100. 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. For example, Gavens discloses a controller that manages read operations and write operations on memory. (Fig. 1 ref. 102; 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; 2:56-58: A controller performs intelligent and higher level memory operations and interfacing.) Thus, Gavens discloses claim element [1b].

4. [1c] a bank of nonvolatile memory, including: [1c1] a plurality of MLC memory modules, each including at least one MLC nonvolatile memory element, and [1c2] at least one SLC memory module including at least one SLC nonvolatile memory element;

101. [1c] 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 [Ex. 1042], Moshayedi [Ex. 1043], and Lee '794 [Ex. 1041] as explained below.).

102. Gavens (Ex. 1045) discloses a bank of non-volatile memory chips including MLC (NAND flash) memory. (Fig. 1; 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; 1:18: Non-volatile memory; 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; 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; Fig. 14A, col. 16:15-22: The array of memory cells 200 is partitioned into a first portion 410 configured for lower density (*e.g.*, pseudo-SLC) operation and a second portion 420 configured for high density (*e.g.*, native MLC, having each cell storing multiple bits) operation; 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; 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; 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; Fig.

8; 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 four possible states; Abstract; 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; Fig. 15 ref. 500; 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; Fig. 6; 11:9-10: A memory array organized in erasable blocks; 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 as understood in NAND flash operation.)

103. As I explained at paragraph 93 above, Gavens differs from the claimed SLC and MLC NAND flash modules by having some portions of its aggregated MLC modules partitioned and configured to operate in pseudo-SLC (2-state or 1 bit) mode. (8:18-20 [“The memory chip 100 includes memory array 200 of memory cells with each cell capable of being configured as a multi-level cell (‘MLC’)]; 18:12-20 [explaining the array of memory cells 200 “partitioned” into portions with higher density (more bits per cell) and lesser density (fewer bits per cell) operation with associated margins of error; 17:59 [“A first portion is configured to store one bit per cell,” citing U.S. Patent No. 5,930,167 to Lee et al. [“Lee ‘167,” Ex. 1050], incorporated by reference.]) Gavens clearly satisfies the [1c1] limitation of multiple MLC memory modules in the understood operation of a NAND flash system that

employs Gavens's schemes of preferential writing to lower density memory. As to the [1c2] limitation of an SLC memory module, it would be obvious to a POSITA to have one MLC module operating entirely in pSLC mode with expectation of success for the relative superiority to MLC in multi-level mode or substitute a "real" SLC module understood by a POSITA to have even better endurance than pseudo-SLC (see ¶ 43 above), both with a clear expectation of success. Use of dedicated SLC and MLC modules for preferential writing was well-known prior to the priority date (¶¶ 40-42):

104. Lee '794 (Ex. 1041) discloses an allocation of logical blocks pre-determined to have most frequent writes (updates, such as system data and user directories) to a distinct SLC module and less frequent writes (user data) to a distinct MLC module. (Lee '794 [Ex. 1041] Fig. 2 refs. 226, 228, 5:10-13: The SLC flash memory array 226 portion of the module would include at least one flash memory "element." Likewise, the MLC flash memory array 228 portion of the module would include at least one MLC flash memory "element"; 6:53-7:39: allocation of logical blocks pre-determined to have most frequent writes (updates, such as system data and user directories) to a distinct SLC module and less frequent writes (user data) to a distinct MLC module.)

105. Sutardja (Ex. 1042) discloses a controller that is adapted to allocate those blocks (understood to be NAND flash) that receive the most frequent writes

by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module (disclosed as higher-cost, faster memory distinct from another lower-cost, slower memory). (Sutardja [Ex. 1042] ¶ 141).

106. Moshayedi (Ex. 1043) discloses a controller that is adapted to allocate those blocks (again understood to be NAND flash) that receive the most frequent writes by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module (explicitly disclosed as distinct from an MLC non-volatile memory module). (Moshayedi, Ex. 1043 [Abstract] disclosing a system that keeps frequently written data, which results in frequently erased blocks, in the SLC flash module, and relatively static data in MLC flash module.

107. Thus, Gavens discloses claim element [1c] with the POSITA's knowledge of obvious substitution of a "pure" SLC module for its partitioned and 2-state (pseudo-SLC) configured MLC module.

5. [1d] at least one random access volatile memory element;

108. Gavens discloses at least one random access volatile memory element. For example, Gavens discloses volatile RAM. (1:26: RAM; Fig. 16A: First section 411 for caching in first portion 410; 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; 20:53-59: The entire disclosure of Gorobets '912 [Ex. 1049], is incorporated by reference:

- Gorobets ‘912 fig. 1, ¶ 56: RAM 130 in the controller 100.
- Gorobets ‘912 fig. 11, ¶ 109: Cache RAM 102 in the controller.
- Gorobets ‘912 fig. 20, ¶ 130: RAM 511 in the controller and in the memory 503.
- Gorobets ‘912 ¶ 18: Controller RAM.)

Again, my view in paragraph 99 applies, and thus Gavens and incorporated references disclose claim element [1d].

6. [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;

109. The ‘546 patent uses “flash translation layer (FTL)” to refer to the well-known concept of translating logical host addresses to physical flash addresses, *i.e.*, the functionality that implements the logical-to-physical address map. (2:40-43.) That is, the FTL “make[s] linear flash memory [flash addresses] appear to the system like a disk drive [via logical host addresses].” (3:61-65.)

110. 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. For example, Gavens discloses that a controller implements a memory block management system that maintains a directory in memory. (Fig. 1 ref. 102; 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; 2:56-58: A controller performs intelligent and higher level memory operations and interfacing; Fig. 15 ref. 550; 17:39-40: Identifying the last written copy as valid data for a subsequent read; 20:53-59.)

111. Gavens 20:53-59: The entire disclosure of Gorobets ‘912 (Ex. 1049), is incorporated by reference. (Gorobets ‘912 Figs. 8-9, ¶¶ 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 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.)

112. See also Gavens 8:40-43: The entire disclosure of U.S. Patent Appl. Pub. No. 2010/0172180 by Paley (“Paley ‘180,” Ex. 1051) is incorporated by

reference. (Paley '180 figs. 8-9, ¶¶ 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.)

113. See also U.S. Prov. Pat. Appl. No. 61/509,257 (Ex. 1003, admitted prior art as stated in the “Description of the Prior Art” section) ¶ 7, also in the BACKGROUND of '546, as reviewed at ¶ 60 above): The flash translation layer (FTL) performs the function of mapping logical pages to physical flash pages.

114. Accordingly, Gavens discloses, or at least renders obvious, that “wherein the at least one controller, or FTL, or a combination of both [Gavens’ controller implementing the FTL layer] maintain an address table [Gavens’

directory, *i.e.*, logical-to-physical map] in one or more of the memory elements [Gavens' non-volatile memory] and random access volatile memory [RAM in Gavens' controller]." Thus, Gavens discloses claim element [1e].

7. [1f] the controller [1f1] having associated controller memory for storing received data therein, the controller [1f2] 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, [1f3] 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, [1f4] 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;

115. 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.

116. 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. (Fig. 1 ref. 102, 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; 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; 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; Fig. 14A: The write input data page is the first copy of the data page in the second portion 420 of higher density; Fig. 15 ref. 510, col. 17:30-31: Programming a first copy of a group of input data in the second portion; 4:18-19, 15:66-67: Data is written to the second portion; 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; 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; 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; Fig. 16A, 18:9-12: A cached copy of data page in the first section 411 the of first portion 410; 5:53-54: Input data is first staged in D1 (SLC) and subsequently folded into D3 (TLC).)

117. See also Gavens 16:29-40: The entire disclosure of Chen ‘528 is incorporated by reference. (Chen ‘528 [Ex. 1040] 4:39-41: The controller 20 contains buffer memory for the user data being written to or read from the memory array.)

118. Further, Gavens 20:53-59: The entire disclosure of Gorobets ‘912 (Ex. 1049) is incorporated by reference:

- Gorobets ‘912 ¶¶ 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.
- Gorobets ‘912 abs., ¶ 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.
- Gorobets ‘912 ¶ 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.

119. See also 17:57-67: The entire disclosure of Lee US5,930,167 is incorporated by reference. (Lee ‘167 [Ex. 1050] Abstract, 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.)

120. 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. (Gavens 20:53-59: The entire disclosure of Gorobets ‘912 [Ex. 1049] is incorporated by reference. Gorobets ‘912 ¶¶ 24, 109: Use of RAM as a controller cache of the memory controller.)

121. Despite the ambiguity in claim elements [1f] and [1g] as to “controller memory” and RAVM, explained at paragraphs 85 and 86 above, if the claim survives 35 U.S.C. § 112 challenge, Gavens, as reviewed above, discloses [1f1] “associated controller memory” and [1f4] RAVM in its cache and by incorporated disclosures that are consistent with Gavens operations for holding incoming data written to the MLC module. Gavens thus discloses this limitation [1f].

8. [1g] the controller [1g1] 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 [1g2] reading the stored data to the controller memory and [1g3] 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;

122. 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.

123. 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. (Abstract; 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; 4:19-21, 15:67-16:1: After the data is

written to the second portion, the data is read back to check for excessive error bits; 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; 6:4-6, 24:45-50: Post-write reading every memory cell to check what has been written; Title: Post-write read to manage errors.)

124. See also Fig. 15 ref. 520, 17:32-33: Reading the first copy from the second portion; 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; 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; 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; 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. Gavens expressly discloses that this error check may be “either by comparison with the original copy which may be cached **or** by checking the EDC portion of the ECC” (16:47-49 [emphasis added]). Although Rao’s direct comparison is wasteful, it was a known alternative. (See ¶¶ 47, 49 and 50 above).

125. 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. (20:53-59): The entire disclosure of Gorobets '912, is incorporated by reference:

- Gorobets '912 ¶¶ 24, 109: Use RAM as a controller cache of the memory controller.
- Gorobets '912 ¶ 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 '912 ¶¶ 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.

126. See also Gavens 8:40-43: The entire disclosure of Paley '180 (Ex. 1049) is incorporated by reference. (Paley '180 ¶ 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.

127. Accordingly, Gavens discloses this limitation [1g].

9. [1h] wherein [1h1] the address table maps logical and physical addresses adaptable to the system, wherein [1h2] the mapping is performed as necessitated by the system to maximize lifetime, and wherein [1h3] the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and

128. 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.

129. Relative to limitation [1h1], Gavens discloses that the directory maps logical addresses and physical addresses adaptable to the NAND flash memory device. Relative to limitation [1h2], to the extent it can be understood if it survives 35 U.S.C. § 112 scrutiny, it would met by the kind of NAND flash wear-management disclosed in Gavens and long known in the art (¶¶ 28, 29, 40-45, and 51-56 above) (or alternatively, the remapping of limitations [1f], [1g] an [1i]): the mapping is performed as necessitated (implemented) by the flash memory device to maximize (increase) the (expected) lifetime of the flash memory device, and [1h3] the mapping maps a block of data in the non-volatile memory array. (8:36-39: A memory block management system implemented in the controller maps the logical sectors to the physical structure of the memory array; 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; 16:34-37: It is desirable to have

longer operational life of memory cells; Fig. 15 ref. 550, col. 17:39-40: Identifying the last written copy as valid data for a subsequent read; 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; 21:21-23: Each block contains data from a well-defined group of logical addresses; 5:41-45: A hot count tracks the number of times each physical block has been cycled through erase and program operations.

130. See also Gavens 8:40-43: The entire disclosure of Paley '180 is incorporated by reference. (Paley '180 [Ex. 1051] Figs. 8-9, ¶¶ 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 metablock 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.)

131. See further Gavens 20:53-59: The entire disclosure of Gorobets '912 is incorporated by reference. (Gorobets '912 [Ex. 1049] Figs. 8-9, ¶¶ 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.)

132. See also Gavens 17:57-67: The entire disclosure of Lee '167 is incorporated by reference. (Lee '167 [Ex. 1050] 2:54-3:7: The usual desire to evenly wear the memory. Avoid using one portion of memory much more than other portions. Maintain separate counts of the number of times that each sector has been

erased and programmed to even the wear among the sectors); Gavens 16:29-40: The entire disclosure of Chen '528 (Ex. 1040) is incorporated by reference (Chen '528 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).

133. Thus, element [1h] is met by the more specific system life-extending approaches disclosed in Gavens, either background “wear leveling” reviewed above or remapping which is disclosed in Gavens to meet elements [1f], [1g] and 1[i]

10. [1i] wherein a failure of the data integrity test performed on the stored data by the controller results in [1i1] a remapping of address space to a different physical range of addresses and [1i2] 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.

134. 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.

135. 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. (Fig. 14B: If post-write read failed, then rewrite the input data page as a second copy of data page in first portion 410 of lower density; 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 fewer data bits per cell; 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). Again, Gavens expressly discloses that this error check may be “either by comparison with the original copy which may be cached **or** by checking the EDC portion of the ECC” (16:47-49 [emphasis added]).

136. See also Gavens Fig. 15, refs. 530, 540, 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; Abstract, 6:23-25: If the copy in the second portion has excessive error bits, then a copy is written to the first portion; 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; 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. Thus, Gavens discloses limitation [1i1].

137. See also 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; 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. Thus, Gavens discloses limitation [1i2] and element [1i].

138. In the rewriting reviewed in the immediately preceding paragraph showing disclosure of limitation [1i1], the rewriting is clearly a transfer of “data corresponding to retained data,” such as in “the original data which was cached” to “those remapped physical addresses from those physical addresses determined to have failed the data integrity test,” that is, the “bad” MLC blocks. Thus, Gavens discloses limitation [1i2].

* * * * *

139. In summary, Gavens, with its incorporated references, discloses each limitation of ‘546 claim 1 as understandable, with the POSITA’s understanding that

Gavens's extensive technology-justified teachings are readily combined and extended (with clear expectation of success) to meet the claim's base MLC and SLC modules and unsupported abstractions.

B. Claim 2 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.

1. [2Pre] The system of claim 1,

140. I incorporate here my opinions set forth in Section X(A) for claim 1.

2. [2] wherein the FTL is a software module, or a firmware module containing software updates.

141. Gavens discloses the FTL is a software module, or a firmware module containing software updates. For example, Gavens discloses that the memory block management system implemented in the controller is software. (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; Fig. 1, ref. 60, 8:44-45: The firmware 60 provides code to implement the functions of the controller 102; 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; 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.)

142. See also 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. Thus, Gavens, in view of the knowledge of a POSITA, discloses claim 2.

C. Claim 3 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.

1. [3Pre] The system of claim 1,

143. I incorporate here my opinions set forth in Section X(A) for claim 1.

2. [3] 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.

144. 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.

145. For example, Gavens discloses that the RAM is embedded in the controller. (20:53-59: The entire disclosure of Gorobets '912 [Ex. 1049] is incorporated by reference:

- Gorobets ‘912 fig. 1, ¶ 56: RAM 130 in the controller 100;
- Gorobets ‘912 fig. 11, ¶ 109: Cache RAM 102 in the controller;
- Gorobets ‘912 fig. 20, ¶ 130: RAM 511 in the controller;
- Gorobets ‘912 ¶ 18: Controller RAM.)

146. See also Gavens (8:40-43: The entire disclosure of Paley ‘180 [Ex. 1050] is incorporated by reference:

- Paley ‘180 Fig. 1, ¶ 119: RAM 130 in the controller 100;
- Paley ‘180 Fig. 15, ¶ 197: Cache RAM 102 in the controller).

147. Thus, Gavens, in view of the knowledge of the POSITA, discloses claim 3.

D. Claim 4 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.

1. [4Pre] The system of claim 1,

148. I incorporate here my opinions set forth in Section X(A) for claim 1.

2. [4] wherein the MLC and SLC nonvolatile memory elements comprise flash memory.

149. Gavens, which would be understood by a POSITA to describe a NAND flash system with multiple MLC NAND flash modules, expressly describes its MLC non-volatile memory modules as including MLC NAND flash memory elements, which by definition are flash memory. (Fig. 4, 9:49-50: A page of memory cells in the memory array 200 is organized in the NAND configuration; 1:18-19, 4:14-18:

Flash memory having a first portion and a second portion storing higher density compared to the first portion; 2:65-66, 8:65-67: Flash memory devices with NAND string structures; Fig. 1 ref. 90: Flash memory device.)

