

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

KIOXIA CORPORATION and KIOXIA AMERICA, INC.

Petitioners,

v.

VIASAT, INC.

Patent Owner.

Case No. IPR2023-00408

Patent No. 8,615,700

Petition for *Inter Partes* Review

TABLE OF CONTENTS

	Page
I. Introduction.....	1
II. Mandatory Notices (37 C.F.R. §42.8)	2
A. Real Parties in Interest (37 C.F.R. §42.8(b)(1)).....	2
B. Related Matters (37 C.F.R. §42.8(b)(2)).....	2
C. Counsel and Service Information (37 C.F.R. §§42.8(b)(3)(4))	3
III. Payment of Fees (37 C.F.R. §42.103)	4
IV. Requirements for <i>Inter Partes</i> Review (37 C.F.R. §42.104)	4
A. Grounds for Standing (37 C.F.R. §42.104(a))	4
B. Identification of Challenge (37 C.F.R. §42.104(b)(1)-(2)) and Relief Requested (37 C.F.R. §42.22(a)(1)).....	4
V. The Board Should Not Decline Review Under Section 314 or 325.....	5
A. <i>General Plastic</i> Factors	5
B. <i>Fintiv</i> Factors.....	5
VI. Relevant Information Concerning the Contested Patent	8
A. Overview of the ‘700 Patent.....	8
B. The ‘700 Patent’s Priority Date.....	12
C. Overview of the ‘700 Patent Prosecution History	13
D. Person of Ordinary Skill in the Art	13
E. Claim Construction (37 C.F.R. §42.104(b)(3)).....	14
VII. The Specific Grounds of Unpatentability.....	15
A. Prior Art.....	15
1. Wu	15
2. Jo	18
3. Wan	20
B. Ground 1: Wu Anticipates Claims 1, 4, 8, 15, 17, and 19	23
1. Claim 1	23
2. Claim 4.....	35
3. Claim 8.....	36

4.	Claims 15 and 17.....	38
5.	Claim 19.....	42
C.	Ground 2: Jo in View of Wu Renders Claims 1, 2, 4, 8, 15, 17, and 19 Obvious	42
1.	Claim 1	42
2.	Claim 2.....	54
3.	Claim 4.....	55
4.	Claim 8.....	58
5.	Claims 15 and 17.....	59
6.	Claim 19.....	61
D.	Ground 3: Jo in View of Wan Renders Claims 1, 2, 4, 8, 15, 17, and 19 Obvious	62
1.	Claim 1	62
2.	Claim 2.....	70
3.	Claim 4.....	70
4.	Claim 8.....	71
5.	Claims 15 and 17.....	71
6.	Claim 19.....	73
VIII.	Conclusion	73

TABLE OF AUTHORITIES

Page(s)

CASES

<i>Apple Inc. v. Fintiv Inc.</i> , IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020)	5, 6, 8
<i>Apple Inc. v. Uniloc 2017 LLC</i> , IPR2020-00224, Paper 10 (PTAB Apr. 6, 2020)	5
<i>General Plastic Co., Ltd. v. Canon Kabushiki Kaisha</i> , IPR2016-01357, Paper 19, 9-10 (PTAB Sept. 6, 2017)	5
<i>In re GPAC Inc.</i> , 57 F.3d 1573 (Fed. Cir. 1995)	14
<i>Kioxia Corp. v. Viasat, Inc.</i> , IPR2022-01067, Paper 1, 13-14 (PTAB May 31, 2022)	3
<i>Maxlite, Inc. v. Jiaxing Super Lighting Elec. Appliance Co.</i> , IPR2020-00208, Paper 14, 6-8 (PTAB Jun. 1, 2021)	12
<i>New Railhead Mfg., LLC v. Vermeer Mfg. Co.</i> , 298 F.3d 1290 (Fed. Cir. 2002)	13
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	14
<i>Viasat, Inc. v. Kioxia Corp., et al.</i> , No. 6:21-cv-01231	2
<i>Viasat, Inc. v. Western Digital Corporation, et al.</i> , No. 6:21-cv-01230	3
<i>Western Digital Technologies, Inc. v. Viasat, Inc.</i> , IPR2022-01126, Paper 1, 13, (PTAB June 10, 2022)	3, 6, 7

STATUTES

35 U.S.C. §102	4
----------------------	---

35 U.S.C. §102(a)	24
35 U.S.C. §102(b)	22
35 U.S.C. §102(e)	19, 24
35 U.S.C. §103	4
35 U.S.C. § 112 ¶6	18
35 U.S.C. §316(a)(11).....	6, 10
OTHER AUTHORITIES	
M.P.E.P. §2141.03	17

LIST OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 8,615,700
1002	File History of U.S. Patent No. 8,615,700 (U.S. Patent Application No. 12/858,510)
1003	Declaration of Dr. Lara Dolecek with <i>Curriculum Vitae</i>
1004	U.S. Patent Application Publication No. 2011/0126079 to Wu <i>et al.</i>
1005	U.S. Patent Application Publication No. 2009/0063934 to Jo
1006	U.S. Patent Application Publication No. 2010/0070681 to Wan <i>et al.</i>
1007	U.S. Provisional Application No. 61/234,911
1008	Intentionally Omitted
1009	Intentionally Omitted
1010	Intentionally Omitted
1011	Intentionally Omitted
1012	Excerpt from Infringement Contentions in <i>Viasat, Inc. v. Western Digital Corporation, et al.</i> , No. 6:21-cv-01230 (W.D. Tex.)
1013	U.S. Patent Application Publication No. 2005/0138521 to Suzuki
1014	Mielke <i>et al.</i> , <i>Bit Error Rate in NAND Flash Memories</i> , IEEE CFP08RPS-CDR 46th Annual International Reliability, Physics Symposium, Phoenix, 2008
1015	Michelson <i>et al.</i> , <i>A 4Gb 2b/cell NAND Flash Memory with Embedded 5b BCH ECC for 36MB/s System Read Throughput</i> , ISSCC 2006 / SESSION 7 / NON-VOLATILE MEMORY / 7.6
1016	Chen <i>et al.</i> , <i>Error Correction for Multi-Level NAND Flash Memory Using Reed-Solomon Codes</i> , SiPS 2008
1017	Duann, <i>SLC & MLC Hybrid</i> , Flash Memory Summit, Santa Clara, Aug. 2008
1018	Duann, <i>Error Correcting Techniques for Future NAND Flash Memory in SSD Applications</i> , Flash Memory Summit, Santa Clara, Aug. 2009
1019	Scheduling Order dated May 10, 2022, in <i>Viasat, Inc. v. Kioxia Corp. et al.</i> , No. 6:21-cv-01231 (W.D. Tex.)
1020	Joint Notice to the Court Concerning Extension of Time

Kioxia Corporation and Kioxia America, Inc. (collectively, “Kioxia” or Petitioner”) respectfully requests *Inter Partes* Review (“IPR”) of claims 1, 2, 4, 8, 15, 17, and 19 (the “Challenged Claims”) of U.S. Patent No. 8,615,700 (Ex. 1001, “the ‘700 patent”), currently assigned—according to Patent Office records—to Viasat, Inc. (“Patent Owner”). There is a reasonable likelihood Petitioner will prevail on at least one challenged claim.

I. Introduction

The Challenged Claims are directed to an apparatus, system and method for correcting errors in flash memory. Flash memory is a form of solid-state, non-volatile memory (*i.e.*, it holds saved data even after power is turned off) whereby bits of information are stored as charge on the floating gate of a transistor, each transistor referred to as a memory cell.

Flash memory was well-known and in commercial use for more than a decade prior to the ‘700 patent. It was also well-known prior to the ‘700 patent that flash memory cells degrade with use, and over time, leading to occasional failure, and that a technique called error correction coding (“ECC”), including forward error correction coding (“FEC”), could be employed to detect and correct bits that have been misread from faulty memory cell(s). ECC/FEC involves encoding the data that is to be written and stored in memory, resulting in the addition of redundancy and/or parity bits to the data. During a subsequent decoding process, these additional ECC

bits are used to determine whether the data retrieved from memory contains errors and, if it does, to locate and correct those errors. The ‘700 patent acknowledges that ECC techniques themselves had been known for years, and states that its memory system embodiments are intended to employ one of these pre-existing ECC techniques.

The Challenged Claims recite an implementation of a flash memory system addressed to a particular physical structure and layout for performing ECC having distinct modules and submodules operating together to read and decode ECC encoded data stored in flash memory.

However, as will be explained in detail in the Grounds presented below and in the accompanying declaration of Dr. Lara Dolecek, there is nothing novel or non-obvious about the particular claimed arrangement of modules and submodules, or about the functions these components perform.

II. Mandatory Notices (37 C.F.R. §42.8)

A. Real Parties in Interest (37 C.F.R. §42.8(b)(1))

The real parties in interest are Kioxia Corporation and Kioxia America, Inc.

B. Related Matters (37 C.F.R. §42.8(b)(2))

The ‘700 patent has been asserted by Patent Owner against Petitioner in *Viasat, Inc. v. Kioxia Corp., et al.*, No. 6:21-cv-01231, currently pending in the United States District Court for the Western District of Texas (the “Litigation”).

The ‘700 patent has also been asserted by Patent Owner against Western Digital Technologies, Inc. and Western Digital Corp. (collectively “WD”) in *Viasat, Inc. v. Western Digital Corporation, et al.*, No. 6:21-cv-01230, currently pending in the United States District Court for the Western District of Texas.

On May 31, 2022, Petitioner filed an IPR challenging claims 1-4, 8, and 10-19 of the ‘700 Patent. *Kioxia Corporation, et al.*, IPR2022-01067, Paper 1 (PTAB May 31, 2022). The Kioxia Petition was instituted on November 29, 2022, IPR2022-01067, Paper 17.

On June 10, 2022, Western Digital Technologies, Inc. (“WDT”) filed an IPR petition challenging claims 1, 2, 4, 8, 15, 17, and 19 of the ‘700 Patent. *Western Digital Technologies, Inc.*, IPR2022-01126, Paper 1 (PTAB June 10, 2022) (hereinafter, the “WD Petition”). The WD Petition was instituted on November 29, 2022, IPR2022-01126, Paper 13. This Petition includes the same prior art and invalidity grounds as WDT’s instituted petition IPR2022-01126 and is accompanied by a Motion for Joinder. Petitioner filed this petition and Motion for Joinder within one month of institution of IPR2022-01126 as permitted by 37 C.F.R. §§ 42.122. Because this petition is accompanied by a Motion for Joinder under 35 U.S.C. § 315(c), the time-bar under Section 315(b) does not apply.

C. Counsel and Service Information (37 C.F.R. §§42.8(b)(3)(4))

Petitioner designates the following lead and back-up counsel:

Lead Counsel

Robinson Vu (Reg. No. 60,211) of Baker Botts L.L.P.

Backup Counsel

Michael A. Silliman (Reg. No. 75,112) of Baker Botts L.L.P.

Service Information: Service information is as follows: Baker Botts L.L.P., One Shell Plaza, 910 Louisiana Street, Houston, Texas 77002-4995; Tel. 713-229-1715; Fax 713-229-7815. Petitioner consents to service by electronic mail at michael.silliman@bakerbotts.com and robinson.vu@bakerbotts.com. A Power of Attorney is filed concurrently herewith under 37 C.F.R. § 42.10(b).

III. Payment of Fees (37 C.F.R. §42.103)

The Director is authorized to charge the filing fee for this Petition, as well as any other fees that may be required in these proceedings, to Deposit Account 02-0384.

IV. Requirements for *Inter Partes* Review (37 C.F.R. §42.104)

A. Grounds for Standing (37 C.F.R. §42.104(a))

Petitioner certifies that the ‘700 patent is available for IPR, and that Petitioner is not barred or estopped from requesting an IPR challenging the claims on the identified grounds.

B. Identification of Challenge (37 C.F.R. §42.104(b)(1)-(2)) and Relief Requested (37 C.F.R. §42.22(a)(1))

Petitioner requests the Board institute IPR on the Challenged Claims because they are anticipated and/or obvious under pre-AIA 35 U.S.C. §§102 and 103 on the following grounds:

Ground	Prior Art	Basis	Claims Challenged
1	U.S. Patent Application Publication No. 2011/0126079 (“Wu”)	§102	1, 4, 8, 15, 17,19
2	U.S. Patent Application Publication No. 2009/0063934 (“Jo”) in View of Wu	§103	1, 2, 4, 8, 15, 17, 19
3	Jo in View of U.S. Patent Application Publication No. 2010/0070681 (“Wan”)	§103	1, 2, 4, 8, 15, 17, 19

V. The Board Should Not Decline Review Under Section 314 or 325

This Petition includes the same prior art and invalidity grounds as instituted IPR2022-01126 filed by Western Digital Technologies, Inc., is accompanied by a Motion for Joinder, and has been timely filed under 37 C.F.R. §§ 42.122. Thus, this Petition is proper under 35 U.S.C. §§ 314(a) and 325(d).

A. General Plastic Factors

General Plastic does not apply here because the petitioner seeks to join an instituted IPR proceeding in an inactive role. *See General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 15-19 (PTAB Sept. 6, 2017); *Apple Inc. v. Uniloc 2017 LLC*, IPR2020-00224, Paper 10 at 4-5 (PTAB Apr. 6, 2020).

B. Fintiv Factors

In the WD IPR, the Board found that the *Fintiv* factors favored institution.

Western Digital Technologies, Inc. v. Viasat, Inc., IPR2022-01126, Paper 13 at 8-15 (PTAB Nov. 29, 2022). For the reasons explained in more detail below, the *Fintiv* factors weigh strongly in favor of institution of this Petition, and even more so because Petitioners merely seek to join an instituted IPR.

Factor 1. Whether a Stay Exists or Is Likely to Be Granted if a Proceeding Is Instituted

In the WD IPR, the Board found Factor 1 was neutral because no motion to stay has been filed yet. *See id.* at 8 (“We decline to speculate on how the district court would rule on a stay, if one were requested.”). Petitioner has likewise not yet filed a stay motion in the Litigation.

Factor 2. Proximity of the Court’s Trial Date to the Board’s Projected Statutory Deadline

In the WD IPR, the Board found that this factor weighs neither for nor against institution. *See id.* at 8-9. Here, the factor remains neutral as the trial in the Litigation may begin either shortly before or within a few months after the projected statutory deadline for a final written decision in this proceeding. *Id.* Patent owner has requested an October 2023 trial date in the Litigation, however, no trial date has yet been set. *See Ex. 1019* at 5. The projected statutory deadline for a final written decision in the WD IPR that Petitioner seeks to join is November 2023. IPR2022-01126, Paper 13 at 8.

Factor 3. Investment in the Parallel Proceeding by the Court and Parties

The Board's Institution Decision in the WD IPR states that "it appears that much work remains to be done in the parallel district court litigation as it relates to invalidity." IPR2022-01126, Paper 13, at 9. Kioxia submits that its Litigation has not progressed significantly further than the WD Litigation, as fact discovery only began on October 3, 2022 and Final Invalidity Contentions are not due until February 9, 2023. *See* Ex. 1019; Ex. 1020. The Board credited WD's diligence in filing the WD IPR, and Kioxia has also timely filed this Petition and Motion for Joinder. IPR2022-01126, Paper 13, at 10. As such, this factor favors institution, as the Board previously recognized. *Id.* ("Under these circumstances, we determine that this factor weighs slightly against exercising our discretion to deny institution.").

Factor 4. Overlap Between Issues Raised in the Petition and in the Parallel Proceeding

In the WD IPR, the Board found that this factor weighs against exercising discretion to deny institution. *Id.* at 11. In similar fashion, Kioxia hereby submits that if the Patent Trial and Appeal Board institutes inter partes review, it will not rely on the Grounds set forth in this Petition at its district court trial. As a result, this factor also weighs in favor of institution.