150. As explained at paragraphs 93 and 101-107, a POSITA would readily and successfully substitute for Gavens' partitions of MLC memory elements configured for 2-state pseudo-SLC operation an SLC non-volatile memory module which is understood to include SLC NAND flash memory elements. As to Gavens's MLC NAND flash memory operated in pseudo-SLC mode, they are also described as flash memory. (Fig. 4, col. 9:49-50: A page of memory cells in the memory array 200 is organized in the NAND configuration; 1:18-19, 4:14-18: Flash memory having a first portion and a second portion storing higher density compared to the first portion; 2:65-66, 8:65-67: Flash memory devices with NAND string structures; Fig. 1: Flash memory device 90.)

151. Thus, Gavens, in view of the knowledge of the POSITA, discloses claim 4.

E. Claim 5 Is Obvious Over Gavens, Including Incorporated References, in View of Knowledge of the POSITA.

1. [5Pre] The system of claim 1,

152. I incorporate here my opinions set forth in Section X(A) for claim 1.

2. [5] The system of claim 1, wherein the random access volatile memory element is dynamic random access memory.

153. In view of a POSITA's understanding that dynamic random access memory (DRAM) is common type of random access memory (RAVM) that may be used for purposes of temporarily holding data to be compared as is the recited role of RAVM in claim 1, it would be an obvious choice for Gavens's alternative data comparison (16:47-49). The background art supports such use. For example, Moshayedi (Ex. 1043) discloses dynamic random access memory. (Moshayedi, Fig. 1, ref. 106, ¶¶ 3-4, 28, 53-55, 67: DRAM.)

154. Further, Duann (Ex. 1034) discloses dynamic random access memory. (Duann page 6: Utilize DRAM as cache.) Thus, Gavens, in view of the knowledge of the POSITA, discloses claim 5.

F. Claim 6 Is Obvious Over Gavens, Including Incorporated references, in View of Knowledge of the POSITA.

1. [6Pre] The system of claim 1,

155. I incorporate here my opinions set forth in Section X(A) for claim 1.

2. [6] wherein the random access volatile memory element is static random access memory.

156. Similar to claim 5, in view of a POSITA's understanding that static random access memory (SRAM) is common type of random access memory (RAVM) that may be used for purposes of temporarily holding data to be compared as is the recited role of RAVM in claim 1, it would be an obvious choice for Gavens's

alternative data comparison (16:47-49). Gavens itself discloses use of SRAM. (8:40-43: The entire disclosure of Paley ‘180 [Ex. 1051] is incorporated by reference. Paley ‘180 ¶¶ 423-425, 502-503, 506: SRAM.) Other uses with NAND flash systems were known, including Duann (Ex. 1034) for a hybrid SLC and MLC system. (Duann page 6: SRAM as cache.) The incorporated Embedded RAM Patents, published March 10, 2011, before the ‘546 priority date, specifically proposed using the “SRAM of the block” among 1,024 blocks of a commercial NAND flash module (Rao ‘452 [Ex. 1004] 2:19-39, Figs. 6a and 6b) to enhance speed of reading and writing. Thus, Gavens, in view of the knowledge of the POSITA, discloses claim 6.

G. Claim 7 Is Obvious Over Gavens, Including Incorporated references, in View of Knowledge of the POSITA and/or Colligan.

1. [7Pre] The system of claim 1,

157. I incorporate here my opinions set forth in Section X(A) for claim 1.

2. [7] wherein the at least one SLC nonvolatile memory element includes a hard disk drive (HDD).

158. A POSITA would understand that it is physically impossible for an SLC flash memory chip (much less an included “element”) to include a hard disk drive, where the latter is orders of magnitude larger than a flash chip. If this and all the inconsistencies with claim 1 were ignored, including the antecedent of being a part of an “SLC” module and changing the nature of an “SLC nonvolatile memory

element,” a POSITA would know that there are some useful combinations of flash and HDD such as disclosed in U.S. Patent No. 8,225,035 to Colligan (Ex. 1052) 2:33-36, 5:40-6:6, 7:67-8:2, and 8:23-25 (flash memory could supplement HDDs for more frequent writes) and the POSITA could **call** a combination of disk(s) 24 and SLC flash 28 of ‘546 Fig. 1 an “SLC nonvolatile memory element” to replace bad MLC blocks as physical address destinations. Thus, Gavens, in view of the knowledge of the POSITA, discloses claim 7 with an overstretched construction of “element.”

XI. Conclusion

159. For the reasons given above, claims 1-7 of the ‘546 patent are invalid.

160. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: August 26, 2024



Carl Sechen, Ph.D.

Carl Sechen

Professor
Dept. of Electrical Engineering
University of Texas at Dallas
carl.sechen@utdallas.edu
972-835-1611

Research Interests

My research interests center primarily on the design and computer-aided design of digital and analog integrated circuits. Ongoing projects include the design of secure ICs that cannot be reverse engineered. One aspect of this is the design of novel types of embeddable transistor-level field-programmable circuits for design obfuscation. We are also working on the design of optimal standard cell libraries.

Education

Ph.D., Electrical Engineering, University of California, Berkeley, 1986
Thesis: *Placement and Global Routing of Integrated Circuits Using Simulated Annealing*
Advisor: Prof. Alberto Sangiovanni-Vincentelli
M.S., Electrical Engineering, Massachusetts Institute of Technology, 1977
Advisor: Prof. Stephen Senturia
B.E.E., Electrical Engineering, University of Minnesota, 1975

Employment history

Professor, University of Texas at Dallas August 15, 2005 – present
Professor, University of Washington July 1999 – August 14, 2005
Associate Professor, University of Washington July 1992 – June 1999
Associate Professor, Yale University July 1990 – June 1992
Assistant Professor, Yale University July 1986 – June 1990

University Administrative Positions

Director of ECS Tech Support Services, UTD, June 2012 – February 2014

Honors and Awards

- *Best Paper Award* at the 2017 IEEE PhD Research in Microelectronics and Electronics Conference (PRIME), for the paper “Improved Lagrangian Relaxation-based Gate Size and VT Assignment for Very Large Circuits”, Bariloche, Argentina, February 2017.
- Nominated for a *Best Paper Award* for: “A Field Programmable Transistor Array Featuring Single-Cycle Partial/Full Dynamic Reconfiguration”, *Proc. of Design Automation and Test in Europe (DATE) Conference*, March 27-31, 2017, Lausanne, Switzerland.
- Nominated for UTD’s President’s Teaching Excellence Award for Graduate/Professional Instruction, 2020.
- Received the *Graduate Teaching Excellence Award* from the ECE Department at UT Dallas, May 2024.
- Received the *Distinguished Teaching Award* for the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, 2014.
- Received the *Distinguished Teacher of the Year Award*, Dept. of Electrical Engineering, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, 2008.
- Elected *IEEE Life Fellow* in 2002

- Elected a Fellow of Asia-Pacific Artificial Intelligence Association (AAIA)
- Received the *Outstanding Research Advisor Award*, Department of Electrical Engineering, University of Washington, 2002.
- Received the *Best Project Award*, NSF Center for the Design of Digital and Analog ICs (CDADIC), July 2002.
- Received the Semiconductor Research Corporation's 2001 *SRC Inventor's Recognition Award*
- Received the Semiconductor Research Corporation's 1994 *SRC Technical Excellence Award*
- Received the Semiconductor Research Corporation's 1988 *SRC Inventor's Recognition Award*

Litigation Support Experience

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Hsuanyeh
Case Name:	Vervain, LLC v. Phison Electronics Corp., No. 1-24-cv-259 (W.D.Tex.)#1 Complaint (2024 0308)
Services Provided:	Expert Declaration
Disposition:	Pending
Date:	Started April 2024

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Gish
Case Name:	Microchip Technology Inc. v. Aptiv Services US, LLC., C.A. No. 23-778 (JDW)
Services Provided:	Expert Declaration
Disposition:	Pending
Date:	Started September 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Baker Botts
Case Name:	Viasat v. Kioxia (6:21-cv-1231)
Services Provided:	Expert Declaration and pending Waco Trial
Disposition:	Pending
Date:	Started September 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Latham Watkins
Case Name:	Vervain v. Western Digital Civil Action No. 6:21-cv-00488-ADA
Services Provided:	Expert Reports for Western Digital; one deposition
Disposition:	settled
Date:	August 2021 to July 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Orrick, Herrington, and Sutcliffe
Case Name:	Godo Kaisha IP Bridge 1 v. Micron Technology, Inc., et al., Civil Action No. 6:20-cv-00178-ADA (W.D. Tex.)

Services Provided:	Two expert reports on behalf of Micron
Disposition:	Case dismissed
Date:	August 2020 – July 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Was Arnold & Porter, now Heim Payne Chorush as well as Debevoise and Gish
Case Name:	Taiwan Semiconductor Manufacturing Co. v. Silicon Storage Tech., Inc. AAA Case No. 01-21-0002-5445
Services Provided:	None yet
Disposition:	pending
Date:	Started July 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Was Arnold & Porter, then Perkins Coie
Case Name:	Intel v. BeSang
Services Provided:	Work on IPRs on behalf of Intel
Disposition:	My work completed July 2024
Date:	March 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Anova Law Group, PLLC
Case Name:	Chongqing HKC Optoelectronics Technology Co., Ltd. v. 138 East LCD Advancements Limited
Services Provided:	Two IPRs on behalf of the Petitioner (Chongqing)
Disposition:	pending
Date:	Started December 2022

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Perkins Coie
Case Name:	Intel v. BitMICRO
Services Provided:	Two IPR declarations on behalf of Intel
Disposition:	pending
Date:	Started October 2021

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Perkins Coie
Case Name:	Impinj v. NXP Civil Action NO. 6:21-cv-00530-ADA
Services Provided:	Expert Reports on behalf of Impinj and one trial testimony in Waco
Disposition:	Complete victory for Impinj; \$45 million settlement, plus \$15 million annually
Date:	October 2021 through November 7, 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Wilmer Cutler Pickering Hale and Dorr LLP (“WilmerHale”)

Case Name:	Civil Action No. 6:19-cv-404-ADA FLASH-CONTROL, LLC, Plaintiff, v. INTEL CORPORATION; IN THE UNITED STATES DISTRICT COURT FOR THE WESTERN DISTRICT OF TEXAS AUSTIN DIVISION
Services Provided:	Expert consultation on behalf of Intel
Disposition:	unkown
Date:	Started January 2020 ended 2022

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Susman Godfrey LLP
Case Name:	<i>Arigna Technology Limited v. Volkswagen AG et al.</i> , Case No. 2:21-cv-00054-JRG-RSP (E.D. Tex.)
Services Provided:	Expert consultation and testimony
Disposition:	Settled
Date:	Started 12/06/2020 ended June 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Susman Godfrey LLP
Case Name:	<i>Arigna Technology Limited v. Bayerische Motoren Werke AG et al.</i> , Case No. 2:21-cv-00172-JRG-RSP (E.D. Tex.)
Services Provided:	Expert consultation
Disposition:	Settled
Date:	Started 12/06/2020 ended June 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Susman Godfrey LLP
Case Name:	<i>Arigna Technology Limited v. Porsche AG, et al.</i> , Case No. 2:21-cv-00173-JRG-RSP (E.D. Tex.)
Services Provided:	Expert consultation
Disposition:	Settled
Date:	Started 12/06/2020 ended June 2023

Expert Engagement

Type of Matter:	Patent litigation
Law Firm:	Susman Godfrey LLP
Case Name:	<i>Arigna Technology Limited v. General Motors Company, et al.</i> , Case No. 2:21-cv-00174-JRG-RSP (E.D. Tex.)
Services Provided:	Expert consultation
Disposition:	Settled
Date:	Started 12/06/2020 ended June 2023

Expert Engagement

Type of Matter:	Patent litigation
-----------------	-------------------

Law Firm:	Susman Godfrey LLP
Case Name:	<i>Arigna Technology Limited v. Daimler AG, et al.</i> , Case No. 2:21-cv-00175-JRG-RSP (E.D. Tex.)
Services Provided:	Expert consultation
Disposition:	Settled
Date:	Started 12/06/2020 ended June 2023

Expert Engagement

Type of Matter:	ITC Investigation
Law Firm:	Susman Godfrey LLP
Case Name:	<i>In The Matter of Certain Power Inverters and Converters, Vehicles Containing the Same and Components Thereof</i> , Inv. No. 337-TA-1267 (ITC)
Services Provided:	Expert consultation and testimony
Disposition:	Settled
Date:	Started 12/06/2020 ended June 2023

Expert Engagement

Type of Matter:	IPR filing
Law Firm:	Kirkland & Ellis LLP
Case Name:	Case 1:19-cv-01006-UNA IPR PACT XPP Schweiz AG v. Intel Corporation IN THE UNITED STATES DISTRICT COURT FOR THE DISTRICT OF DELAWARE
Services Provided:	Expert consultation toward the filing of an IPR on behalf of Intel
Disposition:	Unknown
Date:	Started December 2019 ended 2022

Expert Engagement

Type of Matter:	IPR filing
Law Firm:	Wilmer Cutler Pickering Hale and Dorr LLP (WilmerHale)
Case Name:	VLSI Technology, LLC v. Intel Corporation, No. 3:17-cv-05671-BLF (N.D. Cal.), VLSI Technology, LLC v. Intel Corporation, No. 1:18-cv-00966-CFC-CJB (D. Del.) VLSI Technology, LLC v. Intel Corporation, No. 6:19-cv-00256-ADA (W.D. Tex.)
Services Provided:	Two declarations in support of an Intel IPR and two depositions
Disposition:	Unknown
Date:	Started May 1, 2019, ended July 30, 2020

Expert Engagement

Type of Matter:	Patent infringement and associated IPR filing
Law Firm:	Baker Botts (DC case) and Sterne, Kessler, Goldstein & Fox P.L.L.C. (IPR case)
Case Name:	Case No. 1:18-cv-555-LY UNITED STATES DISTRICT COURT WESTERN DISTRICT OF TEXAS AUSTIN DIVISION

	POLARIS INNOVATIONS, LTD., an Ireland limited company, Plaintiff, v. ADVANCED MICRO DEVICES, INC
Services Provided:	Filing a declaration in support of the IPR and planning for the district court case
Disposition:	Case dissolved
Date:	Started February 2019; ended May 2020

Expert Engagement

Type of Matter:	Trade secret case before an Arbitration Panel
Law Firm:	Anova Law Group
Case Name:	TVL International LLC v. Zhejiang Shenghui Lighting Co., Ltd. Case No. 01-17-0004-7802 AMERICAN ARBITRATION ASSOCIATION CHARLOTTE, NORTH CAROLINA <i>In the matter of arbitration between</i> TVL INTERNATIONAL LLC, Claimant, v . ZHEJIANG SHENGHUI LIGHTING CO., LTD. and SENGLED USA, INC.
Services Provided:	Three expert reports completed, and testimony before the Arbitration Panel
Disposition:	Ended with TVL declared the winner
Date:	April 9, 2018 through August 31, 2018.