Factor 5. Whether the Petitioner and the Defendant in the Parallel Proceeding Are the Same Party

Petitioners are Defendants in the district court proceeding.

Factor 6. Other Circumstances that Impact the Board's Exercise of Discretion, Including the Merits

This Petition includes the same prior art and grounds as the instituted WD IPR. *Id.* at 6-7. Petitioners also submit that additional factors in favor of this institution include that this petition is merely a joinder to an instituted IPR, that Petitioners will take an inactive role (unless and until WD ceases to participate), and that trial is likely to proceed regardless of Petitioners' involvement.

On balance, Petitioners submit that the *Fintiv* factors weigh heavily in favor of granting this Petition and accompanying Motion for Joinder.

* * *

Petitioners submit that the factors and circumstances discussed above weigh strongly against the Board exercising its discretion to deny institution under § 325(d).

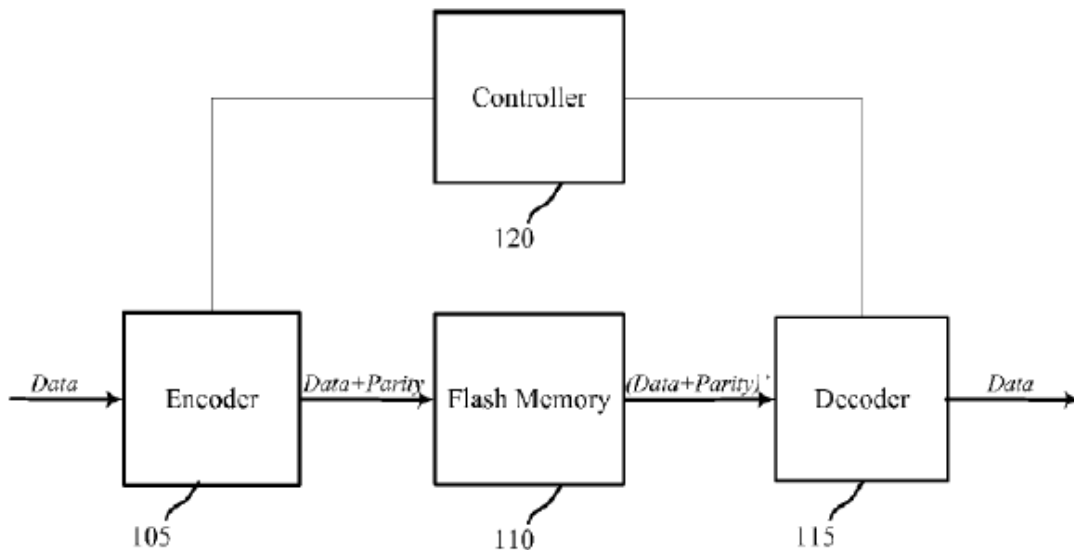
VI. Relevant Information Concerning the Contested Patent

A. Overview of the '700 Patent

The '700 patent relates to particular strategies for implementing forward error correction ("FEC") in flash memory and other types of solid state memory. Ex. 1001, 2:63-3:2, 3:23-24.

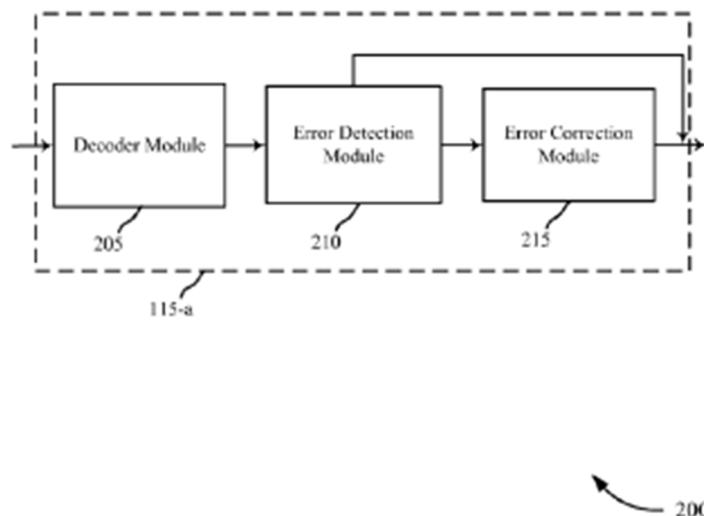
The patent explains that “[b]ecause of th[e] increase in density and the related issues in manufacturing ... the likelihood of data corruption a very real possibility. FEC may be one of the tools used to address these bit errors to improve the reliability and lifespan of flash memory.” *Id.*, 3:12-22.

Figure 1 provides a block diagram of a flash memory decoder system according to the patent’s embodiments:



Id., Fig. 1 (excerpt showing flash memory system 100); *see also id.*, 3:25-42.

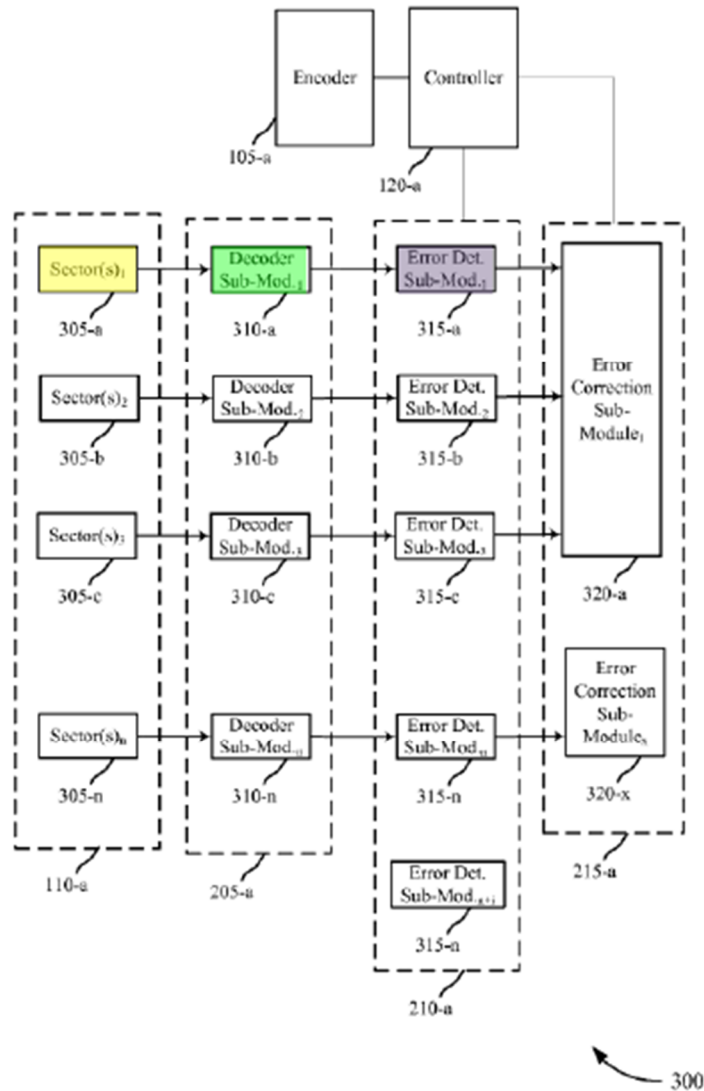
A block diagram of decoder 115 is shown in Figure 2 (below) and labeled 115-a.



Id., Fig. 2; *see also id.*, 5:47-50; 5:52-61.

The decoder module 205 may process received encoded data to generate a number of data streams. *Id.*, 6:9-11. In one embodiment, “the decoder module processor may buffer, de-interleave, or perform certain aspects of the decoding process, an[d] thereby generate decoded (or partially decoded) data streams.” *Id.*, 6:11-14. The error detection module 210 may include a number of error detection sub-modules operating in parallel. *Id.*, 6:14-16. The error correction module 215 may also “include a number of sub-modules (each of which may be time-shared by a number of error detection sub-modules).” *Id.*, 6:25-28.

Figure 3 (below) depicts a more detailed block diagram particularly illustrating error detection and correction sub-modules within the systems of Figures 1 and 2. *Id.*, 6:29-33.



Id., Fig. 3.¹

Referring to Figure 3, the ‘700 patent states that “encoded data may be read from a set of one or more of sectors 305 (e.g., sector(s) 305-a [yellow above]) by a decoder sub-module 310 (e.g., decoder sub-module 310-a [green above]), and processed (e.g., buffered, de-interleaved, decoded to some extent, etc.) to generate a

¹ Unless noted, all coloring and emphasis has been added.

data stream.” *Id.*, 6:46-50. The data stream may then be “forwarded to an error detection sub-module 315 (error detection sub-module 315-a [purple above]).” *Id.*, 6:50-52. The ‘700 patent states that “[e]ach error detection sub-module 315 may forward [a] portion of the respective received stream containing an error to the error correction module 215-a.” *Id.*, 6:63-7:5.

The controller 120-a shown in Figure 3 may “adapt the coding rate to be used by the encoder responsive to a monitored age or error rate associated with the flash memory 110-a.” *Id.*, 7:21-24; *see also id.*, 4:63-5:7; *see also* Ex. 1003 ¶¶26-40.²

B. The ‘700 Patent’s Priority Date

The application for the ‘700 patent was filed August 18, 2010, and claims priority to U.S. Provisional Application No. 61/234,911 (the “‘911 provisional,” Ex. 1007) filed August 18, 2009. Ex. 1002, 1, 32.

Patent Owner bears the burden to establish that every claim limitation of the challenged claims is described in an earlier filed application. *See Maxlite, Inc. v. Jiaying Super Lighting Elec. Appliance Co.*, IPR2020-00208, Paper 14, 6-8 (PTAB Jun. 1, 2021) (to antedate prior art, Patent Owner bears the burden to establish every

² Declaration of Dr. Lara Dolecek, an expert in Error Correction Coding (“ECC”) algorithms and applications—including flash memory applications. Ex. 1003 ¶¶7-12; *see also* CV attached to same.

claim limitation is described in earlier-filed application); *see also New Railhead Mfg., LLC v. Vermeer Mfg. Co.*, 298 F.3d 1290, 1294-95 (Fed. Cir. 2002).

Patent Owner cannot meet this burden, and, in district court, Patent Owner does not assert that the ‘700 patent is entitled to claim priority to the ‘911 provisional. Ex. 1012, 4.

The ‘700 patent is not entitled to claim priority to the ‘911 provisional at least because the provisional does not disclose the “decoding module,” “error detection module,” “error detection sub-module,” or “error correction module.” *See generally* Ex. 1007; *compare* Ex. 1007, Figs. 1-4 *with* Ex. 1001, Figs. 1-7; Ex. 1003 ¶¶48-52.

Accordingly, the ‘700 patent is entitled to a priority date no earlier than August 18, 2010.

C. Overview of the ‘700 Patent Prosecution History

In an office action dated January 31, 2013, the examiner made a non-final rejection of then pending claims 1, 2, 13-15, and 17 as anticipated by U.S. Patent Application Publication No. 2005/0138521 (“Suzuki”; Ex. 1013).

Applicant overcame the rejections without substantive amendment (Ex. 1002, 171-173), and the pending claims were subsequently allowed (Ex. 1002, 182).

D. Person of Ordinary Skill in the Art

Several factors may be considered in determining the proper skill level. M.P.E.P. §2141.03.

Here, the level of skill is apparent from the cited art. *See In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995). Petitioner submits that a person of ordinary skill in the art (“POSA”) would have had at least a B.S. in electrical engineering or computer science or a related field, and four years’ experience designing, developing, or researching error correction coding (“ECC”) systems (which include forward error correction or “FEC” systems). Ex. 1003 ¶¶23-25, 66. This description is approximate—a higher education or skill level might make up for less experience, and vice-versa. *Id.*

E. Claim Construction (37 C.F.R. §42.104(b)(3))

This Petition construes terms consistent with the understanding a POSA would have had, at the time of the invention, in view of the ‘700 patent’s intrinsic evidence and, where appropriate, extrinsic evidence. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005).

Any implicit constructions derived from Petitioner’s analysis of the prior art are for the purposes of this IPR only, and may be based in-part on Patent Owner’s infringement allegations.

Further, by identifying disclosure in the prior art as meeting certain claim limitations Petitioner merely asserts that the scope of such limitation, even if indefinite, and if not limited by 35 U.S.C. § 112 ¶6, would at least include the identified disclosure.

VII. The Specific Grounds of Unpatentability

As evidenced by the prior art and the declaration of Dr. Lara Dolecek (Ex. 1003), ECC and FEC techniques were commonly employed prior to the ‘700 patent. Ex. 1003 ¶¶57-67; Ex. 1014 (single bit ECC); Ex. 1017 (BCH codes); Ex. 1015 (same); Ex. 1016 (Reed-Solomon codes); Ex. 1018 (LDPC codes). It was well-known that these related technologies were well-suited for correcting errors in data retrieved from non-volatile memory, including flash memory, so as to improve memory reliability and extend memory lifetime. Ex. 1003 ¶¶57-66. The prior art also teaches the claimed decoder configurations, rendering the Challenged Claims anticipated and/or obvious.

A. Prior Art

No prior art relied upon was cited during prosecution.

1. Wu

Wu, entitled “Multi-channel memory apparatus and method thereof,” published May 26, 2011 and was filed November 24, 2009. Ex. 1004. It is prior art under 35 U.S.C. §102(e).

Wu is directed to a multi-channel memory apparatus utilizing an efficient ECC encoding and decoding scheme. Ex. 1004 ¶0002; Ex. 1003 ¶70.

Figure 1 of Wu (below) shows a multi-channel memory apparatus 100 according to one embodiment. Ex. 1004 ¶0022; Ex. 1003 ¶¶73-74.

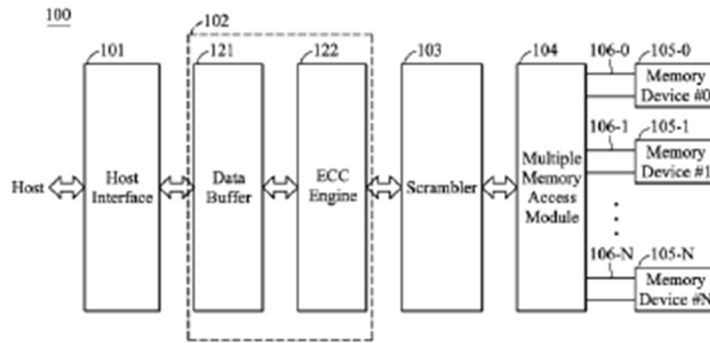


FIG. 1

Multi-channel memory apparatus 100 “comprises a host interface 101, an error correcting module 102, a scrambler 103, a multiple memory access module 104, a plurality of storage channels and a plurality of memory devices 105-0, 105-1 to 105-N. Each storage channel 106-0 to 106-N is coupled to a memory device (such as memory device 105-0, 105-1 to 105-N) for storing the data. The host interface 101 is arranged to communicate with the host device,” *e.g.*, a computing or media device such as a cellular phone. Ex. 1004 ¶0022; Ex. 1003 ¶74. The memory devices 105-0 to 105-N “may be a NAND or NOR type flash, a solid state disk ... etc.” Ex. 1004 ¶0024; Ex. 1003 ¶71.

Wu further discloses that “[t]he error correcting module 102 is shared by the storage channels and is arranged to perform error correction code encoding on the data to be stored into the memory devices and perform error correction code decoding on the data read out from the memory devices, depending on the data

transferring directions (e.g. write or read directions).” Ex. 1004 ¶¶0022; Ex. 1003 ¶71.

Figure 8 of Wu (below) discloses a modification to the multi-channel memory apparatus of Figure 1 further comprising a plurality of ECC engines 822-0, 822-1 to 822-M coupled in parallel. Ex. 1004 ¶¶0033; Ex. 1003 ¶¶70-72.