Expert Engagement

Type of Matter:	Challenge of Patent Validity
Law Firm:	Weil, Gotshal and Manges; but in the future: Orrick, Herrington & Sutcliffe
Case Name:	North Star Innovations, Inc. v Micron Technology, Inc., et al., Case 17-cv-506-LPS-CJB, D. Del.
Services Provided:	Research in support of an IPR regarding Patent 5,943,274 on behalf of Micron
Disposition:	In progress but no work performed to date
Date:	March 5, 2018 through present

Expert Engagement

Type of Matter:	Challenge of Patent Validity
Law Firm:	Baker Botts
Case Name:	Advanced Micro Devices v. Broadcom
Services Provided:	One declaration in support of an IPR regarding Patent 7,472,151 Case No. Unassigned Patent 7,472,151 DECLARATION OF DR. CARL SECHEN IN SUPPORT OF PETITION FOR <i>INTER PARTES</i> REVIEW OF U.S. PATENT NO. 7,472,151

Disposition:	Settled not long after the IPR was filed
Date:	December 3, 2016 through March 27, 2017

Expert Engagement:

Type of Matter:	Trade Secret Misappropriation
Law Firm:	Wilson Sonsini Goodrich & Rosati
Case Name:	GSI Technology, Inc. v. United Memories, Inc. and Integrated Silicon Solution, Inc.; Case No. 5:13-cv-1081-PSG in the United States District Court for the Northern District of California, San Jose Division
Services Provided:	One declaration on behalf of ISSI, one deposition and one trial testimony before a jury
Disposition:	ISSI was victorious at trial and upon appeal
Date:	May 2014 through November 2015

Expert Engagement:

Type of Matter:	Patent validity – PTO re-examination proceedings
Law Firm:	Bruce Itchkawitz, Knobbe Martens Olson & Bear; Rich Kim, Mehran Arjomand, and David Kim, Morrison & Foerster LLP
Case Name:	Netlist, Inc. vs. Google, Inphi Corp., and Smart Modular Technologies <u>Individual cases in the last 5 years:</u> 1. <i>Inter Partes</i> Review No. IPR2017-00667 2. <i>Inter Partes</i> Review No. IPR2017-00549 3. <i>Inter Partes</i> Review No. IPR2017-00692 4. <i>Inter Partes</i> Review No. IPR2017-00562 5. <i>Inter Partes</i> Review No. IPR2017-00561 6. <i>Inter Partes</i> Review No. IPR2017-00587 7. <i>Inter Partes</i> Review No. IPR2017-00560 8. Appeal No. 2015-006849 9. Appeal No. 2015-007761 10. <i>Inter Partes</i> Review No. 2015-01021 11. Reexamination Control No.: 95/001,337 12. <i>Inter Partes</i> Review No. IPR2014-0097 13. <i>Inter Partes</i> Review No. IPR2014-00970 14. <i>Inter Partes</i> Review No. IPR2014-01011 15. <i>Inter Partes</i> Review No. IPR2014-00883 16. <i>Inter Partes</i> Review No. IPR2014-00883 17. Appeal No. 2013-009044
Services Provided:	25 declarations, 5 depositions on behalf of Netlist, Inc., filed July 2011 through May 2015 supporting patent validity challenges before the PTO Worked on 6 separate IPR's, each involved writing an expert report and giving a deposition.
Disposition:	Pending
Date:	June 28, 2011 through March 2018

Expert Engagement:

Type of Matter:	Patent invalidity and patent non-infringement
Law Firm:	Wilson Sonsini Goodrich & Rosati (representing the defendant)

Case Name:	HSM Portfolio, LLC, Technology Properties Limited, LLC v. Advanced Micro Devices (defendant) Case: 1:11-cv-00770-RGA (D. Del.)
Services Provided:	Work toward declaration on patent non-infringement
Disposition:	Settled in defendant's favor
Date:	April 2013 through December 2013

Expert Engagement:

Type of Matter:	Patent non-infringement
Law Firm:	Jones Day (representing the defendant)
Case Name:	Intellectual Ventures v. Xilinx (defendant) IN THE UNITED STATES DISTRICT COURT FOR THE DISTRICT OF DELAWARE INTELLECTUAL VENTURES I LLC and INTELLECTUAL VENTURES II LLC, Plaintiffs, v. ALTERA CORPORATION, MICROSEMI CORPORATION, LATTICE SEMICONDUCTOR CORPORATION and XILINX, INC., Defendants. C.A. No. 10-1065 (LPS) (CONSOLIDATED)
Services Provided:	One declaration on patent non-infringement; one deposition
Disposition:	Settled in defendants' favor
Date:	February 2013 through April 2014

Expert Engagement:

Type of Matter:	Patent non-infringement
Law Firm:	Morrison Foerster (representing the defendant)
Case Name:	Intellectual Ventures v. Altera (defendant) IN THE UNITED STATES DISTRICT COURT FOR THE DISTRICT OF DELAWARE INTELLECTUAL VENTURES I LLC and INTELLECTUAL VENTURES II LLC, Plaintiffs, v. ALTERA CORPORATION, MICROSEMI CORPORATION, LATTICE SEMICONDUCTOR CORPORATION and XILINX, INC., Defendants. C.A. No. 10-1065 (LPS) (CONSOLIDATED)
Services Provided:	One declaration on patent non-infringement; one deposition
Disposition:	Settled in defendant's favor
Date:	February 2013 through December 2013

Expert Engagement:

Type of Matter:	Patent validity and patent infringement
Law Firm:	Antonelli, Harrington & Thompson LLP
Case Name:	Guardian Media Technologies, Ltd., Plaintiff, v. Acer America Corp, et al., Defendants. <u>Cases:</u> CIVIL ACTION NO. 6:10-CV-597-LED Case No. 2:13-CV-08369 PSG (PLAx)
Services Provided:	One declaration on patent validity and two declarations on patent infringement
Disposition:	Settled in Plaintiff's favor
Date:	February 2013 through March 2015

Expert Engagement:

Type of Matter:	Patent infringement
Law Firm:	Winston & Strawn (Nanya)
Case Name:	On Technologies vs. Nanya
Services Provided:	Work on claim term definitions
Disposition:	Case was transferred from East. Dis. Texas to No. Calif. in late 2010; then settled
Date:	March 2010 through October 2010

Expert Engagement:

Type of Matter:	Patent infringement
Law Firm:	White and Case (Panasonic)
Case Name:	Freescall Semiconductor vs. Panasonic
Services Provided:	Invalidity and non-infringement expert reports filed November 2010; deposition December 2, 2010
Disposition:	ITC case settled January 2011
Date:	April 2010 through January 2011

Expert Engagement:

Type of Matter:	Patent infringement
Law Firm:	Weil, Gotshal and Manges (Samsung)
Case Name:	Rambus vs. Samsung
Services Provided:	One Declaration, One Deposition, and one trial testimony for Samsung <u>Declaration:</u> This report overviewed Rambus technology and patents, their alleged "steering" of patent claims, and Samsung's subsequent DDR products. <u>Deposition:</u> Dr. Sechen was asked to consider whether Samsung's subsequent DDR products, DDR2, DDR3, GDDR, GDDR2, GDDR3, and GDDR4 fell within the scope of their 2000 license agreement with Rambus. Representing attorney: Matt Antonelli. <u>Trial Testimony:</u>

	Dr. Sechen testified before Judge Ronald Whyte, San Jose, CA, that Samsung's subsequent DDR products, DDR2, DDR3, GDDR, GDDR2, GDDR3, and GDDR4 fell within the scope of their 2000 license agreement with Rambus. Representing attorney: Matt Antonelli.
Disposition:	Dr. Sechen's involvement completed on 30 September 2008 with the conclusion of the trial.
Date:	May 2008 through September 2008

Expert Engagement:

Type of Matter:	Patent infringement
Law Firm:	O'Melveny & Myers (Magma)
Case Name:	Synopsys vs. Magma Design Automation
Services Provided:	Two Declarations, One Supplemental Declaration and Two Depositions for Magma <u>Declaration:</u> June 10, 2005: Dr. Sechen illustrated that the material in the Synopsys patents (originally patented by Magma, but awarded to Synopsys by the court) were put into the public domain by Magma. In addition, he illustrated that the material in the Synopsys patents were generally in the public domain prior to the patents. <u>Declaration:</u> Dr. Sechen wrote a declaration supporting Magma's proposed claim constructions for the two patents involved on July 21, 2005. <u>Supplemental Declaration:</u> July 13, 2005 Dr. Sechen had a supplemental declaration showing that Synopsys' expert report in fact was consistent and in agreement with his first declaration. <u>Deposition:</u> Aug. 10, 2005: Dr. Sechen was deposed on the July 21, 2005 declaration. Representing attorney: Mark Miller. <u>Deposition:</u> Dec. 16, 2005: Dr. Sechen was deposed on the June 10, 2005 declaration and on the supplemental declaration of July 13, 2005. Representing attorney: Mark Miller.
Disposition:	Settled after court made several rulings in Magma's favor
Date:	Started January 2005; Ended January 2006

Expert Engagement:

Type of Matter:	Trade Secret Misappropriation and Patent Enforcement
Law Firm:	Susman Godfrey
Case Name:	UniRAM Technology vs. Monolithic System Technology and TSMC

Services Provided:	3 Expert Reports, 1 Declaration and 2 Depositions, and 2 trial testimonies for UniRAM. <u>Expert Report:</u> May 19, 2005: Expert report covered claim construction issues <u>Declaration:</u> May 19, 2005: Declaration in support of plaintiff UniRAM's response to TSMC's motion for summary judgment of inequitable conduct. (Subject of the 2nd trial testimony.) <u>Expert Report:</u> May 31, 2007: Expert report that concluded that the UniRAM trade secrets were not and are not generally known, and that the UniRAM trade secrets created significant value for the defendants. <u>Expert Rebuttal Report:</u> Expert rebuttal report filed on June 13, 2007. <u>Deposition:</u> June 21, 2005: Deposition concerning expert report of May 19, 2005. Representing attorney: Russ Chorush. <u>Deposition:</u> June 28, 2007: Deposition regarding expert report of May 31, 2007 and to a lesser degree the May 19, 2005 declaration. Representing attorney: Russ Chorush. <u>Trial Testimony:</u> September 2007 trial concerning Dr. Sechen's expert report of May 31, 2007. Representing attorneys: Russ Chorush (Heim, Payne and Chorush) and Joseph Grinstein (Susman Godfrey). <u>Trial Testimony:</u> January 2008 regarding my declaration of May 19, 2005. Representing attorneys: Russ Chorush (Heim, Payne and Chorush) and Joseph Grinstein (Susman Godfrey). The presiding judge specifically commented on Dr. Sechen's credibility in his final judgment.
Disposition:	\$32 million judgment for UniRAM
Date:	Started February 2005; Ended January 15, 2008

Expert Engagement:

Type of Matter:	Patent validity and infringement (both to PTO and trial)
Law Firm:	Squire, Sanders and Dempsey (Hitachi)
Case Name:	Translogic vs. Hitachi
	4 expert reports, 2 declarations, 1 deposition and trial testimony for Hitachi

Expert Disclosure Statement:

Expert disclosure statement of July 25, 2003.

Dr. Sechen was retained by Squire, Sanders & Dempsey LLP, counsel for defendants Hitachi, Ltd., Hitachi America, Ltd. and Renesas Technology America, Inc. (collectively "Hitachi") to compare claims of U.S. Patent No. 5,162,666 ("the '666 Patent") with certain technical references which were prior art to the '666 Patent. Plaintiff Translogic Technology, Inc. ("Translogic") asserted claims of the '666 Patent against Hitachi.

Supplemental Expert Disclosure:

Supplemental expert disclosure statement of September 19, 2003.

Dr. Sechen was retained by Squire, Sanders & Dempsey LLP, counsel for defendants Hitachi, Ltd., Hitachi America, Ltd. and Renesas Technology America, Inc. (collectively "Hitachi") to compare claims of U.S. Patent No. 5,162,666 with certain technical references which were prior art. Dr. Sechen supplemented the Expert Disclosure Statement he previously submitted in this case based on his review of the Report and Recommendation of Special Master Regarding Claim Construction.

Expert Disclosure Statement:

Expert disclosure statement of February 9, 2004.

Dr. Sechen was retained by Squire, Sanders & Dempsey LLP, counsel for defendants Hitachi, Ltd., Hitachi America, Ltd. and Renesas Technology America, Inc. (collectively "Hitachi") to analyze certain circuits in Hitachi microprocessors that Translogic Technology, Inc. ("TTI") accuses of infringing Patent No. 5,162,666 ("the '666 Patent"). Dr. Sechen was also asked to review aspects of the design and design process for these Hitachi products.

Supplemental and Restated Expert Disclosure Statement:

Supplemental and restated expert disclosure statement of September 20, 2004.

Dr. Sechen was retained by Squire, Sanders & Dempsey LLP, counsel for defendants Hitachi, Ltd., Hitachi America, Ltd. and Renesas Technology America, Inc. (collectively "Hitachi") to analyze certain circuits in Hitachi microprocessors that Translogic Technology, Inc. ("TTI") accuses of infringing Patent No. 5,162,666 ("the '666 Patent"). Dr. Sechen was also asked to review aspects of the design and design process for these Hitachi products. This report supplements my earlier expert disclosure statement, however, it is written as a substitute for that earlier report. This revision now uses the multiplexer drawing convention used by Hitachi, which is different from the standard U.S. convention.

Declaration in Opposition to Plaintiff's Motion for Summary Judgment:

Declaration in opposition to plaintiff's motion for summary judgment, April 2004. Dr. Sechen was asked to analyze certain circuits in Hitachi microprocessors that Translogic Technology accused of infringing their patent No. 5,162,666. Dr. Sechen was also asked to review aspects of the design and design process for these Hitachi products.

Second Declaration in Opposition to Plaintiff's Motion for Summary Judgment:

	Second declaration in opposition to plaintiff's motion for summary judgment, September 20, 2004. Dr. Sechen was asked to analyze certain circuits in Hitachi microprocessors that Translogic Technology accused of infringing their patent No. 5,162,666. Dr. Sechen was also asked to review aspects of the design and design process for these Hitachi products. <u>Deposition:</u> Aug. 18, 2003: Dr. Sechen was deposed on all of the above-mentioned expert disclosures and declarations. Representing attorneys: Nathan Lane and David Elkins of Squire, Sanders and Dempsey, as well as Alan Loudermilk of Loudermilk Associates. <u>Trial Testimony:</u> January 2004: Representing attorneys: Nathan Lane and David Elkins of Squire, Sanders and Dempsey, as well as Alan Loudermilk of Loudermilk Associates.
Disposition:	Concluded, PTO ruled in favor of Hitachi but court ruled in favor of Translogic.
Date:	July 2003 to September 2004

Expert Engagement:

Type of Matter:	Trade Secret Theft
Law Firm:	Keker and Van Ness (Cadence)
Case Name:	Cadence vs. Avanti
Services Provided:	Two Declarations and One Deposition for Cadence <u>Declaration:</u> Short declaration that verified and substantiated the findings of other Cadence experts. <u>Declaration:</u> Dr. Sechen submitted a very long reply declaration that compared an Avanti software product to a Cadence software product, and showing that the products were not in the public domain. <u>Deposition:</u> Deposition focused primarily on the reply declaration. Representing attorney: Michael Page.
Disposition:	Concluded
Date:	July 1995 to July 1996

Graduated Ph.D. Students

1. Kai-Win Lee (Yale May 1990) “Global Routing of Row-Based Integrated Circuits”.
2. Dahe Chen (Yale May 1992) “Mickey: A Graph-Based Macro-Cell Global Router”.
3. Mark Chiang (Yale May 1992) “A Perturbation Approach to the Symbolic Analysis of Analog Circuits”.
4. William Swartz (Yale May 1993) “Automatic Layout of Analog and Digital Mixed Macro/Standard Cell Integrated Circuits”.
5. Ted Stanion (Yale May 1994) “Boolean Algorithms for Combinational Synthesis and Test Generation”.
6. Kalapi Roy (University of Washington June 1994) “A Timing-Driven Multi-Way Partitioning System for Integrated Circuits and Multi-Chip Systems”.
7. Jer-Jaw Hsu (University of Washington December 1994) “Fully Symbolic Analysis of Large Analog Integrated Circuits”.
8. Wern-Jieh Sun (University of Washington December 1994) “Effective and Efficient Placement for Very Large Integrated Circuits”.
9. Qicheng Yu (University of Washington March 1995) “Approximate Symbolic Analysis of Large Analog Integrated Circuits”.
10. Bingzhong David Guan (University of Washington August 1996) “Automatic Layout Generation of Static CMOS Combinational Cells and Blocks”.
11. Eugene Liu (University of Washington, December 1997) “Global Routing and Pin Assignment for Multi-layer Chip-level Layout”.
12. Hsiao-Ping Tseng (University of Washington, December 1997) “Detailed Routing Algorithms for VLSI Circuits”.
13. Gin Yee (University of Washington, June 1999) “Dynamic Logic Design and Synthesis Using Clock-Delayed Domino”.
14. Tyler Thorp (University of Washington, December 1999), “Design and Synthesis of Dynamic Circuits”.
15. Tatjana Serdar (University of Washington, December 2000), “Automatic Datapath Tile Placement and Routing”.
16. Jovanka Ciric (University of Washington, August 2001), “Boolean Matching and Level-Based Technology Mapping”.
17. Yi Han (University of Washington, December 2004), “A High-Performance CMOS Programmable Logic Core for System-on-Chip Applications”.
18. Hiran Tennakoon (University of Washington, August 2005), “Efficient and Accurate Gate Sizing With Piecewise Convex Delay Models”.
19. Miodrag Vujkovic (University of Washington, March 2006), “Efficient Fully-Automated, Refinement-Based Power-Delay Optimization Design Flow for Standard Cell Designs”.
20. Kian Hour (Alfred) Chong (University of Washington, June 2006), “Self-Calibrating Differential Output Prediction Logic”.
21. Xinyu (Sunny) Guo (University of Washington, June 2006), “A High-Throughput Divider Based on Output Prediction Logic”.
22. Sheng Sun (University of Washington, August 2006), “High Performance and Energy Efficient Adder Design”.
23. Mohammad Rahman (UT-Dallas, December 2011), “Power and Leakage Minimization for Digital ICs”.
24. Chiu-Wei Pan (UT-Dallas, August 2012), “High Speed and Power Efficient Compression of Partial Products”.
25. Zhao Wang (UT-Dallas, October 2012), “Accurate Wire Endpoint Delay Estimation”.
26. Akshay Sridharan (UT-Dallas, October 2015), “STARK: Synchronous to Asynchronous Re-design Kit”.
27. Anitha Yella (UT-Dallas, January 2016), “Power Optimization in ICs”.