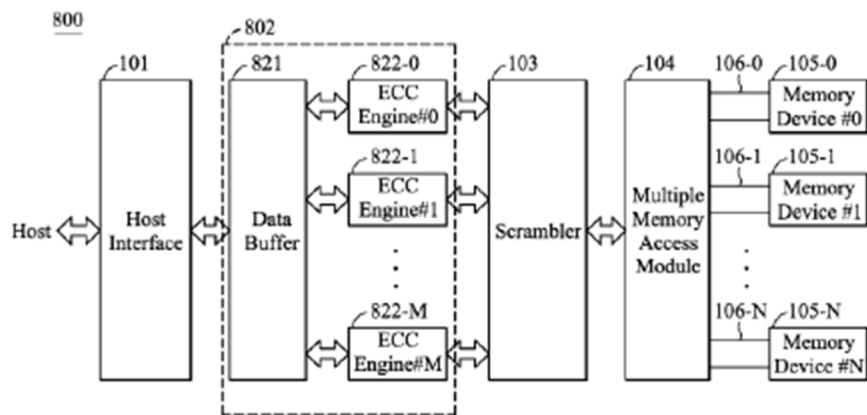


FIG. 8

Wu further discloses that each of these ECC engines 822-0 to 822-M “may comprise one or more error detectors 902-0 to 902-K and an error corrector 901,” as shown in Figure 9 below. Ex. 1004 ¶¶0034; Ex. 1003 ¶¶91-92.

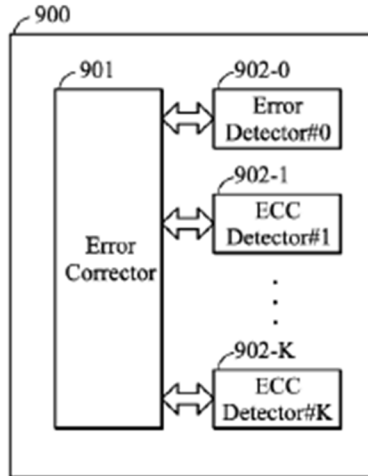


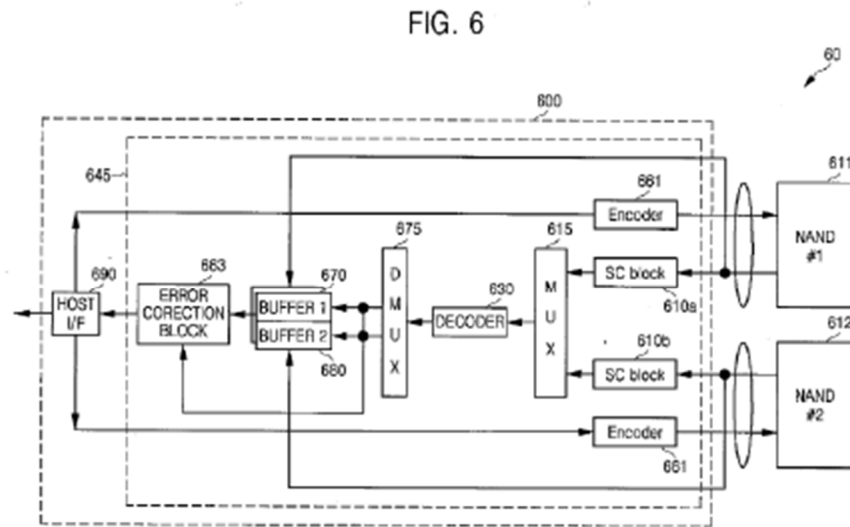
FIG. 9

2. Jo

Jo, entitled “Multi-channel memory system including error correction decoder architecture with efficient area utilization,” published March 5, 2009. Ex. 1005. It is prior art under 35 U.S.C. §102(b).

Jo explains that “[t]here is a trend for flash memory systems to have more and more memory devices [and] to employ multi-level cell (MLC) NAND memory devices for increased storage capacity.” Ex. 1005 ¶0009; Ex. 1003 ¶185. Jo further explains that, as a result of this increased capacity, flash memory systems require more and more ECC modules, and that this enlarges the size of the integrated circuit, and increases the complexity and cost of the flash memory system. Ex. 1005 ¶0009; Ex. 1003 ¶185. Accordingly, Jo discloses a memory system “with a more efficient utilization of area and circuitry in a memory controller.” Ex. 1005 ¶0010; Ex. 1003 ¶185.

Figure 6 of Jo (below) shows an exemplary “two channel flash memory system 60 that can operate in a pipelined mode with a buffer memory structure. Flash memory system 60 includes a flash memory controller 600 and NAND memory devices 611 and 612. Flash memory controller 600 communicates data with memory devices 611 and 612 over two communication channels operating in parallel.” Ex. 1005 ¶¶0036; Ex. 1003 ¶¶186-189.



As Figure 6 shows, “ECC block 645 includes two data encoders 661, one each for the first and second communication channels. ECC block 645 also includes two syndrome computation blocks 610a and 610b, ... one for each channel[.]. ECC block 645 further includes a multiplexer (or data switch) 615, a decoder block 630, a demultiplexer 675, first and second memory buffers 670 and 680, third and fourth memory buffers (not labeled), and an error correction block 663.” Ex. 1005 ¶¶0037; Ex. 1003 ¶¶138-140.

Jo further discloses that data written to the memory devices “is first encoded (e.g. with a BCH code) by a corresponding encoder 661.” Ex. 1005 ¶¶0038; Ex. 1003 ¶141. “When the encoded data is to be read ... it is decoded before being sent to a host device (e.g., a processor).” Ex. 1005 ¶¶0039; Ex. 1003 ¶141.

Jo discloses that errors are detected by the syndrome computation blocks. Syndrome computation involves taking the product, *i.e.*, multiplying, a parity check matrix and a vector of retrieved data to generate a set of “syndrome values.” Ex. 1005 ¶¶0040; Ex. 1003 ¶¶63, 143. Jo discloses that “[i]f the syndrome values for a set of data are zero ...the data set does not include errors. Otherwise, then the data set includes one or more errors.... That is, the syndrome computation blocks 610a and 610b detect whether or not a data set includes errors.” Ex. 1005 ¶¶0040; Ex. 1003 ¶¶143, 159.

3. Wan

Wan, entitled “Method for scrambling data in which scrambling data and scrambled data are stored in corresponding non-volatile memory locations,” published March 18, 2010 and was filed September 12, 2008. Ex. 1006. It is prior art under 35 U.S.C. §102(a) and §102(e).

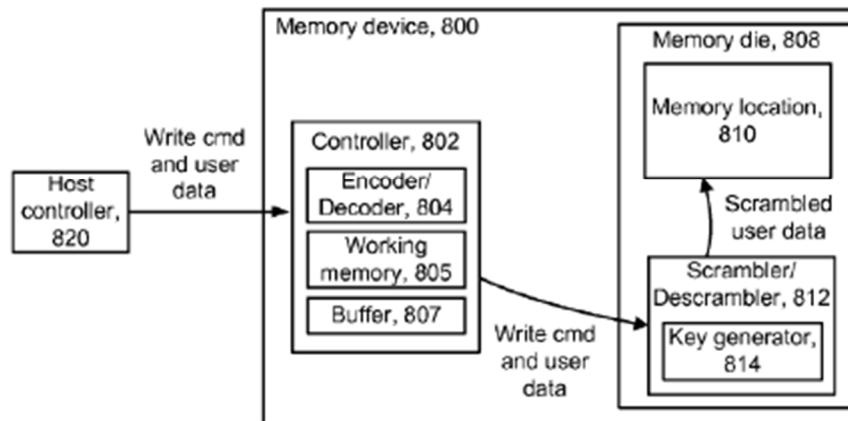
Wan is directed to systems and methods for improving the reliability of NAND flash memory by using data scrambling and descrambling techniques related to those disclosed in Wu. Wan observes that “[d]ata scrambling for non-volatile

memory such as NAND memories has become increasingly important in addressing system reliability issues that are inherently susceptible by design to worst-case data patterns [including fixed repetitive data,] which trigger memory program failures.” Ex. 1006 ¶¶0049; Ex. 1003 ¶225. Wan explains that “[d]ata scrambling randomizes the distribution of data states in the storage elements so that worst-case data patterns are avoided.” Ex. 1006 ¶¶0049; Ex. 1003 ¶225.

Wan notes that “scrambling is considered to be a form of encoding and descrambling is considered to be a form of decoding,” and describe examples “in which random or pseudorandom data is used to encode or decode user data.” Ex. 1006 ¶¶0052; *see also id.*, ¶¶0069 (“the controller decodes, e.g., descrambles, the user data”); Ex. 1003 ¶228.

In one of the flash memory system embodiments disclosed by Wan, and depicted in Figure 8a below, scrambling is “performed by on-chip circuitry using a key generator in a storage system.” Ex. 1006 ¶¶0070; Ex. 1003 ¶¶213-216.

Fig. 8a



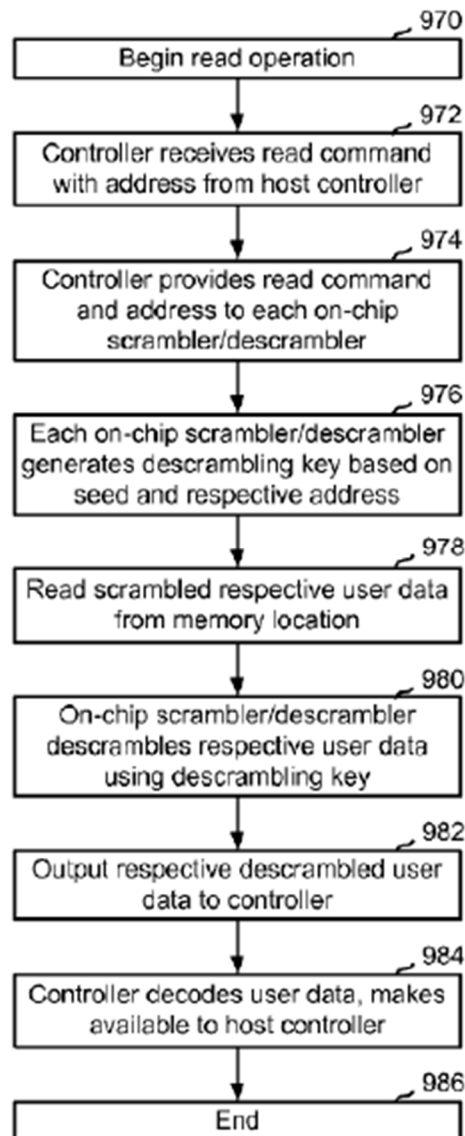
Wan states “[t]his approach provides portability of the memory die so that it can be used with different controllers, for instance, which are made by different manufacturers.” Ex. 1006 ¶¶0070; Ex. 1003 ¶¶216.

According to Wan, to store user data in the memory array 810, “controller 802 may process the user data such as by performing ECC coding and adding overhead data at an encoder/decoder 804.... [T]he overhead data is also scrambled with the user data. This helps avoid header and ECC type data pattern failures.” Ex. 1006 ¶¶0070; Ex. 1003 ¶¶216.

Wan describes the process of reading user data from memory array with reference to Figure 8b, which shows the same structures as Figure 8a above. Wan states, during a read operation, that “scrambler/descrambler 812 may use the ... keys which were used for scrambling.... The descrambled user data is then provided to the controller 802.” Ex. 1006 ¶¶0071; Ex. 1003 ¶¶221.

Wan depicts the descrambling process in detail in Figure 10b (below).

Fig. 10b



See also Ex. 1006 ¶0081 (describing Figure 10b); Ex. 1003 ¶222.

B. Ground 1: Wu Anticipates Claims 1, 4, 8, 15, 17, and 19

1. Claim 1

1[preamble] “A flash memory decoder comprising”

Regardless of whether the preamble is limiting, it is disclosed by Wu. Ex. 1003 ¶¶69-77.

Wu’s Figure 8 (below) discloses a multi-channel memory apparatus comprising a plurality of ECC engines 822-0, 822-1 to 822-M, scrambler 103, multiple memory access module 104, and memory devices 105-0 to 105-N, each functioning as described with reference to Wu’s earlier Figures 1 and 2. Ex. 1004 ¶0033; Ex. 1003 ¶73. The memory devices “may be a NAND or NOR type **flash**.” Ex. 1004 ¶0024; Ex. 1003 ¶¶70-73.

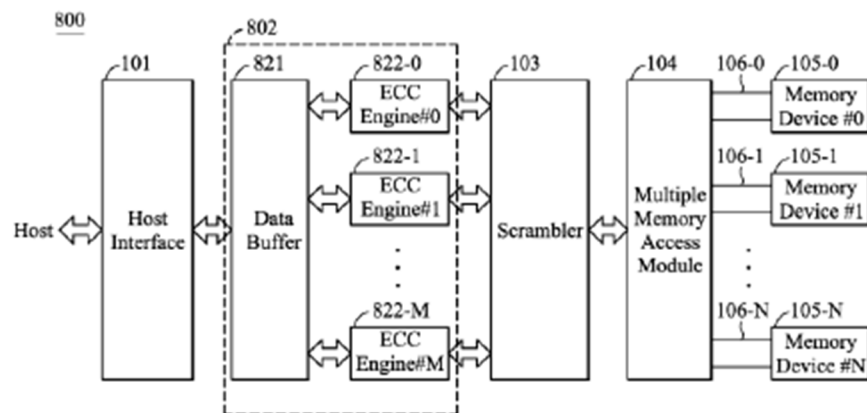


FIG. 8

A “host interface 101 may interpret host commands and receive data from or transmit data to the host device depending on the data transferring directions (e.g. write or read directions).” Ex. 1004 ¶0022; Ex. 1003 ¶74. The plurality of ECC engines 822-0, 822-1 to 822-M perform error correction code encoding and decoding on the data stored in the memory devices depending on the data transferring directions. Ex. 1004 ¶¶0033 (stating that ECC engines 822-0, 822-1 to 822-M are

“for performing error correction code encoding and decoding on the data to be stored in the memory devices” and referring back to corresponding descriptions of Figure 1 depicting error correction module 102), 0022 (“The error correcting module 102 is shared by the storage channels and is arranged to perform error correction code encoding on the data to be stored into [and] error correction code decoding on the data read out from the memory devices, depending on the data transferring directions....”); Ex. 1003 ¶¶72-75.

“[S]crambler 103 may be implemented in a scrambler module (not shown) and is arranged to perform intra-channel disarrangement for the data to be stored into the memory devices and rearrange the scrambled data read out from the memory devices in a reverse manner depending on the data transferring directions.” Ex. 1004 ¶¶0023, 0033; Ex. 1003 ¶¶76-77.

Wu’s “flash memory” (yellow) “decoder” (green) is identified in Figure 8 below. Ex. 1003 ¶77.

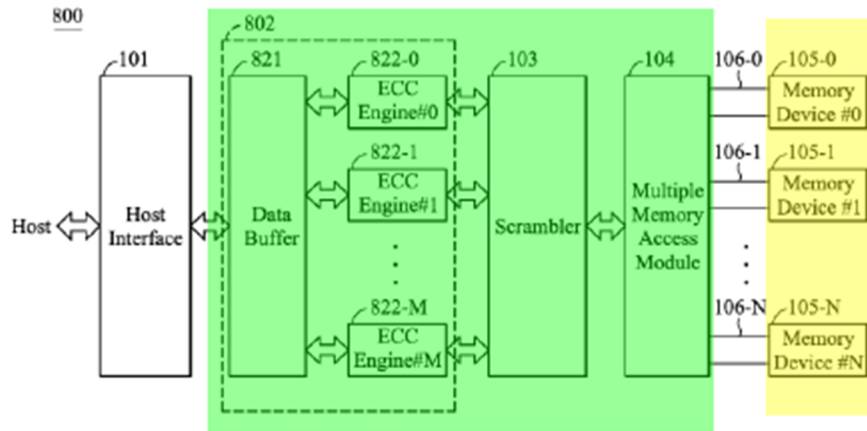


FIG. 8

1[a] “a decoding module configured to: receive encoded data from the flash memory; and decode the received encoded data to generate a plurality of partially decoded data streams;”

As shown in annotated Figure 8 above, Wu’s decoder includes a “**decoding module**,” *i.e.*, the combination of the multiple memory access module 104 and scrambler 103, which may itself “be implemented in a scrambler module.” Ex. 1004 ¶¶0023-0026, 0033, Figs. 1, 2, 8; Ex. 1003 ¶79.

As shown in Figure 8, multiple memory access module 104 is coupled to storage channels 106-0 to 106-N, each storage channel in turn being coupled to a memory device for storing data. Ex. 1004 ¶0022, Fig. 8; Ex. 1003 ¶80.

When reading the memory devices, “the multiple memory access module 104 multiplexes the data received simultaneously from different memory devices 105-0 to 105-N.” Ex. 1004 ¶0024; *see also id.*, ¶0027, Fig. 3, ¶0029 (“The multiple memory access module first schedules the flow queue of the read operation and reads data from the memory device (Step S501).”), Fig. 5; Ex. 1003 ¶80.

Wu further discloses that the error correction encoded data (Ex. 1004 ¶¶0022, 0033) retrieved from memory devices over channels 106-0 to 106-N is intra-channel disarranged by a scrambler (Ex. 1004 ¶0023), and inter-channel disarranged by multiple memory access module 104 (Ex. 1004 ¶0027). Ex. 1003 ¶81. These disclosures indicate that the data stored on the flash memory devices 105-0 to 105-N is encoded data. Ex. 1004 ¶0028, Fig. 4; Ex. 1003 ¶¶81-82. Thus, when reading from the memory devices, the “**decoding module**,” *i.e.*, the multiple memory access module 104 and scrambler 103, is “**configured to: receive encoded data from the flash memory.**” Ex. 1003 ¶83.

Wu further discloses that the “**decoding module**” is “**configured to: ... decode the received encoded data to generate a plurality of partially decoded data streams.**” In particular, Wu discloses that the multiple memory access module 104 may include a “data rearranging unit 302 ... arranged to perform inter-channel disarrangement for the data received from the multiplexing unit and to be stored into the memory devices, *and rearrange the data read out from memory devices through the storage channels.*” Ex. 1004 ¶0027; Ex. 1003 ¶84. “The inter-channel disarrangement is to further enhance the disorderliness of the data [and reduce inter-channel interference] by exchanging the data of different storage channels so that the same type of data may not all be stored in the same memory device.” Ex. 1004 ¶0027; Ex. 1003 ¶84.

Wu further discloses that scrambler module 103 may “perform intra-channel disarrangement for the data to be stored into the memory devices and *rearrange the scrambled data read out from the memory devices* in a reverse manner depending on the data transferring directions.” Ex. 1004 ¶¶0023; Ex. 1003 ¶85. The intra-channel disarrangement/arrangement operation is performed to scramble/unscramble the data of the ECC engine codewords, the error correction coded data, so as to randomly distribute the data ‘0’ and ‘1’ in adjacent locations thereby further reducing interference. Ex. 1004 ¶¶0023; Ex. 1003 ¶85.

As shown in Wu’s Figure 8, the data retrieved from memory devices 105-0 to 105-N and provided to ECC Engines 822-0 to 822-M constitutes disclosure of the claimed “**plurality of partially decoded data streams**” and are shaded **yellow** in Wu’s Figure 8 below.³ Ex. 1003 ¶86. The “partial decoding” is disclosed by the intra-channel rearrangement performed by scrambler 103. Ex. 1003 ¶¶85-86; *see also* Ex. 1006 ¶¶0052 (further explaining that “scrambling is considered to be a form of encoding and descrambling is considered to be a form of decoding”), 0062.

³ While only (M+1) data streams are visible in Figure 8, one for each ECC engine 822-0 to 822-M, as will be further explained with respect to limitation 1[c] below, each of these data stream is further subdivided into (K+1) sub-streams, for a total of (K+1)×(M+1) “partially decoded data streams.” Ex. 1003 ¶¶87-88, 97-104.

Alternatively, the partial decoding is disclosed by the inter-channel rearrangement performed by multiple memory access module 104, a claim interpretation advanced by Patent Owner in the related Litigation (claiming that de-interleaving can be partial decoding). Ex. 1003 ¶¶87-88.⁴

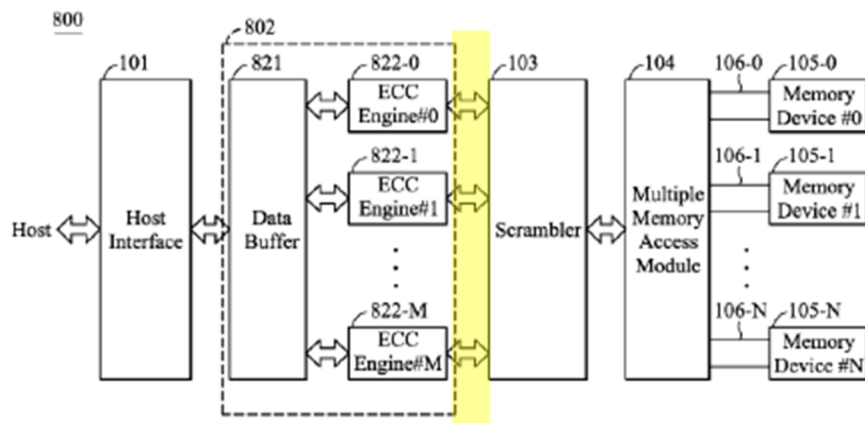


FIG. 8

Ex. 1004, Fig. 8.

1[b] “an error detection module communicatively coupled with the decoding module, and comprising a plurality of error detection sub-modules operating in parallel,”

Wu’s Figure 8 shows “a plurality of ECC engines 822-0, 822-1 to 822-M.”

Ex. 1004 ¶¶0033; Ex. 1003 ¶¶91. Wu further discloses that *each of these* ECC engines “may comprise one or more error detectors 902-0 to 902-K and an error corrector 901,” as shown in Figure 9 below. Ex. 1004 ¶¶0034; *see also id.* (“[T]he ECC

⁴ Patent Owner confidentially designated its contentions. If Patent Owner disputes this, Petitioner respectfully reserves the right to submit the contentions.

engine/sub-ECC engine 900 shown in FIG. 9 may also be applicable to ... the ECC engine/sub-ECC engines 822-0 to 822-M as shown in FIG. 8.”); Ex. 1003 ¶¶91-92.

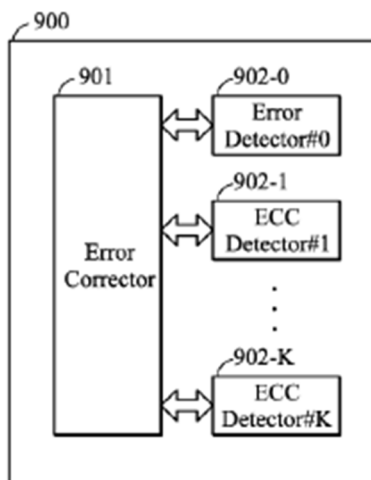


FIG. 9

In other words, each ECC Engine 822-0, 822-1 to 822-M comprises multiple ECC Detectors 902-0 to 902-K. Ex. 1003 ¶92.

Wu further discloses that “the ECC engine (such as the ECC engine ... 822-0 to 822-M and 900 ... etc.) is arranged to perform suitable ECC algorithm, such as Reed-Solomon or Bose, Ray-Chaudhuri, Hocquenghem (BCH) encoding and decoding,” and that “[t]he error corrector 901 is shared by the error detectors 902-0 to 902-K and arranged to correct the error(s) of the data detected by the error detectors.” Ex. 1004 ¶0034; Ex. 1003 ¶93.

Wu’s error detectors 902-0 to 902-K, which are disclosed as residing within each ECC engine 822-0 to 822-M, as shown in yellow in annotated Figure 8 below, collectively constitute disclosure of the claimed “**error detection module**

communicatively coupled with the decoding module, and comprising a plurality of error detection sub-modules operating in parallel.” Ex. 1003 ¶¶94-95.⁵

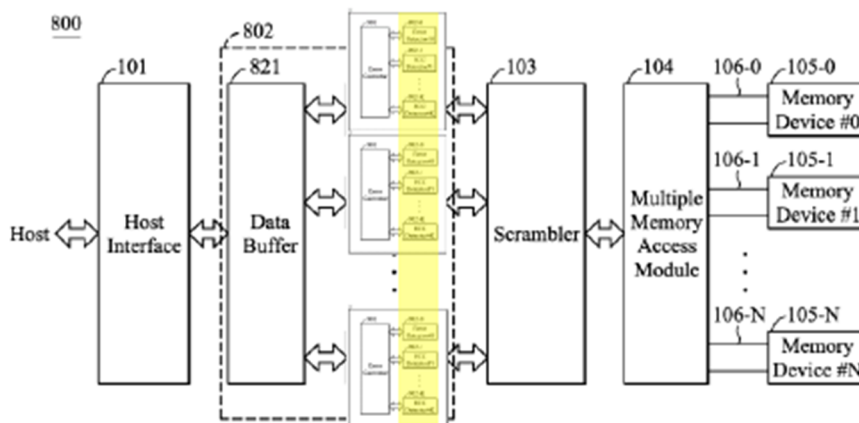


FIG. 8

Ex. 1004, Fig. 8 (annotated by replacing ECC engines 822-0 to 822-M with the more detailed ECC engine structure provided in Wu’s Figure 9).

Given the above disclosures, a POSA would have understood that Wu’s error detectors 902 operate in parallel. Ex. 1003 ¶95. That is the reason such modules are created and arranged in parallel. Ex. 1003 ¶95. A POSA would not understand these modules as alternatively working one-at-a-time, as this would detrimentally impact performance (*e.g.*, by delaying memory access times) and would render the inclusion

⁵ It is worth noting that claim 14 of the ‘700 patent depends from claim 1 and adds “wherein ... the encoded data is encoded with a Hamming code, a BCH code....” Ex. 1001, cl. 14. Therefore, claim 1 expressly extends to BCH decoding, such as disclosed by Wu. Ex. 1004 ¶0034.

of multiple error detectors, as taught by Wu, an unnecessary cost in terms of circuitry, complexity, and space, because only one detector would be needed if the detectors do not operate in parallel. Ex. 1003 ¶95.

1[c] “each error detection sub-module configured to: receive a different one of the plurality of partially decoded data streams; detect whether a portion of the respective received stream contains an error; and forward the portion of the respective received stream containing an error to an error correction module; and”

As noted above, in Wu’s flash memory decoder embodiment in which the ECC engines 822-0 to 822-M of Figure 8 are implemented using the ECC engine structure shown in Figure 9, the error detectors 902-0 to 902-K within each ECC engine are arranged in parallel. Ex. 1003 ¶98. Each of these (K+1) error detectors within each ECC engine, *i.e.*, each “error detection sub-module,” receives (on a memory read operation) a portion of one of the (M+1) partially decoded data streams provided at the output of the “decoder module,” *i.e.*, at the output of scrambler 103. Ex. 1003 ¶98. These portions of partially decoded data streams are themselves the claimed (K+1)×(M+1) “plurality of partially decoded data streams.” Ex. 1003 ¶98.

Wu further discloses that “The error detectors 902-0 to 902-K are arranged to ... detect error(s) in the data read out from the memory devices.” Ex. 1004 ¶0034; Ex. 1003 ¶99.

Wu therefore discloses **“each error detection sub-module configured to: receive a different one of the plurality of partially decoded data streams; [and]**

detect whether a portion of the respective received stream contains an error.”

Ex. 1003 ¶100.

Wu further discloses that the error correctors 901 are “shared by the error detectors 902-0 to 902-K and arranged to correct the error(s) of the data detected by the error detectors.” Ex. 1004 ¶0034; Ex. 1003 ¶101.

Data containing errors is sent to the error correctors 901, which perform, *e.g.*, “decoding ... to correct the error(s) of the data detected by the error detectors.” Ex. 1004 ¶0034, Fig. 9; Ex. 1003 ¶101.

Therefore, Wu further discloses **“each error detection sub-module configured to: ... forward the portion of the respective received stream containing an error to an error correction module.”** Ex. 1003 ¶102.⁶

1[d] “the error correction module, communicatively coupled with and physically separate from the error detection module, and configured to correct the received portions of the respective received streams containing an error.”

⁶ Limitation 1[d] does not preclude error-free data from also being forwarded to the error corrector 901; this forwarding operation is only prohibited in dependent claim 13, which depends from claim 1 and further recites “wherein each error detection sub-module is configured to direct error-free portions of the respective received streams to bypass the error correction module.” Ex. 1003 ¶103.

Wu discloses that each of ECC engines 822-0, 822-1 to 822-M “may comprise one or more error detectors 902-0 to 902-K and an error corrector 901,” as shown in the “block diagram” of Figure 9 below. Ex. 1004 ¶0034; Ex. 1003 ¶106.

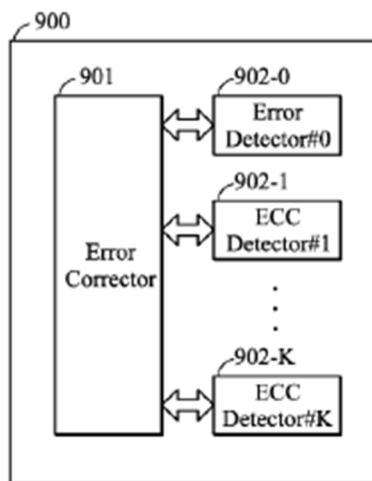


FIG. 9

Wu further discloses that “the ECC engine (such as the ECC engine ... 822-0 to 822-M and 900 ... etc.) is arranged to perform suitable ECC algorithm, such as Reed-Solomon or ... (BCH) encoding and decoding,” and that “[t]he error corrector 901 is shared by,” and therefore distinct and physically separate from, “the error detectors 902-0 to 902-K and arranged to correct the error(s) of the data detected by the error detectors.” Ex. 1004 ¶0034; *see also id.* claim 10 (“The ... apparatus as claimed in claim 1, wherein the ECC engine comprises ... a plurality of error detectors ... and an error corrector, shared by the error detectors); Ex. 1003 ¶107.

Wu’s error correctors 901 residing within each ECC engine 822-0 to 822-M shown in the “multi-channel memory apparatus” of Figure 8 (Ex. 1004 ¶0019)

collectively constitute disclosure of the claimed **“the error correction module,”** shown in **yellow** in annotated Figure 8 below (showing physical components of Wu’s multi-channel memory apparatus), **“communicatively coupled with and physically separate from the error detection module, and configured to correct the received portions of the respective received streams containing an error.”**

Ex. 1003 ¶108.

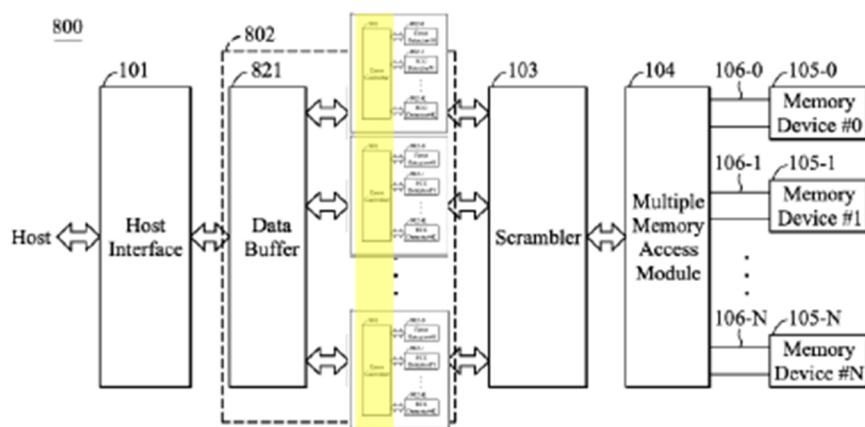


FIG. 8

Ex. 1004, Fig. 8 (annotated to show the ECC engine details shown in Figure 9); *see also id.* ¶¶0022, 0033.