28. Meisam Roshan (UT-Dallas, September 2016), “A MEMS-Assisted Dual-Resonator Temperature-to-Digital Converter”.
29. Huihua (Helen) Huang (UT-Dallas, May 2018), “A 0.1ps Resolution Coarse-Fine Time-to-Digital Converter with 2.21ps Single-Shot Precision”.
30. Jingxiang (Amelie) Tian (UT-Dallas, December 2019), “Transistor-level Programmable Fabric”.
31. Bo Hu (UT-Dallas, March 2021), “Thermal-Aware Placement and High Level Synthesis For Hardware Security”.
32. Lubaba Nahar, (UT-Dallas, July 2022), “CAD Tools for PCB Reverse Engineering and IC Interconnect”.
33. Qiongdan (Olivia) Huang, (UT-Dallas, July 2022), “Toward Accurate Timing Analysis of Transistor-Level Programmable Fabric”.
34. Xiangyu Xu, (UT-Dallas, December 2022), “Routing Methods for Transistor-level Programmable Fabrics”.
35. Vibhav Kumarswami Salimath, (UT-Dallas, December 2023), “Standard Cell Library Composition Optimization for Advanced Process Nodes”.
36. Apurva Jain, (UT-Dallas, January 2024), “Addressing Testing and Verification Challenges in a Transistor-level Programmable Fabric”.

Current Ph.D. Students

1. Thomas Broadfoot, Ph.D. expected 12/24
2. Jinkun Feng, Ph.D. expected 06/26
3. Nazia Bushra, Ph.D. expected 06/27

Teaching Activities

<i>Yr</i>	<i>Qtr</i>	<i>Course</i>	<i>Brief Title, Credits, #students</i>	<i>Stud. Opin. Survey</i>
1992	Aut	EE538	Auto Layout, 4 credits, 20 students	4.38/4.08
1993	Win	EE356	Analog ICs, 4 credits, 55 students	3.21/3.00
1993	Spr	EE535	VLSI Design, 4 credits, 25 students	4.27/4.47
1993	Sum	EE332	Analog ICs, 5 credits, 35 students	3.57/3.48
1993	Aut	EE538	Auto Layout, 4 credits, 15 students	3.60/3.50
1993	Aut	EE332	Analog ICs, 5 credits, 40 students	3.93/3.90
1994	Win	EE433	Analog MOS ICs, 4 credits, 65 students	3.96/4.08
1994	Spr	EE476	Digital ICs, 5 credits, 55 students	3.89/3.89
1994	Aut	EE538	Auto Layout, 4 credits, 15 students	3.63/3.63
1994	Aut	EE433	Analog MOS ICs, 4 credits, 55 students	4.32/4.30
1995	Win	EE476	Digital ICs, 5 credits, 85 students	3.75/3.85
1995	Spr	EE473	Adv. Analog MOS, 5 credits, 45 students	3.53/3.59
1995	Aut	EE476	Digital ICs, 5 credits, 85 students	4.40/4.20
1996	Win	EE535	VLSI Design, 4 credits, 55 students	4.10/4.00
1996	Spr	EE538	Auto Layout, 4 credits, 15 students	4.30/4.50
1996	Aut	EE476	Digital ICs, 5 credits, 125 students	3.40/3.40

1997	Win	EE477	Custom Dig ICs, 4 credits, 45 students	4.30/4.10
1997	Win	EE541	Auto Layout, 4 credits, 20 students	4.70/4.70
1997	Spr	EE535	VLSI Design, 4 credits, 20 students	3.83/4.00
1997	Aut	EE476	Digital ICs, 5 credits, 60 students	3.88/3.89
1998	Win	EE477	Custom Dig ICs, 4 credits, 60 students	3.66/3.61
1998	Spr	EE476	Digital ICs, 5 credits, 60 students	3.39/3.50
1998	Spr	EE535	Digital VLSI Design, 4 credits, 35 stud.	3.57/3.94
1998	Aut	EE476	Digital ICs, 5 credits, 35 students	4.15/4.50
1999	Win	EE477	Custom Dig ICs, 4 credits, 55 students	4.00/3.94
1999	Spr	EE535	Digital VLSI Design, 4 credits, 35 stud.	3.79/4.13
1999	Aut	EE476	Digital ICs, 5 credits, 50 students	4.70/4.70
2000	Win	EE477	Custom Dig ICs, 4 credits, 55 students	4.53/4.27
2000	Spr	EE535	Digital VLSI Design, 4 credits, 35 stud.	4.35/4.44
2000	Aut	EE476	Digital ICs, 5 credits, 125 students	4.20/4.19
2001	Win	EE477	VLSI II, 5 credits, 75 students	4.30/4.37
2001	Win	EE525	VLSI II, 5 credits, 25 students	4.25/4.50
2001	Spr	EE526	VLSI III, 4 credits, 50 students	4.35/4.50
2001	Aut	EE476	VLSI I, 5 credits, 150 students	4.1/4.2
2002	Win	EE477	VLSI II, 5 credits, 75 students	4.1/4.1
2002	Win	EE525	VLSI II, 5 credits, 25 students	4.1/4.1
2002	Spr	EE526	VLSI III, 4 credits, 50 students	3.6/3.3
2002	Aut	EE476	VLSI I, 5 credits, 150 students	3.7/3.6
2003	Win	EE477	VLSI II, 5 credits, 55 students	3.4/3.8
2003	Win	EE525	VLSI II, 5 credits, 25 students	3.6/3.3
2003	Spr	EE526	VLSI III, 4 credits, 32 students	3.3/3.5
2003	Aut	EE476	VLSI I, 5 credits, 30 students	4.5/4.6
2004	Win	EE477	VLSI II, 5 credits, 12 students	3.2/3.7
2004	Win	EE525	VLSI II, 5 credits, 7 students	2.2/2.2
2004	Spr	EE526	VLSI III, 4 credits, 12 students	3.2/3.2
2004	Aut	EE476	VLSI I, 5 credits, 65 students	3.8/3.9
2005	Fall	EE6325	VLSI Design, 3 units, 42 students	4.4/4.6
2006	Spr	EE7325	Advanced VLSI Design, 3 units, 20 students	4.5/4.3
2006	Fall	EE6325	VLSI Design, 3 units, 66 students	4.4/4.2
2007	Spr	EE7325	Advanced VLSI Design, 3 units, 15 students	4.5/4.2
2007	Fall	EE6325	VLSI Design, 3 units, 40 students	4.8/4.6
2008	Spr	EE4325	Introduction to VLSI Design, 40 students	3.5/3.4
2008	Spr	EE7325	Advanced VLSI Design, 3 units, 19 students	4.6/4.4
2008	Sum	EE6325	VLSI Design, 3 units, 36 students	4.0/3.7
2008	Fall	EE6325	VLSI Design, 3 units, 40 students	4.1/4.1
2009	Spr	EE4325	Introduction to VLSI Design, 22 students	4.2/3.9
2009	Spr	EE7325	Advanced VLSI Design, 3 units, 37 students	3.7/3.6
2009	Sum	EE6325	VLSI Design, 3 units, 17 students	4.1/4.0
2009	Fall	EE6325	VLSI Design, 3 units, 67 students	4.3/4.0
2010	Spr	EE4325	Introduction to VLSI Design, 25 students	4.6/4.5
2010	Spr	EE7325	Advanced VLSI Design, 3 units, 35 students	4.4/4.4
2010	Sum	EE6325	VLSI Design, 3 units, 17 students	4.7/4.5
2010	Sum	EE3320	Digital Circuits, 3 units, 17 students	4.1/3.8
2010	Fall	EE6325	VLSI Design, 3 units, 77 students	4.4/4.2
2011	Spr	EE4325	Introduction to VLSI Design, 37 students	4.0/4.1
2011	Spr	EE7325	Advanced VLSI Design, 3 units, 17 students	4.2/4.2
2011	Sum	EE6325	VLSI Design, 3 units, 16 students	4.14

2011	Sum	EE3320	Digital Circuits, 3 units, 26 students	4.13
2011	Fall	EE6325	VLSI Design, 3 units, 91 students	4.73
2012	Spr	EE4325	Introduction to VLSI Design, 27 students	4.25
2012	Spr	EE7325	Advanced VLSI Design, 3 units, 17 students	4.90
2012	Sum	EE6325	VLSI Design, 3 units, 35 students	4.46
2012	Sum	EE3311	Electronic Circuits, 3 units, 22 students	3.62
2012	Fall	EE6325	VLSI Design, 3 units, 144 students	4.76
2013	Spr	EE6325	VLSI Design, 3 units, 41 students	4.85
2013	Spr	EE4325	Introduction to VLSI Design, 7 students	4.75
2013	Sum	EE7325	Advanced VLSI Design, 3 units, 67 students	4.38
2013	Sum	EE3311	Electronic Circuits, 3 units, 46 students	4.58
2013	Fall	EE6325	VLSI Design, 3 units, 129 students	4.64
2014	Spr	EE4325	Introduction to VLSI Design, 25 students	4.69
2014	Sum	EE7325	Advanced VLSI Design, 3 units, 74 students	4.53
2014	Sum	EE3311	Electronic Circuits, 3 units, 40 students	4.75
2014	Fall	EE6325	VLSI Design, 3 units, 139 students	4.79
2015	Spr	EE4325	Introduction to VLSI Design, 25 students	4.72
2015	Sum	EE7325	Advanced VLSI Design, 3 units, 49 students	4.25
2015	Sum	EE3311	Electronic Circuits, 3 units, 25 students	4.80
2015	Fall	EE6325	VLSI Design, 3 units, 142 students	4.73
2016	Spr	EE4325	Introduction to VLSI Design, 25 students	3.67
2016	Sum	EE7325	Advanced VLSI Design, 3 units, 70 students	4.77
2016	Sum	EE3311	Electronic Circuits, 3 units, 35 students	4.88
2016	Fall	EE6325	VLSI Design, 3 units, 140 students	4.79
2017	Spr	EE4325	Introduction to VLSI Design, 32 students	4.10
2017	Sum	EE7325	Advanced VLSI Design, 3 units, 30 students	4.88
2017	Sum	EE3311	Electronic Circuits, 3 units, 32 students	4.50
2017	Fall	EE6325	VLSI Design, 3 units, 80 students	4.82
2018	Spr	EE4325	Introduction to VLSI Design, 32 students	4.63
2018	Sum	EE7325	Advanced VLSI Design, 3 units, 30 students	4.88
2018	Fall	EE6325	VLSI Design, 3 units, 80 students	4.88
2019	Spr	EE4325	Introduction to VLSI Design, 43 students	4.33
2019	Sum	EE3311	Electronic Circuits, 3 units, 61 students	4.23
2019	Fall	EE6325	VLSI Design, 3 units, 60 students	4.84
2020	Spr	EE4325	Introduction to VLSI Design, 33 students	4.75
2020	Sum	EE3311	Electronic Circuits, 3 units, 29 students	4.30
2020	Fall	EE6325	VLSI Design, 3 units, 43 students	4.65
2020	Fall	EE7325	Advanced VLSI Design, 3 units, 16 students	4.17
2021	Spr	EE4325	Introduction to VLSI Design, 41 students	3.40
2021	Sum	EE3311	Electronic Circuits, 3 units, 29 students	4.20
2021	Fall	EE6325	VLSI Design, 3 units, 72 students	4.75
2022	Spr	EE7325	Advanced VLSI Design, 3 units, 17 students	4.17
2022	Spr	EE4325	Introduction to VLSI Design, 46 students	4.13
2022	Sum	EE3311	Electronic Circuits, 3 units, 29 students	-----
2022	Fall	EE6325	VLSI Design, 3 units, 111 students	4.20
2023	Spr	EE4325	Introduction to VLSI Design, 48 students	4.00
2023	Fall	EE6325	VLSI Design, 3 units, 76 students	4.00

Course Development

U. of Washington: EE 541 Automatic Layout of Integrated Circuits (first offering: Fall 1992)
U. of Washington: EE 477 Custom Digital CMOS Circuit Design (first offering: Win 1997)
U. of Washington: EE 476 Digital Integrated Circuit Design (first offering: Spr 1994)

Yale University: EE 880 Automatic Layout of Integrated Circuits.
Yale University: EE 877 Introduction to VLSI CAD Tools
Yale University: EE 878 Introduction to VLSI Synthesis
Yale University: EE 988 Analog Integrated Circuits II

Journal Publications

1. A. Jain, T. Broadfoot, Y. Makris and C. Sechen, "Modeling Bidirectional Switches for Enabling Logic Equivalence Checking in a Transistor-Level Programmable Fabric," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*,
2. A. Fowler, S. Mohammed, M. Shihab, T. Broadfoot, P. Beerel, C. Sechen, and Y. Makris, "A TRAP for SAT: On the Imperviousness of a Transistor-Level Programmable Fabric to Satisfiability Attacks", *Transactions on Cryptographic Hardware and Embedded Systems*, 2025.
3. Roshan, C. Sechen, *et al.*, "A MEMS-Assisted Temperature Sensor with 20 μ K Resolution, Conversion Rate of 200S/s and FOM of 0.04pJK", *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 185-197, January 2017.
4. Z. Wang, X. He, and C. Sechen, "A New Approach for Gate-Level Delay-Insensitive Asynchronous Logic", *Journal of Circuits, Systems & Signal Processing*, October 2014.
5. Z. Wang, C. Pan, Y. Song, and C. Sechen, "High-Throughput Digital IIR Filter Design", *Journal of Algorithms and Optimization (JAO)*, Vol. 2, No. 2, pp. 15-27, April 2014.
6. C. Pan, Z. Wang and C. Sechen, "High Speed and Power Efficient Compression of Partial Products and Vectors," *Journal of Algorithms and Optimization (JAO)*, Vol. 1, No. 1, pp. 39-54, October 2013.
7. Z. Wang and C. Sechen, "A New Algorithm for Accurate Wire Endpoint Delay Estimation", *Journal of Algorithms and Optimization (JAO)*, Vol. 2, No. 1, pp. 30-43, Jan. 2014.
8. M. Rahman, H. Tennakoon, and C. Sechen, "Library-Based Cell-Size Selection using Extended Logical Effort", *IEEE Trans. on Computer-Aided Design*, July 2013.
9. H. Tennakoon and C. Sechen, "Non-convex Gate Delay Modeling and Delay Optimization," *IEEE Trans. on Computer-Aided Design*, Vol. 27, No. 9, September 2008, pp. 1583-1594.
10. Don Bouldin, Warren Snapp, Paul Haug, Dave Sunderland, Roger Brees, Carl Sechen, and Wayne Dai, "Automated Design of Digital Signal Processing ASICs," *IEEE Circuits and Devices*, vol. 20, n. 4, pp. 17-21, July 2004.
11. J. Ciric and C. Sechen, "Efficient Canonical Form for Boolean Matching of Complex Functions in Large Libraries," *IEEE Trans. on Computer-Aided Design*, Vol. 22, No. 5, May 2003, pp. 535-544.
12. T. Thorp and C. Sechen, "Design and Synthesis of Dynamic Circuits," *IEEE Transactions on VLSI Systems*, Vol. 11, No. 1, February 2003, pp. 141-149.
13. G. Hoyer, G. Yee and C. Sechen, "Locally-Clocked Pipelines and Dynamic Logic," *IEEE Transactions on VLSI Systems*, Vol. 10, No. 1, February 2002, pp. 58-62.

14. H. P. Tseng, L. Scheffer and C. Sechen, "Timing- and Crosstalk-Driven Area Routing," *IEEE Transactions on Computer Aided Design*, vol. 20, no. 4, pp. 528-544, April 2001.
15. G. Yee and C. Sechen, "Clock-Delayed Domino for Dynamic Circuit Design," *IEEE Transactions on VLSI Systems*, Vol. 8, No. 4, August 2000, pp. 425-431.
16. L. Liu and C. Sechen, "Multi-layer Chip-level Global Routing Using an Efficient Graph-based Steiner Tree Heuristic," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 10, October 1999.
17. L. Liu and C. Sechen, "Multi-layer Pin Assignment for Macro Cell Circuits", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 10, October 1999.
18. H. P. Tseng and C. Sechen, "A Gridless Multi-Layer Router for Standard Cell Circuits using CTM Cells," *IEEE Transactions on Computer Aided Design*, Vol. 18, No. 10, October 1999.
19. W. Sun and C. Sechen, "A Parallel Standard Cell Placement Algorithm," *IEEE Transactions on Computer Aided Design*, vol. 16, no. 11, November 1997, pp. 1342-1357.
20. Q. Yu and C. Sechen, "Efficient Approximation of Symbolic Network Functions Using Matroid Intersection Algorithms," *IEEE Transactions on Computer Aided Design*, vol. 16, no. 10, October 1997, pp. 1073-1081.
21. Q. Yu and C. Sechen, "Generation of Color-Constrained Spanning Trees with Application in Symbolic Analysis," *Int. Journal of Circuit Theory and Applications*, Vol. 24, No. 5, pp. 597-603, 1996.
22. Q. Yu and C. Sechen, "A Unified Approach to the Approximate Symbolic Analysis of Large Analog Integrated Circuits," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 43, No. 8, pp. 656-669, August 1996.
23. K. Roy and C. Sechen, "A Timing-Driven Partitioning System for Multiple FPGAs," *Int. Journal of VLSI Design*, Vol. 4, No. 4, pp. 309-328, 1996.
24. K. Roy, D. Guan and C. Sechen, "A Sea-of-Gates Style FPGA Placement Algorithm," *Int. Journal of VLSI Design*, Vol. 4, No. 4, pp. 293-307, 1996.
25. T. Stanion, D. Bhattacharya, and C. Sechen, "An Efficient Method for Generating Exhaustive Test Sets," *IEEE Transactions on Computer-Aided Design*, Vol. 14, No. 12, December 1995.
26. W. Sun and C. Sechen, "Efficient and Effective Placement for Very Large Circuits," *IEEE Transactions on Computer-Aided Design*, Vol. 14, No. 3, pp. 349-359, March 1995.
27. J. J. Hsu and C. Sechen, "DC Small Signal Symbolic Analysis of Large Analog Integrated Circuits," *IEEE Transactions on Circuits and Systems I*, Vol. 41, No. 12, pp. 817-828, December 1994.
28. T. Stanion and C. Sechen, "Boolean Division and Factorization Using Binary Decision Diagrams," *IEEE Transactions on Computer-Aided Design*, vol. 13, no. 9, September 1994, pp. 1179-1184.
29. C. Sechen, "Chip-Planning, Placement, and Global Routing of Macro Cell Integrated Circuits Using Simulated Annealing," *Int. J. Computer-Aided VLSI Design*, vol. 2, no. 2, 1990, pp. 127-158.
30. C. Sechen, D. Braun, and A. Sangiovanni-Vincentelli, "ThunderBird: A Complete Standard Cell Layout Package," *IEEE J. of Solid State Circuits*, vol. 23, n. 2, pp. 410- 420, April 1988.
31. C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf Placement and Routing Package," *IEEE J. of Solid State Circuits*, vol. 20, n. 2, April 1985, pp. 510-522.
32. C. Sechen, "An Improved Lock Layer Transistor," *Solid State Electronics*, Vol. 21, No. 6, June 1978, pp. 911-913.

33. S. Senturia and C. Sechen, "The Use of the Charge-Flow Transistor to Distinguish Surface and Bulk Components of Thin-Film Sheet Resistance," *IEEE Transactions on Electron Devices*, vol.24, no.9, p.1207, September 1977.
34. S. Senturia, C. Sechen, and J. Wishneusky, "The Charge-Flow Transistor: A New MOS Device," *Applied Physics Letters*, Vol. 30, No. 2, Jan. 1977, p. 106-108.

Fully Refereed Conference Proceedings

1. A. Jain, T. Broadfoot, Y. Makris, and C. Sechen, "Testing a Transistor-Level Programmable Fabric: Challenges and Solutions", *Proc. IEEE VLSI Test Symposium (VTS)*, Tempe, AZ, April 22-24, 2024.
2. A. Jain, T. Broadfoot, Y. Makris, and C. Sechen, "Quo Vadis Signal? Automated Directionality Extraction for Post-Programming Verification of a Transistor-Level Programmable Fabric", *Proc. Design Automation and Test in Europe (DATE) Conference*, Antwerp, Belgium, April 17-19, 2023.
3. V. Salimath and C. Sechen, "Essential Standard Cell Library Composition", *Proc. 15th IEEE Dallas Circuits and Systems Conference (DCAS 2022)*, June 17-19, 2022, Dallas, TX.
4. L. Nahar, J. Rajendran, Y. Makris and C. Sechen, "MTBoM: Metal Trace to Bill of Materials Generation for PCB Reverse Engineering", *Proc. 15th IEEE Dallas Circuits and Systems Conference (DCAS 2022)*, June 17-19, 2022, Dallas, TX.
5. Q. Huang, J. Tian, T. Broadfoot, X. Xu, B. Hu, M. Shihab, A. Jain, V. Salimath, Y. Makris and C. Sechen", *Proc. 15th IEEE Dallas Circuits and Systems Conference (DCAS 2022)*, June 17-19, 2022, Dallas, TX.
6. M. Shihab, J. Tian, G. Reddy, B. Hu, W. Swartz Jr., B. Schaefer, C. Sechen and Y. Makris, "CASPER: CAD Framework for a Novel Transistor-Level Programmable Fabric", *Proc. IEEE International Symposium on Circuits & Systems (ISCAS)*, May 17-20, 2020, Seville, Spain.
7. V. Salimath and C. Sechen, "Optimal Standard Cell Library Composition for 7nm", *Proc. IEEE International Symposium on Circuits & Systems (ISCAS)*, May 17-20, 2020, Seville, Spain.
8. M. Shihab, B. Ramanidharan, S. Tellakula, G. Reddy, J. Tian, C. Sechen, and Y. Makris, "ATTEST: Application-Agnostic Testing of a Novel Transistor-Level Programmable Fabric", *Proc. 8th IEEE VLSI Test Symposium (VTS'20)*, April 5-8, 2020, San Diego, CA.
9. M. Shihab, J. Tian, G. Reddy, B. Hu, W. Swartz Jr., B. Schaefer, C. Sechen and Y. Makris, "A Transistor-Level Fabric for Design Obfuscation", *Proc. Government Microcircuit Applications and Critical Technology Conference (GOMACTech)*, March 16-19, 2020, San Diego, CA.
10. B. Hu, M. Shihab, Y. Makris, B. Schaefer and C. Sechen, "An Efficient MILP-Based Aging-Aware Floorplanner for Multi-Context Coarse-Grained Runtime Reconfigurable FPGAs", *Proc. Design Automation in Europe Conference (DATE)*, March 9-13, 2020, Grenoble, France.
11. B. Hu, M. Shihab, Y. Makris, B. Schaefer and C. Sechen, "Extending the Lifetime of Coarse-grained Runtime Reconfigurable FPGAs by Balancing Processing Element Usage", *2019 International Conference on Field-Programmable Technology (FPT'19)*, December 9-13, 2019 Tianjin, China.
12. S. Reda, M. Saligane, S. S. Sapatnekar, C. Sechen, *et al.*, "Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project", *Proc. ACM/IEEE Design Automation Conference (DAC)*, June 4-6, 2019, Las Vegas, NV.

13. B. Hu, J. Tian, M. Shihab, G. Reddy, W. Swartz, Y. Makris, B. Schaefer, and C. Sechen, "Functional Obfuscation of Hardware Accelerators through Selective Partial Design Extraction onto an Embedded FPGA", *Proc. Great Lakes Symposium on VLSI (GLSVLSI)*, Washington, D.C., USA, May 9-11, 2019.
14. S. Sapatnekar, L. Saul, C. Sechen, *et al.*, "OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain", *Proc. Government Microcircuit Applications and Critical Technology Conference (GOMACTech)*, March 25-28, 2019, Albuquerque, NM, pp. 1105-1110.
15. M. Shihab, G. Reddy, J. Tian, B. Hu, W. Swartz, B. Carrion Schaefer, C. Sechen, Y. Makris, "Design Obfuscation through Selective Post-Fabrication Transistor-Level Programming," *Proc. of Design Automation and Test in Europe (DATE) Conference*, March 25-29, 2019, Florence, Italy.
16. A. Yella, S. Gunturi and C. Sechen, "Are Standalone Gate Size and VT Optimization Tools Useful?", *Proc. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Devices, Circuits, and Systems Track*, Windsor, ON, Canada, April 30 - May 3, 2017.
17. T. Broadfoot, C. Sechen and J. Rajendran, "On Designing Optimal Camouflaged Layouts", *Proc. IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, May 1-4, 2017, McLean, VA.
18. A. Yella and C. Sechen, "Improved Lagrangian Relaxation-based Gate Size and VT Assignment for Very Large Circuits, *Proc. IEEE PhD Research In Microelectronics and Electronics Conference Latin America (PRIME)*, Bariloche, Argentina, February 20-23, 2017. **BEST PAPER AWARD**
19. J. Tian, G. Reddy, J. Wang, W. Swartz, Y. Makris and C Sechen, "A Field Programmable Transistor Array Featuring Single-Cycle Partial/Full Dynamic Reconfiguration", *Proc. of Design Automation and Test in Europe (DATE) Conference*, March 27-31, 2017, Lausanne, Switzerland. **NOMINATED FOR BEST PAPER AWARD**
20. M. Roshan, C. Sechen, *et al.*, "Dual-MEMS Resonator Temperature-to-Digital Converter with 39 μ K Resolution and a FoM of 0.11pJK²", *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2016, San Francisco, CA.
21. M. Jain and C. Sechen, "Chip Design: Matrix Multiplier Based on Systolic Architecture", *IEEE Computer Society 6th International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, July 13-15, 2015, Dallas, TX.
22. Z. Wang, X. He and C. Sechen, "TonyChopper: A Desynchronization Package", *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov. 2-6, 2014, San Jose, CA. (20% accepted)
23. H. Huang and C. Sechen, "A 14-b, 0.1ps Resolution Coarse-Fine Time-to-Digital Converter in 45 nm CMOS", *2014 Dallas Circuits and Systems Conference (DCAS)*, Oct 12-13, 2014, Dallas, TX.
24. A. Sridharan and C. Sechen, "Minimum Energy Operation using Robust Asynchronous Logic with Sleep Transistors", *Proc. Dallas Circuits and Systems Conference (DCAS)*, Oct 12-13, 2014, Dallas, TX.
25. Yashaswini Prathivadi, Carl Sechen, and Roozbeh Jafari, "A Swarm of Wearable Sensors at the Edge of the Cloud for Robust Activity Recognition", *Proc. First International Workshop on the Swarm at the Edge of the Cloud (SEC'13)*, Sept. 29, 2013, Montreal, Canada.
26. Mohammad-Mahdi Bidmeshki, Carl Sechen, and Roozbeh Jafari, "Low Power Programmable Architecture for Periodic Activity Monitoring", *Proc. SRC Techcon*, 2013, Austin, TX.
27. Thomas Broadfoot, Jingxiang Tian, HeeEun Choi, Mohammad-Mahdi Bidmeshki, Roozbeh Jafari, and Carl Sechen, "Platform Design for Swarm Wearable Computing", *Proc. First International Workshop on the Swarm at the Edge of the Cloud (SEC'13)*, Sept. 29, 2013, Montreal, Canada.

28. A. Sridharan, C. Sechen, and R. Jafari, "Low-Voltage Low-Overhead Asynchronous Logic", *Int. Symp. on Low Power Electronics and Design (ISLPED)*, Beijing, China, Sept. 4-6, 2013. (25% accepted)
29. M. M. Bidmeshki, C. Sechen and R. Jafari, "Low Power Programmable Architecture for Periodic Activity Monitoring," *IEEE Texas Workshop on Integrated System Exploration (TexasWISE)*, 8 March 2013, Round Top, Texas.
30. Hee-Eun Choi, C. Sechen and R. Jafari, "24kb, 11T Subthreshold SRAM for Ultra-Low Power Medical and Health Monitoring Embedded Systems," *IEEE Texas Workshop on Integrated System Exploration (TexasWISE)*, 8 March 2013, Round Top, Texas.
31. M. Rahman, and C. Sechen, "Post-Synthesis Leakage Power Minimization", *Design Automation and Test in Europe (DATE) Conference*, March 12-16, 2012, Dresden, Germany.
32. M. Rahman, H. Tennakoon, R. Afonso and C. Sechen, "Power Reduction via Separate Synthesis and Physical Libraries", *Proc. Design Automation Conference (DAC)*, June 2011, San Diego, CA.
33. C. Pan and C. Sechen, "Power Efficient Partial Product Compression", *Proc. Great Lakes Symposium on VLSI*, Lausanne, May 2-4, 2011, Switzerland.
34. M. Rahman, H. Tennakoon, and C. Sechen, "Power Reduction via Near-Optimal Library-Based Cell-Size Selection", *Design Automation and Test in Europe (DATE) Conference*, March 14-18, 2011, Grenoble, France.
35. M. Rahman, H. Tennakoon, R. Afonso, and C. Sechen, October 17-18, "Design Automation Tools and Libraries for Low Power Digital Design", *Proc. IEEE Dallas Circuits and Systems Conference (DCAS)*, Oct. 17-18, 2010, Richardson, TX.
36. M. Rahman, H. Tennakoon, and C. Sechen, "Near Optimal Power Efficiency Through Gate Sizing", *Proc. Austin Conf. on Integrated Circuits and Systems (ACISC)*, Oct. 26-27, 2009, Austin, TX.
37. H. Huang and C. Sechen, "A 22mW 227Msps 11b Self-Tuning ADC Based on Time-to-Digital Conversion", *Proc. Austin Conf. on Integrated Circuits and Systems (ACISC)*, Oct. 26-27, 2009, Austin, TX.
38. E. Kiefer, W. Swartz, and C. Sechen, "Low Skew Automated Clock Tree Generation", *Proc. Austin Conf. on Integrated Circuits and Systems (ACISC)*, Oct. 26-27, 2009, Austin, TX.
39. R. Afonso, M. Rahman, H. Tennakoon, and C. Sechen, "Power Efficient Standard Cell Library Design", *Proc. IEEE Dallas Circuits and Systems Society Workshop*, Oct. 4-5, 2009, Dallas, TX.
40. C.W. Pan, Y. Song, Z. Wang, and C. Sechen, "DSP Power Reduction through Generalized Carry-Save Arithmetic", *Proc. IEEE Dallas Circuits and Systems Society Workshop*, Oct. 4-5, 2009, Dallas, TX.
41. R. Afonso, M. Rahman, H. Tennakoon, and C. Sechen, "Power Efficient Standard Cell Library Design", *Proc. Austin Conf. on Integrated Circuits and Systems (ACISC)*, Oct. 26-27, 2009, Austin, TX.
42. C.W. Pan, Y. Song, Z. Wang, and C. Sechen, "DSP Power Reduction through Generalized Carry-Save Arithmetic", *Proc. Austin Conf. on Integrated Circuits and Systems (ACISC)*, Oct. 26-27, 2009, Austin, TX.
43. E. Kiefer, W. Swartz, and C. Sechen, "Low Power Automated Clock Tree Generation", *Proc. IEEE Dallas Circuits and Systems Society Workshop*, Oct. 4-5, 2009, Dallas, TX.
44. H. Huang and C. Sechen, "A 22mW 227Msps 11b Self-Tuning ADC Based on Time-to-Digital Conversion", *Proc. IEEE Dallas Circuits and Systems Society Workshop*, Oct. 4-5, 2009, Dallas, TX.
45. S. Sun and C. Sechen, "Post-Layout Comparison of High Performance 64b Static Adders in Energy-Delay Space," *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, Lake Tahoe, CA, October 2007.