2. Claim 4

“The flash memory decoder of claim 1, wherein the error correction module comprises a plurality of error correction sub-modules operating in parallel and fewer in number than the error detection sub-modules.”

Wu’s annotated Figure 8 (above and replicated below) discloses the **“flash memory decoder of claim 1, wherein the error correction module,”** **yellow**,

“comprises a plurality of error correction sub-modules,” error correctors 901 indicated by red boxes, “operating in parallel and fewer in number than the error detection sub-modules,” detectors 902 green. Ex. 1004 ¶¶10034; Ex. 1003 ¶¶112-114.

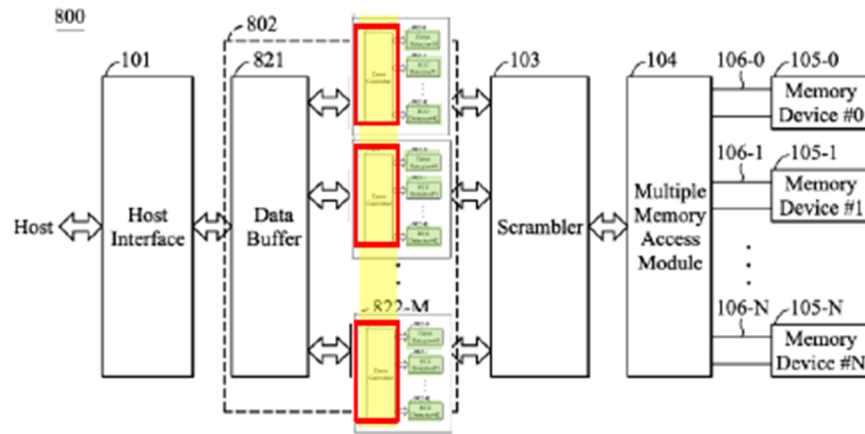


FIG. 8

Ex. 1004, Fig. 8 (annotated to show the ECC engine details shown in Figure 9).

3. Claim 8

8[a] “The flash memory decoder of claim 4, wherein, each error correction sub-module is assigned a set of two or more error detection sub-modules different from error detection sub-modules associated with other of the error correction sub-modules; and”

Wu’s Figure 8, as annotated in §VIII.B.2 above to include the exemplary ECC Engine 822 structure shown in Wu’s Figure 9, discloses “**each error correction sub-module,**” error corrector modules 901 within ECC Engines 822-0- to 822-M, “**is assigned a set of two or more error detection sub-modules,**” (k+1) error detector modules 902 within each ECC Engines 822-0- to 822-M “**different from**

error detection sub-modules associated with other of the error correction sub-modules.” Ex. 1003 ¶¶116-118.

8[b] “each error correction sub-module is configured to receive portions of the respective received streams containing an error detected in parallel by the set of two or more error detection sub-modules assigned to the respective error correction sub-module.”

Wu’s Figure 8 flash memory decoder embodiment includes ECC engines 822-0 to 822-M “coupled in parallel.” Ex. 1004 ¶0033; Ex. 1003 ¶120. Wu further discloses that each ECC engine 822-0 to 822-M can be implemented using the structure shown in Figure 9, which includes an ECC corrector connected to error detectors 902-0 to 902-K further arranged in parallel. Ex. 1004 ¶0034; Ex. 1003 ¶120.

Wu states that “[t]he error corrector 901 is shared by the error detectors 902-0 to 902-K and arranged to correct the error(s) of the data detected by the error detectors.” Ex. 1004 ¶0034; Ex. 1003 ¶121. These teachings establish that Wu discloses “**each error correction sub-module,**” error corrector modules 901 within ECC Engines 822-0 to 822-M, “**is configured to receive portions of the respective received streams containing an error detected in parallel by the set of two or more error detection sub-modules assigned to the respective error correction sub-module,**” (k+1) error detector modules 902 within each ECC Engines 822-0 to 822-M. Ex. 1003 ¶¶122-123; *see also* §VIII.B.1 (limitation 1[b]) (further identifying

Wu's disclosure that error detector modules 902 operate so as to detect errors in parallel).

4. Claims 15 and 17

In significant part, independent claims 15 and 17 recast apparatus claim 1 in system (claim 15) and method (claim 17) format. Ex. 1003 ¶¶125-126. Therefore, claims 15 and 17 are disclosed and anticipated by Wu for the same reason that Wu discloses and anticipates claim 1 (it discloses the claimed apparatus, system and method). *See* §VIII.B.1 [claim 1]; Ex. 1003 ¶¶125-134.

The table below provides a cross-reference showing where disclosure of the limitations of claims 15 and 17 can be found in corresponding claim 1 limitations. Ex. 1003 ¶127. As seen in the table, only limitations 15[a] and 15[b] are not equivalently recited in claim 1. Ex. 1003 ¶127. Analysis of these two limitations is provided below.

Claim 15	Claim 17	Claim 1
15[preamble] A system comprising: 15[a] an encoder configured to encode data for storage on a flash memory;		
15[b] the flash memory, communicatively coupled with the encoding module, and		

Claim 15	Claim 17	Claim 1
configured to store the encoded data; and		
15[c] a decoder, communicatively coupled with the flash memory, and configured to: 15[d] retrieve the encoded data from flash memory to generate a plurality of data streams; and	17[pre] A method comprising: 17[a] receiving encoded data from a flash memory comprising a plurality of data streams;	1[preamble] A flash memory decoder comprising: 1[a] a decoding module configured to: receive encoded data from the flash memory; and decode the received encoded data to generate a plurality of partially decoded data streams;
15[e] process, at each of a plurality of error detection sub-modules of the decoder operating in parallel, a different one of the plurality of data streams,	17[b] processing, at each of a plurality of error detection sub-modules operating in parallel, a different one of the plurality of data streams,	1[b] an error detection module communicatively coupled with the decoding module, and comprising a plurality of error detection sub-modules operating in parallel, 1[c] each error detection sub-module configured to: receive a different one of the plurality of partially decoded data streams;...
15[f] wherein each error detection sub-module is configured to: detect whether a portion of the respective received stream contains an error; and forward the portion	17[c] wherein each error detection sub-module is configured to: detect whether a portion of the respective received stream contains an error; and forward the portion	1[c] each error detection sub-module configured to:... detect whether a portion of the respective received stream contains an error; and forward the portion of the respective

Claim 15	Claim 17	Claim 1
of the respective received stream containing an error to an error correction module of the decoder; and	of the respective received stream containing an error to an error correction module; and	received stream containing an error to an error correction module; and
15[g] correct, with the error correction module physically separate from the error detection module, the forwarded portions of the respective received streams containing an error.	17[d] correcting, with the error correction module physically separate from an error detection module, the forwarded portions of the respective received streams containing an error.	1[d] the error correction module, communicatively coupled with and physically separate from the error detection module, and configured to correct the received portions of the respective received streams containing an error.

15[a] “an encoder configured to encode data for storage on a flash memory,”

15[b] “the flash memory, communicatively coupled with the encoding module, and configured to store the encoded data; and”

Wu discloses a multi-channel memory apparatus with an error correcting module 802 comprising “a plurality of ECC engines 822-0, 822-1 to 822-M” that “perform[] error correction code encoding and decoding on the data to be stored in the memory devices” Ex. 1004 ¶0033, Fig. 8; Ex. 1003 ¶129.

Wu further states that “operations of the data buffer 821 and the ECC engine/sub-ECC engines 822-0 to 822-M are similar to the data buffer 121 and the ECC engine 122 as shown in FIG. 1” and refers the reader to Figures 1 and 2 and the corresponding description “[f]or detailed description of the ... the data buffer 821

[and] the ECC engine/sub-ECC engines 822-0 to 822M ... as shown in FIG. 8.” Ex. 1004 ¶0033; Ex. 1003 ¶130.

In other words, Wu discloses that ECC engine/sub-ECC engines 822-0 to 822-M collectively perform the function of ECC engine 122 shown in Figure 1. Ex. 1003 ¶131. Wu further states, with respect to Figure 1, that “[t]he error correcting module 102 [comprising ECC engine 122 and data buffer 121] is arranged *to perform error correction code encoding on the data to be stored into the memory devices* [105-0 to 105-N, which may be NAND or NOR flash memory].” Ex. 1004 ¶0022; *see also id.* ¶¶0024-25, 0034 (“[T]he ECC engine (such as the ECC engine 122, 222, 822-0 to 822-M and 900 ... etc.) is arranged to perform suitable ECC ... encoding and decoding”); Ex. 1003 ¶131.

Wu therefore discloses “**an encoder**,” ECC engine/sub-ECC engines 822-0 to 822-M (which are also “decoding module” components) **yellow** “**configured to encode data for storage on a flash memory**,” comprised of flash memory devices 105-0 to 105-N, **green**, and “**the flash memory, communicatively coupled with the encoding module, and configured to store the encoded data.**” Ex. 1003 ¶132.

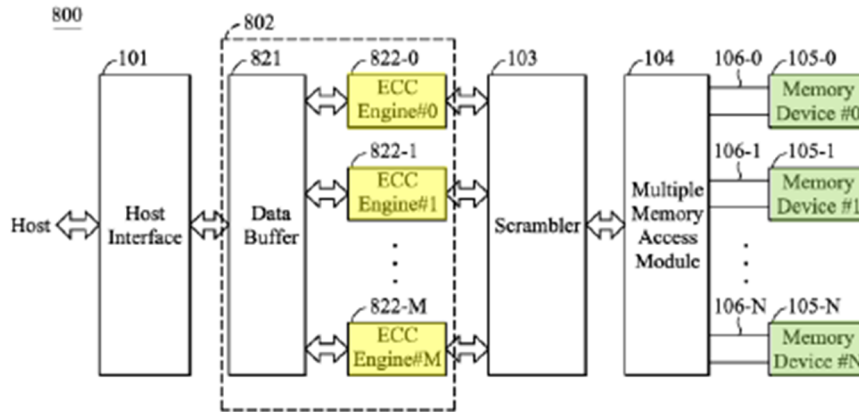


FIG. 8

Ex. 1004, Fig. 8.

5. Claim 19

“The method of claim 17, wherein the error correction module comprises a plurality of error correction sub-modules operating in parallel and fewer in number than the error detection sub-modules.”

Claim 19 depends from claim 17 and contains the same limitations as claim 4, which depends from claim 1. The limitations of claim 19 are therefore disclosed by Wu for the reasons provided with respect to claims 1, 4, and 17. *See* §§VIII.B.1, 2, 4; Ex. 1003 ¶¶135-136.

C. Ground 2: Jo in View of Wu Renders Claims 1, 2, 4, 8, 15, 17, and 19 Obvious

1. Claim 1

1[preamble] “A flash memory decoder comprising”

Regardless of whether the preamble is limiting, it is disclosed by Jo. Ex. 1003 ¶¶137-145.

Figure 6 of Jo (below) shows “a two channel flash memory system 60 that ... includes a flash memory controller 600 [that] communicates data with [NAND flash] memory devices 611 and 612 over two communication channels operating at the same time as each other in parallel.” Ex. 1005 ¶0036; Ex. 1003 ¶139.

“Flash memory controller 600 includes ECC block 645 [that] includes two data encoders 661, one each for the first and second communication channels[;] two syndrome computation blocks 610a and 610b, ... one for each of the first and second communication channels[;] a multiplexer (or data switch) 615, a decoder block 630, a demultiplexer 675, first and second memory buffers 670 and 680, third and fourth memory buffers (not labeled), and an error correction block 663.” Ex. 1005 ¶0037; Ex. 1003 ¶140.

Jo further discloses that “[w]hen data from a host device (e.g., a processor) is to be written to the memory devices ... the data is first encoded ... by a corresponding encoder 661.” Ex. 1005 ¶0038; *see also id.*, ¶0023;

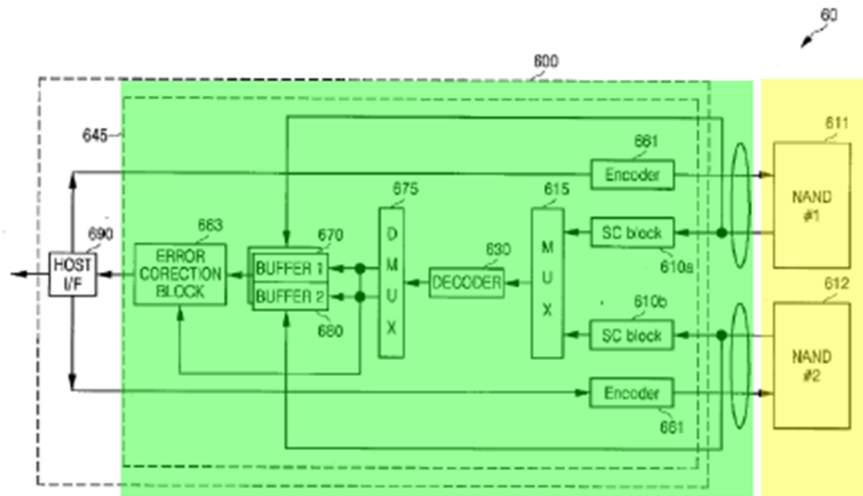
Ex. 1003 ¶141. “When the encoded data is to be read ... it is decoded before being sent to a host device....” Ex. 1005 ¶0039; Ex. 1003 ¶141.

Jo discloses that, as part of the decoding process, “syndrome computation blocks 610a and 610b detect whether or not a data set includes errors.” Ex. 1005 ¶0040; Ex. 1003 ¶143.

“Beneficially, ECC block 645 includes multiplexer 615 so that the two communication channels can share one key equation solver (error decoder block 630.... Once the errors are identified in a data set, then the errors are corrected in error correction block 663 (e.g., an exclusive-or (XOR) circuit) and the decoded data set may then be sent to a host device (e.g., a processor).” Ex. 1005 ¶¶0041; Ex. 1003 ¶144.

The two-channel memory system depicted in Jo’s Figure 6 (and in Jo’s Figures 3 and 4 in alternate embodiments) discloses a “**flash memory**” (yellow) “**decoder**” (green) identified in Figure 6 below. Ex. 1003 ¶145.

FIG. 6



Ex. 1005 Fig. 6.

1[a] “a decoding module configured to: receive encoded data from the flash memory; and decode the received encoded data to generate a plurality of partially decoded data streams;”

Jo discloses that, during a write operation to NAND flash memory devices 611 and 612, data from a host device (*e.g.*, a processor) is “first encoded (*e.g.* with a BCH [or Bose, Ray-Chaudhuri, Hocquenghem] code) by a corresponding encoder 661 for each communication channel. Then, the encoded data is transmitted in parallel across the two communication channels to memory devices 611 and 612.” Ex. 1005 ¶0038; Ex. 1003 ¶147.

Jo discloses that, “[w]hen the encoded data is to be read from memory devices 611 and 612 via the two communication channels, it is decoded before being sent to a host device (*e.g.*, a processor).” Ex. 1005 ¶0039; Ex. 1003 ¶148.

Jo does not expressly disclose that the data received over the two communications channels at the syndrome computation blocks 610a and 610b that are part of an “error detection module” (*see* limitation 1[b] below) has been partially decoded by a “decoding module”; however, Jo in combination with Wu, prior art that is closely analogous Jo as reflected in §§VIII.A.1-2 above, teaches this limitation. Ex. 1003 ¶149.

Wu teaches the use of a scrambler module having, *e.g.*, a plurality of scramblers, such as the scramblers 703-1 to 703-N “each coupled to a storage channel and arranged to perform intra-channel disarrangement for the data to be stored into the corresponding memory device and rearrange the data read out from the corresponding memory device.” Ex. 1004 ¶0032; Ex. 1003 ¶150.