46. M. Rahman, H. Tennakoon, and C. Sechen, "Optimal Area-versus-Delay Circuit Sizing Using Extended Logical Effort", *Proc. Austin Conf. on Integrated Circuits and Systems (ACISC)*, May 14-15, 2007, Austin, TX.
47. K. H. Chong, L. McMurchie and C. Sechen, "A 64b Adder Using Self-calibrating Differential Output Prediction Logic," *Proc. Int. Solid-State Circuits Conference (ISSCC)*, February 2006, San Francisco, CA.
48. J. Zhang, M. Vujkovic, D. Wadkins, and C. Sechen, "Post-Layout Energy-Delay Analysis of Parallel Multipliers," *Proc. Int. Symp. on Circuits and Systems (ISCAS)*, May 2006, Greece.
49. M. Vujkovic, D. Wadkins and C. Sechen, "Efficient Post-Layout Power-Delay Curve Generation," *Prof. 15th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 21-23 September 2005, Leuven, Belgium. In: *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*, Series: Lecture Notes in Computer Science, Vol. 3728 Paliouras, Vassilis; Vounckx, Johan; Verkest, Diederik (Eds.) 2005, XV, 753 p., Softcover ISBN: 3-540-29013-3.
50. X. Guo and C. Sechen, "High Throughput Divider Using Output Prediction Logic," *Proc. PRIME (Ph.D. Research in Microelectronics) Conf.*, July 25-28, 2005, Lausanne, Switzerland.
51. X. Guo and C. Sechen, "A High Throughput Divider Implementation," *Proc. Custom Integrated Circuits Conference (CICC)*, 18-21 September 2005, San Jose, CA.
52. H. Tennakoon and C. Sechen, "Efficient and Accurate Gate Sizing with Piecewise Convex Delay Models," *Proc. Design Automation Conference (DAC)*, Anaheim, CA, June 13-17, 2005.
53. S. Sun, Y. Han, X. Guo, K. Chong, L. McMurchie and C. Sechen, "409ps 4.7 FO4 64b Adder Based on Output Prediction Logic in 0.18um CMOS," *Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, May 11-12, 2005, Tampa, FL.
54. X. Guo and C. Sechen, "High Speed Redundant Adder and Divider in Output Prediction Logic," *Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, May 11-12, 2005, Tampa, FL.
55. J. Kim, L. McMurchie and C. Sechen, "Mitigation of Single- and Multiple-Cycle-Duration SETs using Double-Mode Redundancy (DMR) in Time", *Proc. IEEE Aerospace Conference*, Big Sky, Montana, Mar. 5-12, 2005.
56. J. Lan, D. Lam, L. McMurchie, and C. Sechen, "SEE-Hardened-by-Design Area-Efficient SRAMs", *Proc. IEEE Aerospace Conference*, Big Sky, Montana, Mar. 5-12, 2005.
57. Y. Han, L. McMurchie and C. Sechen, "A High Performance CMOS Programmable Logic Core," *Proc. Custom Integrated Circuits Conference (CICC)*, October 3-6, 2004, Orlando, FL.
58. V. Tang, J. Lan, D. Lam, L. McMurchie, and C. Sechen, "High-Performance SEE-Hardened Programmable DSP Array," *Proc. 2004 Military and Aerospace Programmable Logic Device (MAPLD) Conference*, Washington, DC, Sept. 8-10, 2004.
59. D. Lam, J. Lan, L. McMurchie, and C. Sechen, "Radiation-Hardened-by-Design Area-Efficient SRAMs," *Proc. Hardness by Design (HBD) Workshop*, Albuquerque, NM, August 24-25, 2004.
60. M. Vujkovic, D. Wadkins, B. Swartz and C. Sechen, "Efficient Timing Closure Without Timing Driven Placement and Routing," *Proc. Design Automation Conference (DAC)*, San Diego, CA, June 7-11, 2004.
61. Y. Lam, L. McMurchie, and C. Sechen, "SEU Hardening of Peripheral Circuitry in Self-Scrubbing SRAMs," *Proc. Single Event Effects Symposium*, Manhattan Beach, CA, April 27-29, 2004.

62. V. Tang, J. Lan, D. Lam, L. McMurchie and C. Sechen, "Low-Overhead SEE-Hardened Programmable DSP Array," *Proc. Single Event Effects Symposium*, Manhattan Beach, CA, April 27-29, 2004.
63. L. McMurchie and C. Sechen, "RADAR -- Reconfigurable Analog and Digital Array for Radiation-Hardened Circuits", *Proc. IEEE Aerospace Conference*, Big Sky, Montana, Mar. 6-13, 2004.
64. Y. Han and C. Sechen, "A Novel High Speed FPGA Architecture, *Proc. SRC Techcon*, Dallas, TX, Aug. 25 – 28, 2003.
65. C. Sechen, "The End of Libraries as We Know Them," *Proc. 40th Design Automation Conference (DAC)*, pp. 642-643, 2003.
66. S. Kio, K. Chong, and C. Sechen, "A Low Power Delayed-Clock Generation and Distribution System," *Proc. Int. Symp. on Circuits and Systems (ISCAS)*, May 25-28, 2003, Bangkok, Thailand.
67. Warren Snapp, Paul Haug, Dave Sunderland, Roger Brees Don Bouldin, Wayne Dai, and Carl Sechen, "MSP Liberator ASIC Design Flow Produces Full Custom Performance Required for Next Generation Military Electronics," *Proc. Government Microelectronics and Applications Conference (GOMAC)*, April 1-3, 2003, Tampa, FL.
68. H. Tennakoon and C. Sechen, "Gate Sizing Using Lagrangian Relaxation Combined with a Fast Gradient-Based Preprocessing Step," *Proc. Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 2002, San Jose, CA.
69. M. Vujkovic and C. Sechen, "Optimized Power-Delay Curve Generation for Standard Cell ICs," *Proc. Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 2002, San Jose, CA.
70. L. McMurchie and C. Sechen, "WTA – Waveform-Based Timing Analysis for Deep Submicron Circuits," *Proc. Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 2002, San Jose, CA.
71. M. Vujkovic and C. Sechen, "Optimized Power-Delay Curve Generation for Standard Cell ICs," *Proc. Int. Workshop on Logic Synthesis (IWLS)*, June 4-7, 2002, New Orleans, LA.
72. Yi Han, Larry McMurchie, Jovanka Ciric, and Carl Sechen, "Giga hertz-Speed FPGA Architecture based on Output Prediction Logic", *Tenth ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA 2002)*.
73. J. Ciric and C. Sechen, "Efficient Canonical Form for Boolean Matching of Complex Functions in Large Libraries," *Proc. of IEEE Int. Conf. on Comp. Aided Design (ICCAD)*, San Jose, CA, November 5-7, 2001.
74. R. Rutenbar, M. Baron, T. Daniel, R. Jayaraman, Z. Or-Bach, J. Rose and C. Sechen, "When Will FPGAs Kill ASICs?" *Proc. Design Automation Conference (DAC)*, June 18-22, 2001, Las Vegas, NV.
75. J. Ciric and C. Sechen, "Efficient Canonical Form for Boolean Matching of Complex Functions in Large Libraries," *Proc. of IEEE Int. Workshop on Logic Synthesis*, Lake Tahoe, CA, June 12-15, 2001.
76. S. Kio, L. McMurchie, and C. Sechen, "Application of Output Prediction Logic to Differential CMOS," *Proc. of IEEE Computer Society Annual Workshop on VLSI*, Orlando, FL, April 19-20, 2001.
77. S. Sun, L. McMurchie, and C. Sechen, "A High-Performance 64-bit Adder Implemented in Output Prediction Logic," *Proc. 19th Conference on Advanced Research in VLSI*, March 14-16, 2001, Salt Lake City, UT.
78. T. Serdar and C. Sechen, "Automatic Datapath Tile Placement and Routing," *Proc. of the Design, Automation and Test in Europe (DATE) Conference*, March 13-16, 2001, Munich, Germany.
79. S. Kio, L. McMurchie, and C. Sechen, "Output Prediction Logic", *Proc. SRC Techcon*, September 21-23, 2000, Phoenix, AZ.

80. M. Vujkovic, G. Yee, and C. Sechen, "Post-Fabrication Automatically Tunable Programmable Delay Elements for Clock-Delayed Domino Logic," *Proc. SRC Techcon*, September 21-23, 2000, Phoenix, AZ.
81. G. Hoyer and C. Sechen, "A Locally-Clocked Dynamic Logic Serial/Parallel Multiplier," *Proc. SRC Techcon*, September 21-23, 2000, Phoenix, AZ.
82. L. McMurchie, S. Kio, G. Yee, T. Thorp and C. Sechen, "Output Prediction Logic: A High Performance CMOS Design Technique", *Proc. Int. Conf. On Computer Design (ICCD)*, September 17-20, 2000, Austin, TX.
83. G. Hoyer and C. Sechen, "A Locally-Clocked Dynamic Logic Serial/Parallel Multiplier", *Proc. of Custom Integrated Circuits Conference (CICC)*, May 21-25, 2000, Orlando, FL.
84. Gin Yee, Ron Christopherson, Tyler Thorp, Ban P. Wong, and Carl Sechen, "An automated shielding algorithm and tool for dynamic circuits," *Proc. Intl. Symposium on Quality Electronic Design (ISQED)*, San Jose, CA, March 20-22, 2000.
85. J. Ciric, G. Yee, C. Sechen, "Delay Minimization and Technology Mapping of Two-Level Structures and Implementation Using Clock-Delayed Domino Logic," *Proc. Design Automation & Test in Europe (DATE2000)*, 27-30 March 2000, Paris, France.
86. T. Serdar and C. Sechen, "AKORD: Transistor Level and Mixed Transistor/Gate Level Placement Tool for Digital Data Paths," *Proc. Int. Conf. on Computer-Aided Design (ICCAD)*, San Jose, CA, November 1999.
87. T. Thorp, G. Yee and C. Sechen, "Monotonic Static CMOS and Dual Vt Technology," *Proc. Int. Symp. on Low Power Electronics and Design (ISPLED)*, San Diego, CA, August 16-17, 1999.
88. T. Thorp, G. Yee and C. Sechen, "Monotonic Logic and Dual Vt Technology", *Proc. of Int. Conf. on Computer Design (ICCD)*, Austin, TX, October 1999.
89. G. Hoyer, G. Yee and C. Sechen, "SEU Tolerant Locally-Clocked Dynamic Logic, *Proc. Government Microelectronics and Applications Conference (GOMAC)*, March 1999, Monterey, CA.
90. T. Thorp and C. Sechen, "Domino logic synthesis using complex static gates", *Proc. Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 9-11, 1998, San Jose, CA.
91. H. Tennakoon and C. Sechen, "Linear Time Heuristic for Timing Optimization", *Proc. SRC TECHCON Conf.*, Sept. 9-11, 1998, Las Vegas, NV.
92. G. Hoyer, G. Yee and C. Sechen, "Locally-clocked Dynamic Logic", *Proc. Midwest Conf. On Circuits and Systems*, Aug. 10-12, 1998, Notre Dame, IN.
93. J. Ciric and C. Sechen, "Clock-delayed Domino Synthesis Using Decomposition into Two-level Structures", *Proc. SRC TECHCON Conf.*, Sept. 9-11, 1998, Las Vegas, NV.
94. G. Hoyer and C. Sechen, "Locally-clocked Dynamic Logic", *Proc. SRC TECHCON Conf.*, Sept. 9-11, 1998, Las Vegas, NV.
95. H. P. Tseng, L. Liu and C. Sechen, "Chip-level Area Routing", *Proc. International Symposium on Physical Design (ISPD)*, April 1998, Monterey, CA.
96. H. P. Tseng, L. Scheffer and C. Sechen, "Timing and Crosstalk Driven Area Routing," *Proc. IEEE Design Automation Conference (DAC)*, June 1998, San Francisco, CA.
97. T. Thorp, G. Yee and C. Sechen, "Dynamic Logic Synthesis using Alternating Dynamic and Static Gates," *Proc. Int. Workshop on Logic Synthesis*, June 1998, Lake Tahoe, CA.
98. G. Yee, R. Chandra, H. Ganesan, and C. Sechen, "Wire Delay in the Presence of Crosstalk," *ACM/IEEE Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Dec. 4-5, 1997, Austin, Texas.
99. M. Lefebvre, D. Marple and C. Sechen, "The Future of Custom Cell Generation in Physical Synthesis," *Proc. Design Automation Conference (DAC)*, Anaheim, CA, June 1997, pp. 446-451.
100. G. Yee and C. Sechen, "Dynamic Logic Synthesis," *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, May 5-8, 1997, Santa Clara, CA, pp. 345-348.

101. H. P. Tseng and C. Sechen, "Multi-layer Over-the-Cell Routing with Obstacles," *Proc. IEEE Custom Integrated Circuits Conference*, May 5-8, 1997, Santa Clara, CA, pp. 565-568.
102. L. Liu and C. Sechen, "Multi-layer Chip-Level Global Routing Using an Efficient Graph-based Steiner Tree Heuristic," *Proc. European Design and Test Conference (EDTC)*, March 18-20, 1997, Paris, France.
103. H. P. Tseng and C. Sechen, "A Gridless Multi-layer Router for Standard Cell Circuits Using Central Terminal Model Cells," *Proc. European Design and Test Conference (EDTC)*, March 18-20, 1997, Paris, France.
104. B. Guan and C. Sechen, "Large Standard Cell Libraries and Their Impact on Layout Area and Circuit Performance," *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, October 7-9, 1996, Austin, TX.
105. G. Yee and C. Sechen, "Clock-delayed Domino for Adder and Random Logic Design," *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, October 7-9, 1996, Austin, TX.
106. G. Yee and C. Sechen, "Low voltage and low power: how low can you go?," *Proceedings International Conference on Computer Design. VLSI in Computers and Processors*, 1996, pp. 342-, doi: 10.1109/ICCD.1996.563576.
107. B. Guan and C. Sechen, "ASIC Automatic Layout Generation Using Large Standard Cell Libraries," *Proc. International Conference on ASIC (ASICON)*, October 21-24, 1996, Shanghai, China.
108. B. Guan and C. Sechen, "Efficient Standard Cell Generation When Diffusion Strapping Is Required," *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, May 1996, San Diego, CA.
109. Q. Yu and C. Sechen, "Approximate Symbolic Analysis of Large Analog Integrated Circuits," *Proc. of the European Conference on Circuit Theory and Design (ECCTD)*, August 27-31, 1995, Istanbul, Turkey.
110. W. Swartz and C. Sechen, "Timing Driven Placement for Large Circuits," *Proc. 1995 Design Automation Conference (DAC)*, San Francisco, CA, June 1995.
111. T. Stanion and C. Sechen, "A Method for Finding Good Ashenhurst Decompositions and Its Application for FPGA Synthesis," *Proc. 1995 Design Automation Conference (DAC)*, San Francisco, CA, June 1995.
112. B. Guan and C. Sechen, "An Area Minimizing Layout Generator for Random Logic Blocks," *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, May 1-4, 1995, Santa Clara, CA.
113. Q. Yu and C. Sechen, "Efficient Approximation of Symbolic Network Functions Using Matroid Intersection Algorithms," *Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 1-3, 1995, Seattle, WA.
114. J. Hsu and C. Sechen, "Simplified Symbolic Pole and Zero Expressions from the Sifting Approach to Symbolic Analysis," *Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 1-3, 1995, Seattle, WA.
115. T. Stanion and C. Sechen, "Quasi-Algebraic Decomposition of Switching Functions," *Proc. 16th Conference on Advanced Research in VLSI*, March 27-29, 1995, Chapel Hill, NC, pp. 358-367.
116. K. Roy and C. Sechen, "Multiple FPGA Partitioning with Performance Optimization," *Proc. of the IEEE/ACM International Symposium on FPGAs*, March 12-14, Monterey, CA, 1995, pp. 146-152.
117. W. Sun and C. Sechen, "A Loosely Coupled Parallel Algorithm for Standard Cell Placement," *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 6-10, 1994, Santa Clara, CA, pp. 137-144.
118. Q. Yu and C. Sechen, "Approximate Symbolic Analysis of Large Analog Integrated Circuits," *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 6-10, 1994, Santa Clara, CA, pp. 664-671.

119. J. J. Hsu and C. Sechen, "Fully Symbolic Analysis of Large Analog Integrated Circuits," *IEEE Custom Integrated Circuits Conference (CICC)*, May 2-4, 1994, San Diego, CA, pp. 457-460.
120. Q. Yu and C. Sechen, "Generation of Color-Constrained Spanning Trees with Application in Symbolic Analysis," *Fourth Great Lakes Symposium on VLSI*, March 4-5, 1994, Notre Dame, IN, pp. 252-255.
121. K. Roy and C. Sechen, "A Timing Driven N-Way Chip and Multi-Chip Partitioner," *Proc. SRC TECHCON Conference*, September 28-30, 1993, Atlanta, GA.
122. W. Swartz and C. Sechen, "Precise Timing Driven Placement for Large Circuits," *Proc. SRC TECHCON Conference*, September 28-30, 1993, Atlanta, GA.
123. W. Swartz and C. Sechen, "A New Generalized Row-Based Global Router," *Proc. of the SRC TECHCON Conference*, September 28-30, 1993, Atlanta, GA.
124. T. Stanion and C. Sechen, "Maximum Projections of Don't Care Conditions in a Boolean Network," *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 7-11, 1993, Santa Clara, CA, pp. 674-679.
125. W. Sun and C. Sechen, "Efficient and Effective Placement for Very Large Circuits," *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 7-11, 1993, Santa Clara, CA, pp. 170-177.
126. K. Roy and C. Sechen, "A Timing Driven N-Way Chip and Multi-Chip Partitioner," *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 7-11, 1993, Santa Clara, CA, 240- 247.
127. W. Swartz and C. Sechen, "A New Generalized Row-Based Global Router," *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 7-11, 1993, Santa Clara, CA, pp. 491- 498.
128. K. Roy, B. Guan and C. Sechen, "A Sea-of-Gates Style FPGA Placement Algorithm," *Int. Conf. on VLSI Design*, January 5-8, 1994, Calcutta, India.
129. J. J. Hsu and C. Sechen, "Low-Frequency Symbolic Analysis of Large Analog Integrated Circuits," *IEEE Custom Integrated Circuits Conference (CICC)*, May 9-12, 1993, San Diego, CA, pp. 14.7.1-14.7.5.
130. T. Stanion and C. Sechen, "Boolean Division and Factorization Using Binary Decision Diagrams," *IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS '92)*, Dec. 8-11, 1992, Sydney, Australia.
131. K. W. Lee and C. Sechen, "A Global Router for Sea-of-Gates Circuits," *Proc. European Design Automation Conference*, Amsterdam, The Netherlands, February 26-28, 1991, pp. 242-247.
132. D. Chen and C. Sechen, "Mickey: A Macro Cell Global Router", *Proc. European Design Automation Conference*, Amsterdam, The Netherlands, February 26-28, 1991, pp. 248- 252.
133. W. Swartz and C. Sechen, "New Algorithms for the Placement and Routing of Macro Cells," *Proc. Int. Conf. on Comp. Aided Design (ICCAD)*, Santa Clara, CA, November 11-15, 1990, pp. 336-339.
134. D. Chen and C. Sechen, "Mickey: A New Macro Cell Global Router," *Proc. SRC Techcon Conference*, San Jose, CA, October 16-18, 1990, pp. 241-244.
135. W. Swartz and C. Sechen, "New Algorithms for the Placement and Routing of Macro Cells," *Proc. SRC Techcon Conference*, San Jose, CA, October 16-18, 1990, pp. 233-236.
136. K. W. Lee and C. Sechen, "A New Global Router for Sea-of-Gates Circuits," *Proc. 9th Australian Microelectronics Conference*, Adelaide, S. Australia, July 2-4, 1990, pp. 233-244.
137. K. W. Lee and C. Sechen, "A New Global Router for Row-Based Layout," *Proc. of Int. Conf. on Computer-Aided Des. (ICCAD)*, Nov. 7-10, 1988, Santa Clara, CA, pp. 180-183.
138. C. Sechen and D. Chen, "An Improved Objective Function for Mincut Circuit Partitioning," *Proc. of Int. Conf. on Computer-Aided Des. (ICCAD)*, Nov. 7-10, 1988, Santa Clara, CA, pp. 502-505.

139. C. Sechen, "Chip-Planning, Placement, and Global Routing of Macro/Custom Cell Circuits Using Simulated Annealing," *Proc. of the 25th Design Automation Conference (DAC)*, June 1988, Anaheim, CA, pp. 73-80.
140. C. Sechen and K. W. Lee, "An Improved Simulated Annealing Algorithm for Row-Based Placement," *Proc. of Int. Conf. on Computer-Aided Des. (ICCAD)*, Nov. 9-13, 1987, Santa Clara, CA, pp. 478-481.
141. C. Sechen, "Average Interconnection Length Estimation for Random and Optimized Placements," *Proc. of Int. Conf. on Computer-Aided Des. (ICCAD)*, Nov. 9-13, 1987, Santa Clara, CA, pp. 190-193.
142. J. Burns, A. Casotto, M. Igusa, F. Marron, F. Romeo, A. Sangiovanni-Vincentelli, C. Sechen, H. Shin, G. Srinath, and H. Yaghtiel, "Mosaico: An Integrated Macro-Cell Layout System," in C. Sequin, editor, *Proc. of VLSI '87*, Vancouver, Canada, August, 1987.
143. C. Sechen and A. Sangiovanni-Vincentelli, "TimberWolf3.2: A New Standard Cell Placement and Global Routing Package," *Proc. 23rd Design Automation Conference (DAC)*, June 29-July 2, 1986, Las Vegas, NV, pp. 432-439.
144. D. Braun, C. Sechen, and A. Sangiovanni-Vincentelli, "ThunderBird: A Complete Standard Cell Layout Package," *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 1986, Rochester, NY.
145. F. Romeo, C. Sechen, and A. Sangiovanni-Vincentelli, "Simulated Annealing Research at Berkeley," *Proc. Int. Conf. on Computer Design (ICCD)*, October 1984, Rye Brook, NY.
146. C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf Placement and Routing Package," *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 1984, Rochester, NY.
147. M. Huberman, S. Senturia, and C. Sechen, "The Use of the Charge-Flow Transistor to Distinguish Surface and Bulk Components of Thin-Film Sheet Resistance," *IEEE Device Research Conference (DRC)*, Cornell University, June 1977.
148. S. Senturia and C. Sechen, "The Charge-Flow Transistor," *IEEE Int. Electron Devices Meeting (IEDM)*, Washington DC, Dec. 1976.
149. L. Liu and C. Sechen, "A Multi-Layer Chip-Level Global Router," *Proc. of the Fifth ACM/SIGDA Physical Design Workshop*, April 15-17, 1996, Reston, Virginia.
150. L. Liu and C. Sechen, "Multi-Layer Pin Assignment for Macro Cell Circuits," *Proc. of the Fifth ACM/SIGDA Physical Design Workshop*, April 15-17, 1996, Reston, Virginia.
151. G. Yee and C. Sechen, "Clock-Delayed Domino for Adder and Combinatorial Logic Design," *Proc. of the Fifth ACM/SIGDA Physical Design Workshop*, April 15-17, 1996, Reston, Virginia.
152. B. Guan and C. Sechen, "Efficient Standard Cell Generation When Diffusion Strapping is Required," *Proc. of the Fifth ACM/SIGDA Physical Design Workshop*, April 15-17, 1996, Reston, Virginia.
153. H. P. Tseng and C. Sechen, "A Gridless Multi-Layer Channel Router Based on a Combined Constraint Graph and Tile Expansion Approach," *Proc. of the Fifth ACM/SIGDA Physical Design Workshop*, April 15-17, 1996, Reston, Virginia.
154. J. J. Hsu and C. Sechen, "The Sifting Approach to Symbolic Analysis of Large Analog Integrated Circuits," *Proc. of the Third International Workshop on Symbolic Methods and Applications to Circuit Design (SMACD)*, Sevilla, Spain, October 6-7, 1994, pp. 211-230.
155. Q. Yu and C. Sechen, "Approximate Symbolic Analysis of Large Analog Integrated Circuits," *Proc. of the Third International Workshop on Symbolic Methods and Applications to Circuit Design (SMACD)*, Sevilla, Spain, October 6-7, 1994, pp. 241-259.
156. Q. Yu and C. Sechen, "Efficient Approximation of Symbolic Network Functions Using Matroid Intersection Algorithms," *Proc. of the Third International Workshop on Symbolic Methods and Applications to Circuit Design (SMACD)*, Sevilla, Spain, October 6-7, 1994, pp. 261-277.

157. T. Stanion and C. Sechen, "Maximum Projections of Don't Care Conditions in a Boolean Network," *Proc. of the International Workshop on Logic Synthesis*, Lake Tahoe, CA, May 23-26, 1993.
158. W. Sun, K. Roy, and C. Sechen, "Fast, High-Quality Placement for Large Circuits," *Proc. of the 4th ACM/SIGDA Physical Design Workshop*, Lake Arrowhead, CA, April 19-21, 1993.
159. K. Roy, D. Guan and C. Sechen, "FPGA MCM Partitioning and Placement," *Proc. of the 4th ACM/SIGDA Physical Design Workshop*, Lake Arrowhead, CA, April 19-21, 1993.
160. T. Stanion and C. Sechen, "Incremental and Simultaneous Computation of Satisfiability and Observability Don't Cares," *Proc. IEEE/ACM Winter VLSI Workshop*, Jan. 25-27, 1993, Asilomar, CA.
161. R. Weier and C. Sechen, "TimberWolfAR: An Arbitrary Design Rule Multi-Layer Area Router," *Int. Workshop on Layout Synthesis*, May 18-21, 1992, MCNC, Research Triangle Park, NC.
162. W. Swartz and C. Sechen, "A Generalized Row-Based Global Router for FPGAs," *Int. Workshop on Layout Synthesis*, May 18-21, 1992, MCNC, Research Triangle Park, NC.
163. K. Roy and C. Sechen, "A Timing Driven N-Way Chip Partitioner," *Int. Workshop on Layout Synthesis*, May 18-21, 1992, MCNC, Research Triangle Park, NC.
164. K. W. Lee and C. Sechen, "A Global Router for Sea-of-Gates Circuits," *Proc. Int. Workshop on Layout Synthesis*, Research Triangle Park, NC, May 8-11, 1990.
165. D. Chen and C. Sechen, "Mickey: A New Macro Cell Global Router", *Proc. Int. Workshop on Layout Synthesis*, Research Triangle Park, NC, May 8-11, 1990.
166. W. Swartz and C. Sechen, "New Algorithms for the Placement and Routing of Macro Cells," *Proc. Int. Workshop on Layout Synthesis*, Research Triangle Park, NC, May 8-11, 1990.
167. C. Sechen, "TimberWolfSG: A Sea-of-Gates Layout Package," *Proc. of SRC Workshop on Sea-of-Gates Design*, Sept. 22-23, 1988, Berkeley, CA.
168. C. Sechen, "A Global Router for the Sea-of-Gates Environment," *Proc. IFIP Workshop on Physical Design of Sea-of-Gates VLSI*, August 11-12, 1988, Lenggries, Germany.
169. J. Lam, J. M. Delosme, and C. Sechen, "A New Simulated Annealing Schedule for Row-Based Placement," *Proc. IEEE International Workshop on Placement and Routing*, May 10-13, 1988, Research Triangle Park, NC.
170. F. Romeo, C. Sechen, and A. Sangiovanni-Vincentelli, "Some Theoretical Results on Simulated Annealing and TimberWolf: A Placement and Routing Package for the Layout of Integrated Circuits," *Proc. Workshop on Statistical Physics in Engineering and Biology*, IBM T.J. Watson Research Center, Yorktown Heights, NY, April 1984.

Books Authored

Carl Sechen, **Placement and Global Routing of Integrated Circuits Using Simulated Annealing**, 278 pages, Kluwer Academic Publishers, Boston, MA, 1988.

Patents

C. Sechen, Y. Makris, T. Broadfoot, "Field Programmable Transistor Arrays", US Patent 11,362,662 issued June 14, 2022. (New patent relative to the below two.)

C. Sechen, Y. Makris, G. Reddy, J. Tian, "Field Programmable Transistor Arrays", US Patent 10,855,285 B2, issued 1 December 2020. (New patent relative to the 2019 one.)

C. Sechen, Y. Makris, G. Reddy, J. Tian, “Field Programmable Transistor Arrays”, US Patent No. 10,511,308, issued December 17, 2019.

C. Sechen, L. McMurchie, G. Yee and T. Thorp, “Method of High Performance CMOS Design”, US Patent No. 6,549,038 issued April 15, 2003.

Research Funding

- “Cost-Effective IC Redaction through a CMOS/MRAM Transistor-Level Programmable Fabric”, PI: C. Sechen, Co-PIs: J. Friedman, Y. Makris. NSF CHEST I/UCRC. June 1, 2023 – May 31, 2024. \$100,000.
- NSF Proposal: SaTC: TTP: Medium: Hardware Intellectual Property Protection through Hybrid ASIC/TRAP Integrated Circuit Design, 8/15/22-7/31/26, \$900,000 (PI: Y. Makris, co-PI: C. Sechen).
- “CAD Flow Development for Design Obfuscation through Post-Fabrication Transistor-Level Programming (TRAP)”, Dr. Yiorgos Makris, Dr. Carl Sechen, Dr. Benjamin Carrion Schaefer, ECE Department, UT Dallas, 800 W. Campbell Rd., ECSN 4.914, Richardson, TX 75080, \$150,000. NSF CHEST I/UCRC. March 1, 2022 – February 28, 2023.
- “CAD Flow Development for Design Obfuscation through Post-Fabrication Transistor-Level Programming (TRAP)”, Dr. Yiorgos Makris, Dr. Carl Sechen, Dr. Benjamin Carrion Schaefer, Dr. William Swartz Jr., ECE Department, UT Dallas, 800 W. Campbell Rd., ECSN 4.914, Richardson, TX 75080, \$150,000. NSF CHEST I/UCRC. March 1, 2021 – February 28, 2022.
- IBM / NSWC CW3085860, “Rapid Assured Microelectronics Prototypes using Advanced Commercial Capabilities (RAMP)”, 10/8/20-3/20/21, \$143,576 (PI: Y. Makris, co-PI: C. Sechen)
- “DEFENSEWERX-Advanced Microelectronics Design & Prototype Challenge”, Benjamin Carrion Schaefer, Yiorgos Makris, Carl Sechen, William Swartz Jr., InertialWave/Defensewerks/Air Force, August 1, 2020 – November 30, 2020, \$200,000.
- “CAD Flow Development for Design Obfuscation through Post-Fabrication Transistor-Level Programming (TRAP)”, Dr. Yiorgos Makris, Dr. Carl Sechen, Dr. Benjamin Carrion Schaefer, Dr. William Swartz Jr., ECE Department, UT Dallas, 800 W. Campbell Rd., ECSN 4.914, Richardson, TX 75080, \$100,000. NSF CHEST I/UCRC. March 1, 2020 – February 28, 2021.
- “TRAP-based Circuit Obfuscation Development and Evaluation”, PIs: Y. Makris, B. Schaefer, W. Swartz and C. Sechen, The Boeing Company, \$149,914, February 1, 2020 – August 31, 2020.