As shown in Wu Figure 7 (below), Wu's scramblers 703-0 to 703-N are associated with corresponding memory devices 105-0 to 105-N (which "may be a NAND or NOR type flash" and collectively constitute a "decoding module." Ex. 1004 ¶¶0024, 0032; Ex. 1003 ¶151.

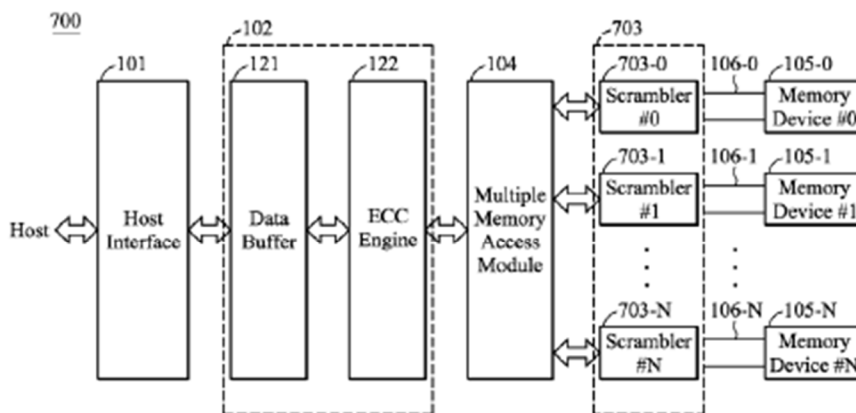


FIG. 7

Ex. 1004, Fig. 7.

Wu discloses that scramblers 703-0 to 703-N are similar to the scrambler 103 as shown in Wu's Figure 1, but operate only on the data read and written to a single corresponding memory device 105-0 to 105-N. Ex. 1004 ¶0032; Ex. 1003 ¶152. Accordingly, each scrambler 703-0 to 703-N may "perform intra-channel disarrangement for the data to be stored into the memory devices and *rearrange the scrambled data read out from the memory devices* in a reverse manner depending on the data transferring directions." Ex. 1004 ¶0023; Ex. 1003 ¶153. The intra-

channel disarrangement/arrangement operation is performed to reduce interference.

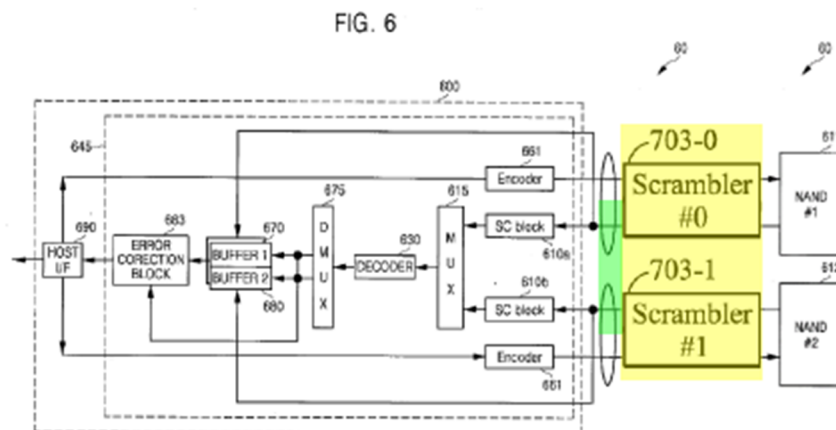
Ex. 1004 ¶0023; Ex. 1003 ¶153.

A POSA would understand that Jo's flash memory system for interfacing a host with NAND flash memory devices 611 and 612 would likewise benefit from the interference reduction provided by the use of Wu's scramblers 703 at each of Jo's memory channels. Ex. 1003 ¶154. In fact, it was known at the time that data scrambling had become of increased importance in addressing NAND flash memory reliability issues. Ex. 1006 ¶0049; Ex. 1003 ¶154.

A POSA would likewise have an expectation that such a combination would be successful; would perform as intended; and could be implemented with minimal, if any experimentation. Ex. 1003 ¶155. A POSA would further understand, *e.g.*, that Wu's scramblers 703-0 to 703-N could be implemented by adding circuitry to Jo's memory controller 600, to Jo's memory devices 611 and 612, or by providing standalone scrambling circuitry. Ex. 1003 ¶155. Further, locating Wu's scramblers as shown in Jo's annotated Figure 6 below helps avoid header and ECC type data pattern failures (Ex. 1006 ¶0070; Ex. 1003 ¶156), and, if located on Jo's memory devices, may promote portability of the memory die so that they can be used with different controllers (Ex. 1006 ¶0070; Ex. 1003 ¶156).

Accordingly, Jo's Figure 6 embodiment, as modified by Wu to further include scramblers for disarranging, *i.e.*, partially decoding, data stored in each memory

device 611 and 612 upon a read operation, discloses and renders obvious the claimed “a decoding module configured to: receive encoded data from the flash memory; and decode the received encoded data to generate a plurality of partially decoded data streams.” Ex. 1003 ¶157; *see also* Ex. 1006, ¶¶0052 (further explaining that “descrambling is considered to be a form of decoding”), 0062. The combination is shown in the Figure below, with the “**decoding module**” (yellow) and the “**plurality of partially decoded data streams**” (green). Ex. 1003 ¶¶156-157.



Ex. 1005, Fig. 6 (annotated to include scramblers disclosed in Wu’s Figure 7 [Ex. 1004]).

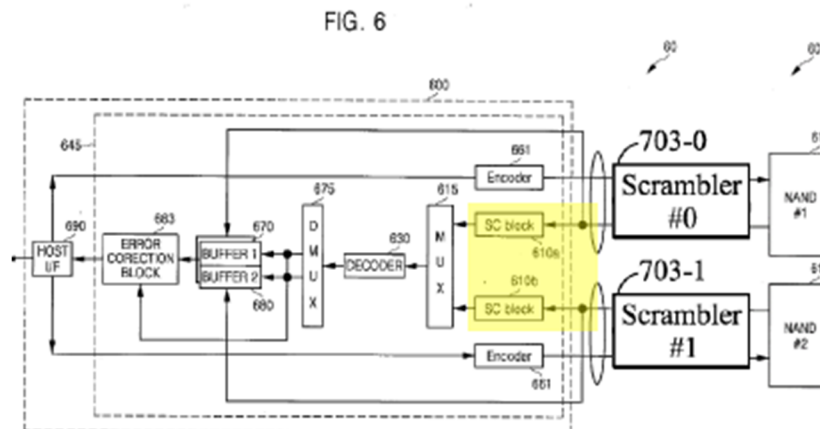
1[b] “an error detection module communicatively coupled with the decoding module, and comprising a plurality of error detection sub-modules operating in parallel,”

Jo discloses that when the encoded data stored in memory devices 611 and 612 is read “via the two communication channels ... syndrome computation block

610a computes the syndrome of a first set of data (e.g., data from an Nth sector of memory) received via the first communication channel at the same time that syndrome computation block 610b computes the syndrome of a second set of data (e.g., data from an Mth sector of memory) via the second communication channel. If the syndrome values for a set of data are zero, then this indicates that the data set does not include errors. Otherwise, then the data set includes one or more errors and therefore, the errors should be located and corrected. That is, the syndrome computation blocks 610a and 610b detect whether or not a data set includes errors.” Ex. 1005 ¶¶0039-40; *see also id.*, cl. 7 (“[T]he first error detector calculates a first syndrome value of a first data [and] the second error detector calculates a second syndrome value of a second data set.”); Ex. 1003 ¶¶142, 159.

Therefore, Jo disclose “**an error detection module,**” (yellow below), “**communicatively coupled with the decoding module,**” as shown in limitation 1[a], “**and comprising a plurality of error detection sub-modules,**” including syndrome computation blocks 610a and 610b, “**operating in parallel**” (*see, e.g.*, Ex. 1005 ¶39 (“[S]yndrome computation block 610a computes the syndrome of a first set of data ... at the same time that syndrome computation block 610b computes the syndrome of a second set of data.”), cl. 16 (“wherein the first syndrome computation block calculates the first syndrome value for the data received via the first channel

while the second syndrome computation block calculates the second syndrome value for the data received via the second channel”).⁷ Ex. 1003 ¶¶160-161.



Ex. 1005, Fig. 6 (annotated to include scramblers disclosed in Wu’s Figure 7 [Ex. 1004]).

1[c] “each error detection sub-module configured to: receive a different one of the plurality of partially decoded data streams; detect whether a portion of the respective received stream contains an error; and forward the portion of the respective received stream containing an error to an error correction module; and”

As depicted in annotated Figure 6 in §VIII.C.1 (limitation 1[b]) above, Jo, as modified by Wu, discloses “**each error detection sub-module,**” blocks 610a and 610b, “**configured to: receive a different one of the plurality of partially decoded**

⁷ As noted above, claim 1 of the ‘700 patent expressly covers flash memory decoders that perform BCH decoding, such as disclosed by Jo. *See* n.5 *supra*; Ex. 1005 ¶¶0038.

data streams” from the scramblers of Wu (the decoding module). Ex. 1003 ¶¶163-164.

Jo further discloses that its “error detection sub-modules” 610a and 610b compute the syndromes of a first set of data from memory 611 and a second set of data from memory 612, respectively. Ex. 1005 ¶¶0039-40; Ex. 1003 ¶165. Jo explains that, through this computation, “the syndrome computation blocks 610a and 610b detect whether or not a data set includes errors.” Ex. 1005 ¶¶0039-40; Ex. 1003 ¶166.

Jo therefore disclose **“each error detection sub-module configured to: ... detect whether a portion of the respective received stream contains an error.”** Ex. 1003 ¶167.

Jo further discloses that when “the data set includes one or more errors ... the errors should be located and corrected.” Ex. 1005 ¶0040; Ex. 1003 ¶168.

Jo describes the error location and correction process as follows with reference to Figure 6: “When data sets are read into memory controller 600 from the two memory chips 611 and 612, errors in the data sets received from the two communication channels are detected at the same time. At this time, the data sets are stored in buffer memories 670 and 680, waiting for correction and transmission to the host.” Ex. 1005 ¶0042; Ex. 1003 ¶168.

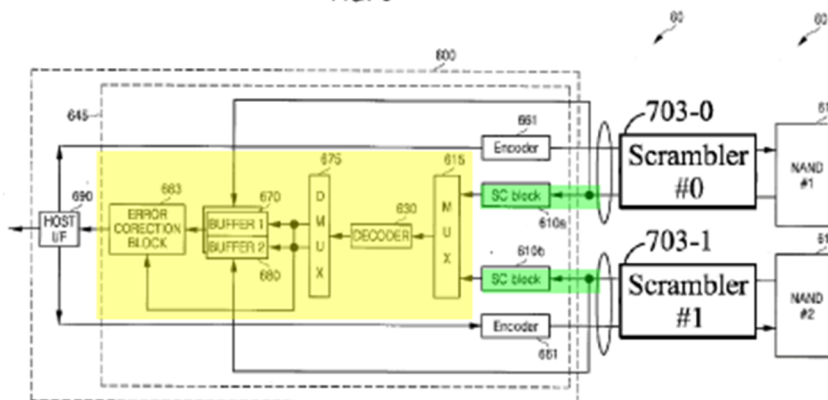
After errors are detected “multiplexer 615 multiplexes between the first and second communication channels for locating [the] errors that occur in data sets received from the memory devices via the communication channels.” Ex. 1005 ¶0041; Ex. 1003 ¶169.

“Decoder block 630 calculates the error locations for a first data set read data [sic] from memory device 611, and then the first set of data stored in buffer memory 670 is corrected....” Ex. 1005 ¶0043; Ex. 1003 ¶170. “[T]he errors are corrected in error correction block 663 (e.g., an exclusive-or (XOR) circuit) and the decoded data set may then be sent to a host device (e.g., a processor).” Ex. 1005 ¶0041; Ex. 1003 ¶170.

“While the first data set stored in buffer memory 670 is being transmitted to the host, decoder block 630 calculates error locations for a second data set read data from memory device 612, and then the second set of data stored in buffer memory 680 is corrected and transmitted to the host....” Ex. 1005 ¶0043; Ex. 1003 ¶170.

Accordingly, Jo discloses “**each error detection sub-module [green below] configured to: ... forward the portion of the respective received stream containing an error to an error correction module.**” Jo’s “error correction module” includes the blocks (yellow), and the forwarded portion of the stream is received at buffers 670 and 680. Ex. 1003 ¶¶171-172.

FIG. 6



Ex. 1005, Fig. 6 (annotated to include scramblers disclosed in Wu’s Fig. 7 [Ex. 1004]).

1[d] “the error correction module, communicatively coupled with and physically separate from the error detection module, and configured to correct the received portions of the respective received streams containing an error.”

Jo’s Figure 6 discloses an **“error correction module,”** identified in §VIII.C.1 (limitation 1[c]) above as comprising circuit components including “multiplexer (or data switch) 615,” “key equation solv[ing] error decoder block 630,” demultiplexer 675, “first and second memory buffers 670 and 680, third and fourth memory buffers (not labeled),” and “error correction block 663 (e.g., an exclusive-or (XOR) circuit)” **“communicatively coupled with,”** at the SC block 610a/b outputs and multiplexer 615 inputs, **“and physically separate from the error detection module”** identified in §VIII.C.1 (limitation 1[b]) above as comprising circuit components including syndrome computation blocks 610a and 610b “physically separate” from the components of the “error correction module.” Ex. 1005 Fig. 6, ¶¶0036-37, 0041; Ex.

1003 ¶¶158-175. Jo notes that this structure promotes efficient use of circuitry within the memory controller. Ex. 1005 ¶0010, Title; Ex. 1003 ¶175.

Jo further discloses that its **“error correction module [is] configured to correct the received portions of the respective received streams containing an error.”** See §VIII.C.1 (limitation 1[c]) (identifying the error correction module and showing that, once detected by the SC blocks, the multiplexer 615, key equation solver decoder block 630, buffer memory, and error correction block 663 of Jo’s “error correction module” operate to “correct the received portions of the respective received streams containing an error”); *see also* Ex. 1005 ¶¶0041-43; Ex. 1003 ¶176-177.

2. Claim 2

“The flash memory decoder of claim 1, wherein each of the error detection sub-modules is configured to read data from a set of one or more sectors of flash memory different from other of the error detection sub-modules.”

With reference to Figure 6 (above), Jo discloses that “syndrome computation block 610a [a first “error detection sub-module”] computes the syndrome of a first set of data (e.g., data from an Nth sector of memory [in NAND memory 611]) received via the first communication channel at the same time that syndrome computation block 610b [a second “error detection sub-module”] computes the syndrome of a second [different] set of data (e.g., data from an Mth sector of memory

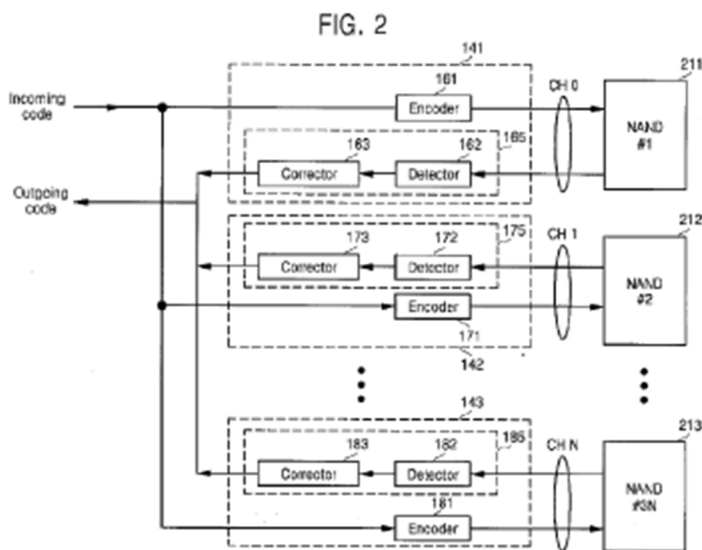
[in NAND memory 612]) via the second communication channel.” Ex. 1005 ¶0036; Ex. 1003 ¶181.