- “DEFENSEWERX-Advanced Microelectronics Design & Prototype Challenge”, Benjamin Carrion Schaefer, Yiorgos Makris, Carl Sechen, William Swartz Jr., InertialWave/Defenseworks/Air Force, Nov. 15, 2019 – Mar. 15, 2020, \$87,980.
- “OpenROAD: Foundations and Realizations of Open, Accessible Design”, co-PIs Carl Sechen and William Swartz, Defense Advanced Research Products Agency (DARPA) IDEA Program, May 1, 2018 through April 30, 2019, \$287,341.
- CSR: “Small: Host-Assisted Software-Defined Solid-State Disk”, PI: Carl Sechen, October 1, 2014 – September 30, 2019, \$265,067.
- TerraSwarms Research Center, DARPA/SRC FCRP research center based at UC Berkeley, Jan. 15, 2013 – Jan. 14, 2014, co-PI: Carl Sechen portion: \$100,000.
- “Optimal Gate Size and Vt Selector, and Innovative Approach to Path-Based Timing Analysis”, PI: Carl Sechen, Texas Instruments, Jan. – Dec., 2012, \$50,000. Current status: completed.
- “Optimal Gate Size and Vt Selection”, Texas Instruments, Wireless Division, Jan. – Dec., 2011, \$50,000, PI: Carl Sechen.
- “Power Minimization Research”, Texas Instruments, ASIC Division, Jan. – Dec., 2011, \$50,000, PI: Carl Sechen.
- “Power Minimization Research”, Texas Instruments, ASIC Division, Jan. – Dec., 2010, \$75,000, PI: Carl Sechen.
- “Low Power HD Video Decoder Research”, Texas Instruments, Wireless Division, Jan. – Dec., 2010, \$50,000, PI: Carl Sechen.
- “Power Minimization Research”, Texas Instruments, ASIC Division, Jan. – Dec., 2009, \$50,000, PI: Carl Sechen.
- “Low Power HD Video Decoder Research”, Texas Instruments, Wireless Division, Jan. – Dec., 2009, \$50,000, PI: Carl Sechen.
- “Power Optimization and Circuit Hardening”, MARCO Focus Center Research Program (C2S2), \$155,000, September 2005 – August 2008.
- “Ultra High Performance Digital Circuit Design and Synthesis”, National Science Foundation (NSF), \$80,000, (Jan. 2006 – Dec., 2008).
- “Low Power HD Video Decoder Research”, Carl Sechen, Texas Instruments, Wireless Division, \$80,000, Jan. – Dec., 2008.
- “Power Minimization Research”, Carl Sechen, Texas Instruments, ASIC Division, \$37,000, Jan. – Dec., 2008.
- “Power Minimization and Yield Optimization in Digital IC's”, Carl Sechen, Semiconductor Research Corporation TxACE, \$150,000, August 2007 – July 2010.

- “Analog-to-Digital Converter based on Time-to-Digital Conversion”, Carl Sechen, Texas Instruments, \$40,000, Sep. 2007 – Aug. 2008.
- “Low Power HD Video Decoder Research”, Carl Sechen, Texas Instruments, Wireless Division, \$45,000, Jan. – Dec., 2007.
- “Three-Transistor Memory Cell Research”, Carl Sechen, Texas Instruments, \$35,000, Jan. – Dec., 2007.
- “Power Minimization Research”, Carl Sechen, Texas Instruments, ASIC Division, \$37,000, Jan. – Dec., 2007.
- “Low Power HD Video Decoder Research”, Carl Sechen, Texas Instruments, Wireless Division, \$30,000, Sept. – Dec., 2006.
- “DRAM-Density SRAM”, Carl Sechen, Honeywell, \$105,000, January 2006 – April 2007.
- “RADHARD by Design,” DARPA/Boeing, Phase I, \$45,000 (co-PI, my portion), November 2004 – August 2005).
- “CLASS: Clockless Logic, Analysis, Synthesis, and Systems,” DARPA/Boeing, Phase II/III, \$709,000 (co-PI, my portion), (Sept. 2004 – August 2006).
- “MSP Phase 2: Digital IC Optimization Techniques,” DARPA/Boeing, \$285,000 (co-PI, my portion), (September 2004 – August 2006).
- “Power Optimization and Circuit Hardening,” MARCO FCRP C2S2 Center, \$100,000, PI (Oct. 04 – Sep. 05).
- “High Speed and Low Power Digital IC Design,” NSF ITR, \$350,000 (my portion) co-PI, (Sept. 00 – Aug. 05).
- “Reconfigurable RF, Analog, and Digital Array for Radiation-Hardened Communication Circuits,” Air Force Research Lab, \$110,000, PI (Sep 04 – Aug 05).
- “High Speed & Low Power Digital Circuit Synthesis and Design,” MARCO FCRP C2S2 Center, \$100,000, PI (Oct. 03 – Sep. 04), full indirect.
- “CLASS: Clockless Logic, Analysis, Synthesis, and Systems,” DARPA/Boeing, Phase I, \$135,000 (co-PI, my portion), (Oct. 2003 – May 2004).
- “Reconfigurable RF, Analog, and Digital Array for Radiation-Hardened Communication Circuits,” Air Force Research Lab, \$105,000, PI (Sep 03 – Aug 04), full indirect.
- “Low Power Digital IC Design,” NSF Center for the Design of Analog and Digital ICs (CDADIC), \$55,000, PI (Sep 03 – Aug 04), no indirect.
- “Reconfigurable RF, Analog and Digital Array for Radiation-Hardened Wireless Communication Circuits,” DARPA/SPAWAR, \$150,000, PI (April 03 – May 04), full indirect.

- “Development of Pulse-Based Logic,” DARPA, \$100,000, PI (March 03 – May 04), full indirect.
- “High Performance and Power Efficient Digital IC Optimization Techniques,” MSP Phase 1: Boeing/DARPA, \$400,000, PI (April 02 – August 03), full indirect.
- Unrestricted grant, Intel Corporation, \$60,000, PI (May 03 – April 04).
- “High Performance Digital IC Design,” MARCO/DARPA C2S2 Center, \$170,000, PI (Jan. 01 – Aug. 03), full indirect.
- “Ultra High Performance Digital Circuit Design and Synthesis,” National Science Foundation, \$180,000, PI, (Sep 02 – Aug 05).
- “Reconfigurable RF, Analog, and Digital Array for Radiation-Hardened Communication Circuits,” Air Force Research Lab, \$95,000, PI (Sep 02 – Aug 03), full indirect.
- “PLL Design, Clock Generation and High-Speed/Low-Power DSP-block Design,” NSF Center for the Design of Analog and Digital ICs (CDADIC), \$55,000, PI (Sep 02 – Aug 03), no indirect.
- Unrestricted grant, Intel Corporation, \$60,000, PI (Mar 02 – Feb 03), no indirect.
- “High Performance Digital IC Design,” Semiconductor Research Corporation (SRC), June 1, 2000 – May 31, 2003, \$525,000 (full indirect).
- “High Performance Digital IC Design,” NSF Center for the Design of Analog and Digital ICs (CDADIC), \$60,000 (no indirect), PI (Sep. 00 - Aug. 01).
- Unrestricted grant, Intel Corporation, \$60,000, PI (Mar 01 – Feb 02).
- Unrestricted grant, Sun Microsystems, \$30,000, PI (Feb 00 – Mar 01).
- Unrestricted grant, Compaq Computer Corp., \$25,000, PI (Oct. 99 – Sep. 00).
- Unrestricted grant, Cadabra, Inc., \$20,000, PI (Feb. 00 – Mar 01).
- “Ultra High Speed Digital Circuit Synthesis and Layout NSF Center for the Design of Analog and Digital ICs (CDADIC), \$55,000 (no indirect), PI (Sep. 99-Aug. 00).
- “Ultra High Speed Digital Circuit Synthesis and Layout”, National Science Foundation, \$255,000 (full indirect), PI (Sep 99 – Aug 02).
- “Ultra High Speed Digital Circuit Synthesis and Layout”, Semiconductor Research Corporation, \$154,000 (full indirect), PI (Mar 99-Mar 00).
- “Ultra High Speed Digital Circuit Synthesis and Layout”, , Semiconductor Research Corporation, \$154,000 (full indirect), PI (Mar 99-Mar 00).

- “Synthesis of Ultra High Speed Random Logic Blocks”, Semiconductor Research Corporation, \$132,000, PI (Mar 98-Feb 99).
- “Ultra High Speed Digital Circuit Synthesis and Layout”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$65,000, PI (Sep. 98-Aug. 99).
- “Radiation-hard High Speed Digital Circuit Synthesis”, Boeing, \$35,000, PI (Sep. 98-Aug. 99).
- “Datapath Cell Layout”, Compaq/Digital Equipment, \$25,000, PI (Sep. 98 – Aug. 99).
- Unrestricted grant, Sun Microsystems, \$8,000, PI (June 99 – May 00).
- Unrestricted grant, Compaq Computer Corp., \$25,000, PI (Oct. 98 – Sep. 99).
- Unrestricted grant, Sun Microsystems, \$30,000, PI (Jan. 99 – Dec. 99).
- Unrestricted grant, TimberWolf Systems, Inc., \$20,000, PI (Apr. 98 – Mar. 99).
- Unrestricted grant, Sun Microsystems, Inc., \$8,000, PI (June 98 – May 99).
- Various industrial gifts, \$65,000, PI (Sep 97-Aug 98).
- “High Speed Digital Circuit Design”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$50,000, PI (Sep. 97-Aug. 98).
- “Automatic Layout Research”, Semiconductor Research Corporation, \$132,000, PI (Mar 97-Feb 98).
- Various industrial gifts, \$112,000, PI (Sep 96-Aug 97).
- “High Speed Circuit Design and Mixed Signal Layout”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$50,000, PI (Sep. 96-Aug. 97).
- “Symbolic Analysis of Large Analog Circuits”, National Science Foundation, \$90,000, PI (Sep 94-Aug 97).
- “Automatic Layout Research”, Semiconductor Research Corporation, \$132,000, PI (Mar 96-Feb 97).
- “High Speed Circuit Design and Mixed Signal Layout”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$45,000, PI (Sep. 95-Aug. 96).
- Various industrial gifts, \$117,000, PI (Sep 95-Aug 96).
- “Automatic Layout Research”, Semiconductor Research Corporation, \$132,000, PI (Mar 95-Feb 96).

- “High Speed Circuit Design and Mixed Signal Layout”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$45,000, PI (Sep. 94-Aug. 95).
- Various industrial gifts, \$208,000, PI (Sep 94-Aug 95).
- “Automatic Layout Research”, Semiconductor Research Corporation, \$144,000, PI (Mar 94-Feb 95).
- “High Speed Circuit Design and Mixed Signal Layout”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$45,000, PI (Sep. 93-Aug. 94).
- Various industrial gifts, \$343,000, PI (Sep 93-Aug 94).
- “Automatic Layout Research”, Semiconductor Research Corporation, \$140,000, PI (Mar 93-Feb 94).
- “High Speed Circuit Design and Mixed Signal Layout”, NSF Center for the Design of Analog and Digital ICs (CDADIC), \$45,000, PI (Sep. 92-Aug. 93).
- Various industrial gifts, \$312,000, PI (Sep 92-Aug 93).
- “Automatic Layout Research”, Semiconductor Research Corporation, \$140,000, PI (Mar 92-Feb 93).
- Various industrial gifts, \$306,000, PI (Sep 91-Aug 92).
- Yale University, Various Industrial Gifts, \$1.65 million, PI (1986-1992).

Conference Service Activities

- Session Chairman for the Design Automation and Test in Europe Conference, 2012-2014.
- Session Chairman for the IEEE Design Automation Conference, 1987, 1990, 1991, 1997.
- Session Chairman for the International Symposium on Physical Design, 1997, 1998.
- Session Chairman for the IEEE Int. Conf. on Computer Design (ICCD), 1988, 89, 90, 91, 92.
- Session Chairman for the IEEE Int. Conf. on Comp. Aided Design (ICCAD), 1989, 1990, 1991, 1992, 1993, 1998.
- Elected Chairman for the Placement and Floorplanning sub-committee for the 2004 Design Automation Conference (DAC), 2004.
- Elected to the Placement and Floorplanning sub-committee for the 2003 Design Automation Conference (DAC), 2003.
- Elected Program Chair for the 2000 ACM/IEEE Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU).
- Elected chairman of the placement and floorplanning committee of the 2001 Int. Conf. On Computer Aided Design (ICCAD).
- Elected chairman of the placement and floorplanning committee of the 2000 Int. Conf. On Computer Aided Design (ICCAD).
- Elected chairman of the placement and routing committee of the 1999 Int. Conf. On Computer Aided Design (ICCAD).
- Elected chairman of the placement and floorplanning committee of the 1998 Int. Conf. On Computer Aided Design (ICCAD).
- Elected Program Chairman of the ACM Physical Design Workshop, Lake Arrowhead, CA, April 1993.
- Elected Chairman of the 1992 Int. Workshop on Layout Synthesis, May, MCNC, Research Triangle Park, NC.
- Elected Chairman of the 1990 Int. Workshop on Layout Synthesis, May, MCNC, Research Triangle Park, NC.
- CAD Track Chairman for the IEEE Int. Conf. on Computer Design (ICCD), 1991.

Service as a Referee

- Technical Program Committee for the Design Automation Conference, 2014, 2015.
- Technical Program Committee for the Design Automation and Test in Europe Conference, 2012-2014.
- Technical Program Committee for the IEEE Int. Symposium on VLSI Design, 2010-2011.
- Technical Program Committee for the IEEE Int. Conf. on Comp. Aided Design (ICCAD), 1989-93, 98.
- Technical Program Committee for the IEEE Int. Conf. on Comp. Design (ICCD), 1987-97.
- Technical Program Committee for the European Design and Test Conference (EDTC), 1995, 96, 97, 98.
- Technical Program Committee for the Design Automation and Test in Europe (DATE) Conference, 1998, 99, 00, 01, 02, 03.
- Technical Program Committee for ACM International Physical Design Workshop, 1996.
- Technical Program committee for the International Symposium on Physical Design, 1997, 98.
- Technical Program Committee for the International Workshop on Layout Synthesis, 1988, 90, 92.
- IEEE/ACM Design Automation Conference, 1986-1998.
- National Science Foundation, 1986-present.

- IEEE Transactions on Computer Aided Design, 1986-present.
- IEEE Transactions on VLSI Systems, 1992-present.
- ACM Transactions on Design Automation of Electronic Systems, 1995-present.

University Service

UTD

Academic Affairs Committee 2022-2024

Director of ECS Tech Support Services, June 2012 – February 2014

Information Technology Planning and Policy Committee, Sept. 1, 2016 – August 31, 2019

UW

Faculty Council on Research, 1994-1997

Faculty Senate, 2002-2004

Department Service

UTD

Chair of the TA Assignment Committee, Spring 2024 and ongoing

Chair of the Computer Resources Committee, ECE Dept., 2007-2021

Advisory Committee to TxACE (SRC Texas Analog Center of Excellence) 2008-2009

Chair of the Analog Search Sub-Committee, EE Dept., 2007-2008

Faculty Search Committee (computer engineering) 2010-11

Teaching Assistant (TA) Selection Committee, 2021-23

UW

Faculty Search Committee, 2005-2006

Group Chair, VLSI and Digital Systems Group, 1997-2005

Faculty Search Committee, 98-99.

Graduate Studies and Research Committee, 1992-93, 96-97, 97-98.

Group Chair, Electronics, 94-95.

Graduate Admissions Committee, Electrical Engineering Dept., 1992-93.

Qualifying Examination Subcommittee, 1992-93.

Computer Resources Committee, 1992-93.

Outside University Service

Selected by the NSF to serve as a panelist for (invited) full proposals submitted to the CISE Expeditions in Computing (Expeditions) program, NSF 08-568, April 6-7, 2010.

Selected by the NSF to serve on the 2008 proposal review panel (April 14-15, 2008) to review a number of proposals that were submitted to the Computing Processes and Artifacts (CPA) cluster of the Computer, Information Sciences & Engineering (CISE) directorate of the National Science Foundation for Fiscal Year 2009 funding.

Selected by the NSF to serve on the 2007 proposal review panel (January 8-9, 2007) to review a number of proposals that were submitted to the Computing Processes and Artifacts (CPA) cluster of the Computer, Information Sciences & Engineering (CISE) directorate of the National Science Foundation for Fiscal Year 2008 funding.

Selected by the NSF to serve on the 2005 proposal review panel (October 20-21, 2005) to

review a number of proposals that were submitted to the Computing Processes and Artifacts (CPA) cluster of the Computer, Information Sciences & Engineering (CISE) directorate of the National Science Foundation for Fiscal Year 2006 funding.

Selected by the NSF to serve on the 2004 proposal review panel (May 5-6, 2004) to review a number of proposals that have been submitted to the Computing Processes and Artifacts (CPA) cluster of the Computer, Information Sciences & Engineering (CISE) directorate of the National Science Foundation for Fiscal Year 2004 funding.

Selected by the NSF to serve on the 2002 CAREER proposal evaluation panel for the Design Automation Program (Division of Communications-Computer Research or C-CR) of the National Science Foundation.

Selected by the NSF to serve on the 1999 CAREER proposal evaluation panel for the Design Automation Program (Division of Communications-Computer Research or C-CR) of the National Science Foundation.

Co-Director of the NSF Center for the Design of Analog and Digital ICs (CDADIC), 1997-2005; this center provides about \$450,000 annually to the University of Washington.

Professional Society Memberships

Eta Kappa Nu
Tau Beta Pi
Sigma Xi
IEEE Fellow
ACM