Jo therefore discloses that “each of the error detection sub-modules is configured to read data from a set of one or more sectors of flash memory different from other of the error detection sub-modules.” Ex. 1003 ¶182.

3. Claim 4

“The flash memory decoder of claim 1, wherein the error correction module comprises a plurality of error correction sub-modules operating in parallel and fewer in number than the error detection sub-modules.”

Jo explains that “[t]here is a trend for flash memory systems to have more and more memory devices.” Ex. 1005 ¶0009; Ex. 1003 ¶185. Jo’s Figure 2 (below) depicts an exemplary prior art flash memory system with $N > 2$ memory devices. Ex. 1005 ¶0007; Ex. 1003 ¶185.



Ex. 1005, Fig. 2.

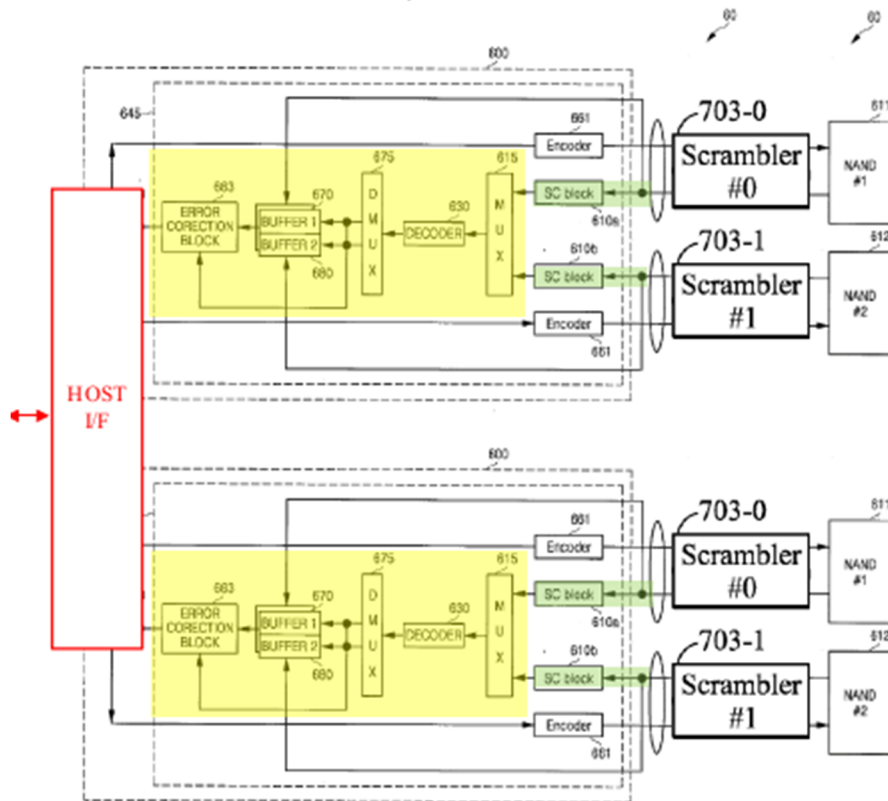
To reduce circuitry area, Jo proceeds to disclose an exemplary “error corrector adapted to correct errors detected by each of the at least first and second error detectors” corresponding to first and second memory devices. Ex. 1005 ¶¶0012, 0041, Figs. 3, 4, 6; Ex. 1003 ¶¶144, 186.

While the exemplary embodiments of Jo Figures 3, 4 and 6 depict two memory devices and two channels, Jo expressly discloses that its inventive concept is not limited to two memory device/channel systems. Ex. 1005 ¶0012 (“In one aspect of the inventive concept, a memory system comprises: *at least* two memory devices; and a memory controller having *at least* first and second communication channels.”); Ex. 1003 ¶187.

A POSA would understand Jo as disclosing, at a minimum, that its 2-channel embodiments can be scaled to 2N channel embodiments, *e.g.*, by combining 2-channel embodiments. Ex. 1003 ¶187. Further, such replication would have been obvious to try, as it would allow Jo’s 2-channel embodiments to be scaled to memory systems having larger capacity, *i.e.*, more flash memory devices. Ex. 1003 ¶¶188-191; Ex. 1005 ¶0006.

A disclosed and obvious exemplary 4-channel system is illustrated below **“wherein the error correction module comprises a plurality of error correction sub-modules,”** (yellow), **“operating in parallel and fewer in number than the error detection sub-modules”** (green). Ex. 1003 ¶189.

FIG. 6



Ex. 1005 Fig. 6 (annotated to include scramblers disclosed in Wu's Figure 7; reproduced twice to depict a 4-channel memory device; and with a modified host interface (in red) expanded to interface with the four NAND memory devices via the four encoders 661 and two error correction blocks 663).

This decoder configuration allows the area saving technique disclosed by Jo to be applied to systems having larger memory capacity, as demanded by market trends expressly noted by Jo. Ex. 1005 ¶¶0009, 0041; Ex. 1003 ¶191. Modifications, including to the host interface, to accommodate more memory devices are implicitly disclosed by Jo's teaching that its representative embodiments are not limited to 2-

channel devices, and in any case would have required no undue experimentation and would have been trivial to implement for a POSA. Ex. 1003 ¶191. This is further confirmed by the fact that Jo discloses a prior art host interface 190 shown in Figure 1 that interfaces with more than two NAND memory devices. Ex. 1005 ¶0006; Ex. 1003 ¶191.

4. Claim 8

8[a] “The flash memory decoder of claim 4, wherein, each error correction sub-module is assigned a set of two or more error detection sub-modules different from error detection sub-modules associated with other of the error correction sub-modules; and”

As shown in Jo’s annotated Figure 6 (see §VIII.C.3 (claim 4) above) Jo discloses **“each error correction sub-module,” yellow**, **“is assigned a set of two or more error detection sub-modules,” shaded green** **“different from error detection sub-modules associated with other of the error correction sub-modules.”** Ex. 1003 ¶194.

8[b] “each error correction sub-module is configured to receive portions of the respective received streams containing an error detected in parallel by the set of two or more error detection sub-modules assigned to the respective error correction sub-module.”

Jo’s annotated Figure 6 (see §VIII.C.4 (claim 4) above), includes two error correction sub-modules. Each of these sub-modules, and their connected error detection sub-modules, independently meet limitations 1[b], [c], and [d] based on the disclosures identified in §VIII.C.1. Ex. 1003 ¶¶194-195.

These same disclosures establish that Jo discloses limitation 8[b]. Ex. 1003 ¶196.

For example, as explained with reference to limitation 1[c] and [d], Jo discloses an “**error correction module**” (and sub-module in the context of claims 4 and 8) that “**is configured to receive portions of the respective received streams containing an error detected in parallel by the set of two or more error detection sub-modules assigned to the respective error correction sub-module.**” Ex. 1003 ¶197.

Jo describes an operation of the error correction module (or sub-module) with reference to Figure 6. Ex. 1003 ¶198. While errors in the data read from memory devices 611 and 612 are being detected in parallel by assigned error detection sub-modules comprising blocks 610a and 610b, the data is received and stored in buffer memories 670 and 680 of the error correction module (or sub-module). *See* §VIII.C.1 (limitations 1[b], [c], and [d]); Ex. 1005 ¶¶0036-37, 0039-43; Ex. 1003 ¶198. Data containing errors is corrected by the error correction module (or sub-module) and transmitted to a host. *See* §VIII.C.1 (limitation 1[d]); Ex. 1005 ¶¶0036-37, 0041-43; Ex. 1003 ¶198.

5. Claims 15 and 17

As discussed above, independent claims 15 and 17 recast apparatus claim 1 in system (claim 15) and method (claim 17) format. Therefore, claims 15 and 17 are

disclosed and rendered obvious by Jo and Wu for the same reason that Jo and Wu disclose and render obvious claim 1 (the combination discloses the claimed apparatus, system and method). *See* §VIII.C.1 (claim 1).

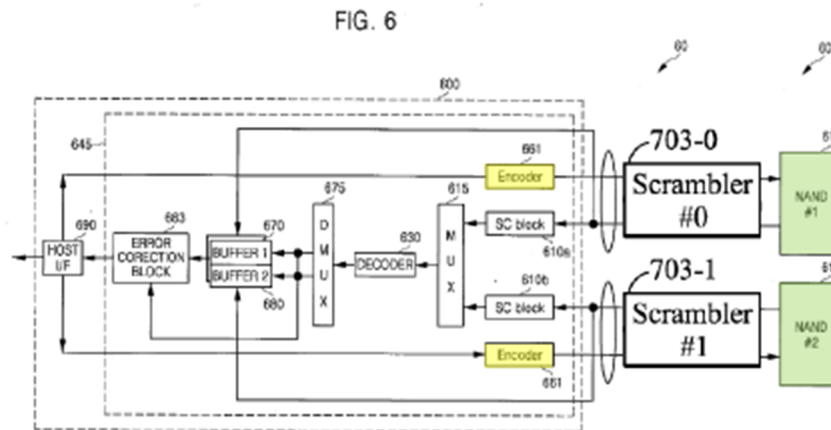
The table in §VIII.B.4 provides a cross-reference for the claims, where only limitations 15[a] and 15[b] are not equivalently in claim 1. Analysis of these two limitations follows.

15[a] “an encoder configured to encode data for storage on a flash memory,”

15[b] “the flash memory, communicatively coupled with the encoding module, and configured to store the encoded data; and”

Figure 6 of Jo (below) shows a “[f]lash memory controller 600” that “includes two data encoders 661, one each for the first and second communication channels.” Ex. 1005 ¶0037; Ex. 1003 ¶201. Jo explains that “[i]n operation, ECC block 645 communicates data with memory devices 611 and 612.... When data from a host device (e.g., a processor) is to be written to the memory devices via the two communication channels, the data is first encoded (e.g. with a BCH code) by a corresponding encoder 661 for each communication channel. Then, the encoded data is transmitted in parallel across the two communication channels to memory devices 611 and 612.” Ex. 1005 ¶0038; *see also id.* ¶36 (disclosing that memory devices 611 and 612 are NAND flash memory); Ex. 1003 ¶201.

Jo therefore discloses “**an encoder,**” encoders 661 (yellow) below and collectively forming the recited “**encoding module,**” “**configured to encode data for storage on a flash memory,**” comprised of memory devices 611 and 612 (green), and “**the flash memory, communicatively coupled with the encoding module, and configured to store the encoded data.**” Ex. 1003 ¶202.



Ex. 1005, Fig. 6 (annotated to include scramblers disclosed in Wu’s Figure 7); *see also* Ex. 1004 ¶¶0023, 0032 (disclosing that scramblers 703 “perform intra-channel disarrangement for the data to be stored into the memory devices”).

6. Claim 19

“The method of claim 17, wherein the error correction module comprises a plurality of error correction sub-modules operating in parallel and fewer in number than the error detection sub-modules.”

Claim 19 depends from claim 17 and contains the same limitations as claim 4, which depends from claim 1. The limitations of claim 19 are therefore disclosed

by Jo and rendered obvious by Jo and Wu for the reasons provided with respect to claims 1, 4, and 17. *See* §§VIII.C.1, 3, 5; Ex. 1003 ¶¶204.

D. Ground 3: Jo in View of Wan Renders Claims 1, 2, 4, 8, 15, 17, and 19 Obvious

Unpatentability under Ground 3 is based on substantially the same Jo disclosures identified in Ground 2. Ground 3, however, relies on a different reference, Wan, for disclosure of the “decoding module” of limitation 1[a]. While Wan and Wu both disclose a “decoding module” comprised of scrambling/descrambling modules, the implementation of these modules differs in each reference. For example, Wan discloses that the scrambler/descrambler may be implemented on a memory die. Wu and Wan also qualify as prior art under different subsections of §102.

1. Claim 1

1[preamble] “A flash memory decoder comprising”

Regardless of whether the preamble is limiting, it is disclosed by Jo. *See* §VIII.C.1; Ex. 1003 ¶¶206-209.

1[a] “a decoding module configured to: receive encoded data from the flash memory; and decode the received encoded data to generate a plurality of partially decoded data streams;”

As noted in §VIII.C.1, above, Jo discloses that, during a write operation to NAND flash memory devices 611 and 612, data from a host device (*e.g.*, a processor) is “first encoded (*e.g.* with a BCH code) by a corresponding encoder 661

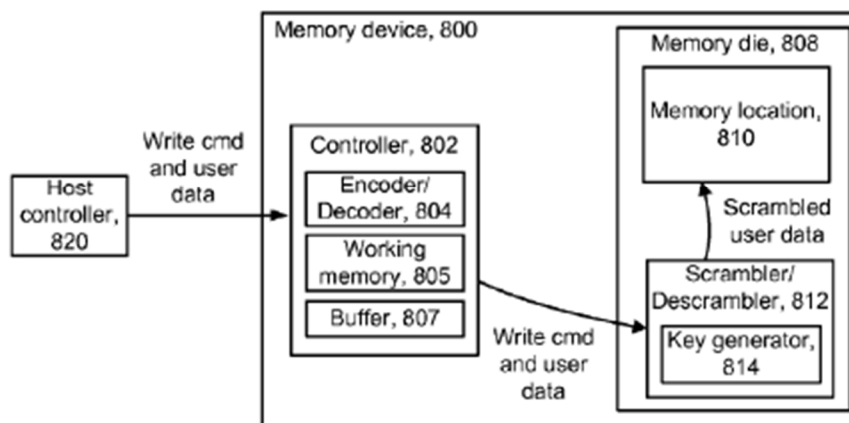
for each communication channel. Then, the encoded data is transmitted in parallel across the two communication channels to memory devices 611 and 612,” which may be memory chips. Ex. 1005 ¶¶0038, 0042; Ex. 1003 ¶211.

As further noted in §VIII.C.1, Jo discloses that, “[w]hen the encoded data is to be read from memory devices 611 and 612 via the two communication channels, it is decoded before being sent to a host device (e.g., a processor). Ex. 1005 ¶0039; Ex. 1003 ¶212.

Jo does not expressly disclose that the data received over the two communications channels at the syndrome computation blocks 610a and 610b that are part of an “error detection module” (see §VIII.C.1 (limitation 1[b])) has been partially decoded by a “decoding module”; however, Jo in combination with Wan teaches this limitation. Ex. 1003 ¶¶212-213.

Wan discloses a flash memory system embodiment depicted in Figure 8a below. Ex. 1003 ¶213.

Fig. 8a



Ex. 1006, Fig. 8a.

Wan discloses that the memory die, 808 in Figure 8a above, includes a memory array of NAND flash storage elements capable of storing one or multiple bits of information. Ex. 1006 ¶¶0042 (“[A] memory system suitable for implementing the present invention uses the NAND flash memory structure.”), 0045 (“Each storage element can store ... one bit of digital data.... A storage element can also store multiple levels of information, for example, multiple bits of digital data.”), 0050 (“A chip refers to a memory die on which a memory array is formed.”), 0054 (“A memory array 509 having a number of blocks of storage elements may be formed on the memory die 508.”), 0089 (“Memory die 1198 includes a two-dimensional array of storage elements 1000.”), cl. 20; Ex. 1003 ¶214.

As disclosed in Wan’s Figure 8a (above) and 8b, and corresponding discussion in the specification, data scrambling and descrambling is performed on the data to be stored to, and read from memory. Ex. 1006 ¶0070; Ex. 1003 ¶215. These operations are performed by scrambler/descrambler circuitry 812 residing on memory die, or chips, 808. Ex. 1006 ¶0070; Ex. 1003 ¶215.

Wan further discloses that “[t]his approach provides portability of the memory die so that it can be used with different controllers, for instance, which are made by different manufacturers.” Ex. 1006 ¶0070; Ex. 1003 ¶¶216-217.

To write user data to memory array 810, Wan states that “the host controller 820 provides a write command and associated user [data] to the controller 802, which in turn communicates one or more corresponding write commands and the user data to the scrambler/descrambler 812 on the memory die 808.” Ex. 1006 ¶0070; Ex. 1003 ¶¶216, 218. A key generator 814 within scrambler/descrambler 812 may be used to generate different random or pseudo random keys used to scramble the user data. Ex. 1006 ¶0070; *see also id.*, ¶0072; Ex. 1003 ¶216, 218. “[T]he key which is used to scramble a portion of user data can be based on an address of a memory location at which the portion of user data is to be stored.... When the user data is subsequently read back, the same key can be generated for descrambling or other decoding.” Ex. 1006 ¶0074; Ex. 1003 ¶219.

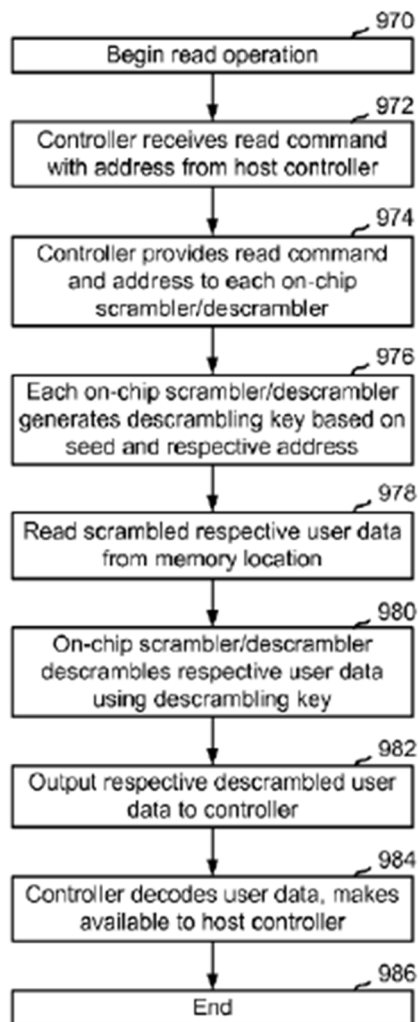
Wan further states that, prior to sending to the scrambler/descrambler 812, “controller 802 may process the user data such as by performing ECC coding and adding overhead data [*e.g.*, coding redundancy/parity bits] at an encoder/decoder 804. With this approach, the overhead data is also scrambled with the user data. This helps avoid header and ECC type data pattern failures.” Ex. 1006 ¶0070; Ex. 1003 ¶220.

With respect to reading user data from the memory array Wan states that when the controller 802 receives a read command from the host controller 820 “[t]he controller 802 provides one or more corresponding read commands to the

scrambler/descrambler 812. The scrambler/descrambler 812 may use the key generator 814 to generate the different random or pseudo random keys which were used for scrambling.... The descrambled user data is then provided to the controller 802 and stored in the buffer 807 for read out by the host controller 820.” Ex. 1006 ¶10071; Ex. 1003 ¶221.

Wan depicts the read procedure in detail in Figure 10b (below).

Fig. 10b



Ex. 1006, Fig. 10b; *see also id.*, ¶0081 (describing Figure 10b); Ex. 1003 ¶222.

Wan also discloses a removable flash storage card embodiment in Figure 12 (below) in which “an off-chip controller 1150 (analogous to ... controller 802 in FIG. 8a and 8b) is included in the same memory device (e.g., a removable storage card) as ... one or more memory die.” Ex. 1006 ¶0089; Ex. 1003 ¶223. In this embodiment, “[c]ommands and Data are transferred ... between the controller and the one or more memory die 1198 via lines 1118.” Ex. 1006 ¶0089; Ex. 1003 ¶223. Each memory die 1198 contains on-chip scrambler/descrambler circuitry within the read/write circuits 1165 “which provides the functionality described in connection, e.g., with FIGS. 8a to 10b.” Ex. 1006 ¶0101; *see also id.* ¶0093, Figs. 12 and 13; Ex. 1003 ¶223.

A POSA would understand that Jo’s flash memory system for interfacing a host with NAND flash memory devices 611 and 612 (as shown in Jo’s Figure 6 above) would benefit from the reliability improvement provided by incorporating Wan’s on die scrambler/descrambler circuitry into Jo’s NAND flash memory die 611 and 612. Ex. 1003 ¶224. The benefits and motivation for doing so would, *e.g.*, include those described in §VIII.C.1 (limitation 1[a]) above. Ex. 1003 ¶224.

Also, Wan notes that “[d]ata scrambling for non-volatile memory such as NAND memories has become increasingly important in addressing system reliability issues that are inherently susceptible by design to worst-case data patterns

[including fixed repetitive data,] which trigger memory program failures.” Ex. 1006 ¶0049; Ex. 1003 ¶225. Wan further explains that “[d]ata scrambling randomizes the distribution of data states in the storage elements so that worst-case data patterns are avoided.” Ex. 1006 ¶0049; Ex. 1003 ¶225. That is, Wan itself teaches that modifying Jo’s memory devices 611 and 612 to include the scrambler/descrambler circuitry disclosed by Wan and summarized above would improve memory device reliability. Ex. 1003 ¶225.

Wan further discloses that adding scrambler/descrambler to the memory die “provides portability of the memory die so that [they] can be used with different controllers,” for instance flash memory controller 600 disclosed by Jo. Ex. 1006 ¶0070; Ex. 1003 ¶226.

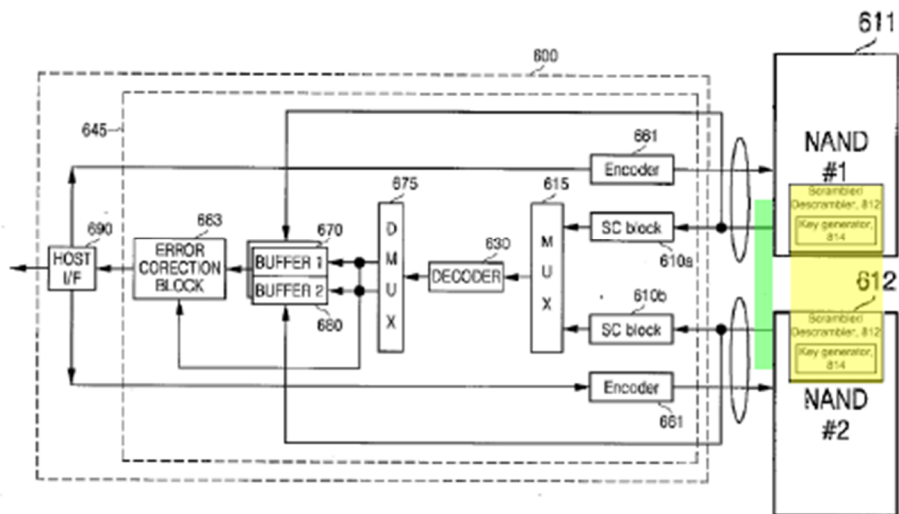
A POSA would likewise have an expectation that this combination of Jo and Wan, closely analogous art as reflected by the disclosures identified above and in §§VIII.A.2-3, would be successful; would perform as intended; and could be implemented with minimal, if any experimentation. Ex. 1003 ¶227.

Wan also explains that “descrambling is considered to be a form of decoding,” and that Wan uses the terms scrambling and descrambling to describe examples “in which random or pseudorandom data is used to encode or decode user data.” Ex. 1006 ¶0052; *see also id.*, ¶0069 (“the controller decodes, e.g., descrambles, the user data”); Ex. 1003 ¶228. Thus, Wan’s scrambler/descrambler circuitry partially

encodes and decodes data which is further encoded and decoded by Jo's ECC block 645. Ex. 1003 ¶228.

Jo's Figure 6 embodiment, as modified by Wan to incorporate on-die scrambler/descrambler circuitry into Jo's NAND flash memory chips 611 and 612 that scrambles ECC encoded data to be stored upon write operations and descrambles, *i.e.*, partially decodes, data retrieved from memory chips 611 and 612 upon read operations prior to ECC decoding, discloses and renders obvious the claimed **“a decoding module configured to: receive encoded data from the flash memory; and decode the received encoded data to generate a plurality of partially decoded data streams.”** Ex. 1003 ¶229. The combination is shown in the Figure below, with the **“decoding module”** (yellow) and the **“plurality of partially decoded data streams”** (green). Ex. 1003 ¶229.

FIG. 6



Ex. 1005, Fig. 6 (annotated to show scrambler/descramblers of Wan's Figures 8a and 8b [Ex. 1006] within Jo's memory chips 611 and 612).

1[b] “an error detection module communicatively coupled with the decoding module, and comprising a plurality of error detection sub-modules operating in parallel,”

Limitation 1[b] is disclosed by Jo for the reasons provided in §VIII.C.1. Ex. 1003 ¶¶230, 158-162.

1[c] “each error detection sub-module configured to: receive a different one of the plurality of partially decoded data streams; detect whether a portion of the respective received stream contains an error; and forward the portion of the respective received stream containing an error to an error correction module; and”

Limitation 1[c] is disclosed by Jo for the reasons provided in § VIII.C.1. Ex. 1003 ¶¶231, 163-172.

1[d] “the error correction module, communicatively coupled with and physically separate from the error detection module, and configured to correct the received portions of the respective received streams containing an error.”

Limitation 1[d] is disclosed by Jo for the reasons provided in § VIII.C.1. Ex. 1003 ¶¶232-233, 173-179.

2. Claim 2

Claim 2 is rendered obvious by Jo and Wan for the reasons provided in §VIII.C.2. Ex. 1003 ¶¶234-236, 180-183.

3. Claim 4

Claim 4 is rendered obvious by Jo and Wan for the reasons provided in §VIII.C.3 above. Ex. 1003 ¶¶237-239, 184-192.

4. Claim 8

Claim 8 is rendered obvious by Jo and Wan for the reasons provided in §VIII.D.1 [Ground 3: claim 1] and VIII.C.4 [Ground 2: claim 8]. Ex. 1003 ¶¶240-242.

5. Claims 15 and 17

Claims 15 and 17 largely recast apparatus claim 1 in system (claim 15) and method (claim 17) format. Therefore, claims 15 and 17 are disclosed and rendered obvious by Jo and Wan for the same reason that Jo and Wan disclose and render obvious claim 1 (the combination discloses the claimed apparatus, system and method). *See* §VIII.D.1 [claim 1].

The table in §VIII.B.4 provides a cross-reference for the claims, where only limitations 15[a] and 15[b] are not equivalently in claim 1. Analysis of these to limitations follows.

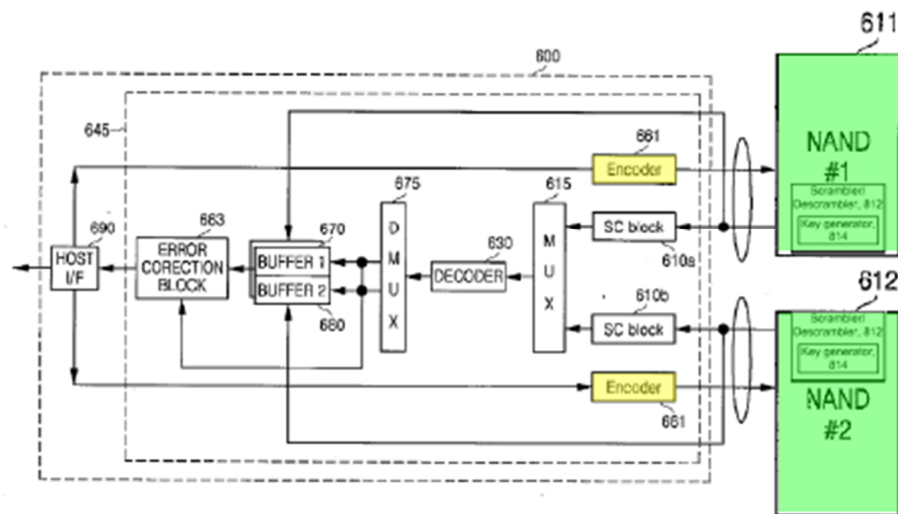
15[a] “an encoder configured to encode data for storage on a flash memory,”

15[b] “the flash memory, communicatively coupled with the encoding module, and configured to store the encoded data; and”

As explained in §VIII.C.5, Jo discloses “**an encoder,**” encoders 661 (yellow) below and collectively forming the recited “encoding module,” “**configured to encode data for storage on a flash memory,**” comprised of memory devices 611

and 612 (green), and “the flash memory, communicatively coupled with the encoding module, and configured to store the encoded data.” Ex. 1003 ¶¶244-246.

FIG. 6



Ex. 1005, Fig. 6 (annotated to show scrambler/descramblers of Wan’s Figures 8a and 8b within Jo’s memory chips 611 and 612); *see also* Ex. 1006 ¶0070 (disclosing a data storage operation in which Wan’s “host controller 820 provides a write command and associated user [data] to the [Wan’s] controller 802, which in turn communicates one or more corresponding write commands and the user data to the scrambler/descrambler 812 on the memory die 808. The controller 802 may process the user data such as by performing ECC coding and adding overhead data at an encoder/decoder 804 [prior to sending the data to the scrambler/descrambler 812].”), Fig. 8a; Ex. 1003 ¶245.

6. Claim 19

“The method of claim 17, wherein the error correction module comprises a plurality of error correction sub-modules operating in parallel and fewer in number than the error detection sub-modules.”

Claim 19 depends from claim 17 and contains the same limitations as claim 4, which depends from claim 1. The limitations of claim 19 are therefore disclosed by Jo and rendered obvious by Jo and Wan for the reasons provided with respect to claims 1, 4, and 17. *See* §§VIII.D.1, 3, 5; Ex. 1003 ¶¶247-248.

VIII. Conclusion

The Challenged Claims are unpatentable. Petitioner requests cancellation.

Respectfully submitted,

BAKER BOTTS L.L.P.

December 27, 2022

Date

/Robinson Vu/

Robinson Vu (Reg. No. 60,211)

LEAD COUNSEL FOR KIOXIA
CORPORATION AND KIOXIA
AMERICA, INC.

CERTIFICATE OF COMPLIANCE

I hereby certify that this paper complies with the type-volume limitation of 37 C.F.R. §42.24 (as determined by the Microsoft Word word-processing system used to prepare this paper) because it contains 13,340 words, excluding the parts of the paper exempted by 37 C.F.R. §42.24(a).

December 27, 2022

Date

/Robinson Vu/

Robinson Vu (Reg. No. 60,211)

LEAD COUNSEL FOR KIOXIA
CORPORATION AND KIOXIA
AMERICA, INC.

CERTIFICATE OF SERVICE

I hereby certify that the foregoing Petition for *Inter Partes* Review and the Exhibits listed on the List of Exhibits were served on December 27, 2022, via Federal Express overnight upon the following:

Viasat, Inc.
c/o Patent Department
6155 El Camino Real
Carlsbad, CA 92009

With Courtesy Copy via Email to Patent Owner's Litigation Counsel of Record:

J. Scott McBride (scott.mcbride@bartlitbeck.com)
Matthew R. Ford (matthew.ford@bartlitbeck.com)
Nevin M. Gewertz (nevin.gewertz@bartlitbeck.com)
Nosson D. Knobloch (nosson.knobloch@bartlitbeck.com)
Meg E. Fasulo (meg.fasulo@bartlitbeck.com)
Melissa R. Smith (melissa@gillamsmithlaw.com)

December 27, 2022

Date

/Robinson Vu/

Robinson Vu (Reg. No. 60,211)

LEAD COUNSEL FOR KIOXIA
CORPORATION AND KIOXIA
AMERICA, INC.