PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 1 OF 4	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	4,112,502	09-05-1978	James Herman Scheuneman	
	2	5,404,485	04-04-1995	Ban, Amir	
	3	5,942,004	08-24-1999	Paolo Cappelletti	
	4	5,956,743	09-21-1999	Bruce, et al.	
	5	6,023,781	02-08-2000	Katsuki Hazama	
	6	6,279,133	08-21-2001	Vafai, et al.	
	7	6,591,394	07-08-2003	Lee, et al.	
	8	6,628,723	09-30-2003	Gerlach, et al.	
	9	6,731,538	05-04-2004	Noda, et al.	
	10	6,732,322	05-04-2004	Miyauchi, et al.	
	11	6,839,870	01-04-2005	Fanfelle	
	12	6,871,303	03-22-2005	Steven J. Halter	
	13	6,895,543	05-17-2005	Katsuki Hazama	
	14	7,061,804	06-13-2006	Chun, et al.	
	15	7,184,356	02-27-2007	Noguchi, et al.	
	16	7,389,465	06-17-2008	Radke, et al.	
	17	7,451,383	11-11-2008	Kim et al.	
	18	7,478,314	01-13-2009	Cheong, et al.	
	19	7,643,342	01-05-2010	Litsyn, et al.	
	20	7,739,576	06-15-2010	William H. Radke	
	21	7,809,900	10-05-2010	Radoslav Danilak	
	22	7,844,879	11-30-2010	Ramamoorthy, et al.	
	23	7,860,200	12-28-2010	Furman et al.	
	24	7,904,619	03-08-2011	Radoslav Danilak	
	25	7,904,672	03-08-2011	Radoslav Danilak	
	26	7,904,780	03-08-2011	Brandman, Yigal	
	27	7,962,831	06-14-2011	Park, et al.	
	28	8,046,542	10-25-2011	William H. Radke	
	29	8,051,358	11-01-2011	William H. Radke	

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STÁTEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 2 OF 4	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	30	8,065,583	11-22-2011	William H. Radke	
	31	8,090,980	01-03-2012	Radoslav Danilak	
	32	8,145,855	03-27-2012	Wan, et al.	
	33	2002/0120820	08-29-2002	Higuchi, et al.	
	34	2002/0159285	10-31-2002	Morley, et al.	
	35	2003/0088821	05-08-2003	Yokokawa et al.	
	36	2003/0112879	06-19-2003	Antia et al.	
	37	2003/0156454	08-21-2003	Wei, et al.	
	38	2004/0268065	12-30-2004	Hilton, et al.	
	39	2005/0114587	05-26-2005	Chou, et al.	
	40	2006/0221752	10-05-2006	Fasoli, et al.	
	41	2007/0061689	03-15-2007	Park, et al.	
	42	2007/0124647	05-31-2007	Chen, et al.	
	43	2007/0171714	07-26-2007	Wu, et al.	
	44	2007/0171730	07-26-2007	Ramamoorthy, et al.	
	45	2007/0266295	11-15-2007	Kevin M. Conley	
	46	2007/0266296	11-15-2007	Kevin M. Conley	
	47	2008/0010582	01-10-2008	Nieto, et al.	
	48	2008/0034272	02-07-2008	Wu, et al.	
	49	2008/0137414	06-12-2008	Park, et al.	
	50	2008/0163023	07-03-2008	Hong, et al.	
	51	2008/0168319	07-10-2008	Lee, et al.	
	52	2008/0294960	11-27-2008	Sharon et al.	
	53	2008/0320361	12-25-2008	Fukuda, et al.	
	54	2008/0320373	12-25-2008	Kim et al.	
	55	2009/0013233	01-08-2009	William H. Radke	
	56	2009/0013234	01-08-2009	William H. Radke	
	57	2009/0037627	02-05-2009	Rofougaran, Ahmadreze (Reza)	
	58	2009/0100307	04-16-2009	Lee, Schweiray Joseph	

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 3 OF 4	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	59	2009/0132889	05-21-2009	William H. Radke	
	60	2009/0327589	12-31-2009	Mark Moshayedi	
	61	2009/0327840	12-31-2009	Mark Moshayedi	
	62	2010/0023800	01-28-2010	Harari, et al.	
	63	2010/0115376	05-06-2010	Shalvi, et al.	
	64	2010/0199153	08-05-2010	Okamura et al.	
	65	2011/0019475	01-27-2011	Mark Moshayedi	
	66	2011/0029716	02-03-2011	Mark Moshayedi	
	67	2011/0060969	03-10-2011	Ramamoorthy, et al.	
	68	2011/0167199	07-07-2011	Radoslav Danilak	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]
	69	JP 2000149592	05-30-2000	NEC		English Abstract
	70	KR 100766042	10-04-2007	Samsung		English Abstract only
	71	WO 2006070668	07-06-2006	IBM		English Abstract
	72	WO 2004062113	07-22-2004	Interdigital Tech Corp.		
	73	WO 2007084751	07-26-2007	Marvell World Trade Ltd.		

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	74	DANESHGARAN, et al.; "An extensive search for good punctured rate-k/(k+1) recursive convolutional codes for serially concatenated convolutional codes," Information Theory, IEEE Transactions on, vol.50, no.1, pp. 208-217, Jan. 2004.	
	75	FRENGER, et al.; "Rate-compatible convolutional codes for multirate DS-CDMA systems," Communications, IEEE Transactions on, vol.47, no 6, pp.828-836, Jun 1999.	

Examiner Signature	Date Considered
***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T[1] - Place a check mark in this area when an English language Translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 4 OF 4	Attorney Docket No.	DATUMTE.076A1

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	76	INTEL CORPORATION; Understanding the Flash Translation Layer (FTL) Specification, Application Note AP-684; retrieved from the Internet <URL: http://staff.ustc.edu.cn/~jpq/paper/flash/2006-Intel%20TR-Understanding%20the%20flash%20translation%20layer%20%28FTL%29%20specification.pdf>; December 1998; retrieved January 26, 2012; Intel Corporation.	
	77	JIM HANDY; Does MLC Flash Belong in Enterprise SSDs; retrieved from the Internet <URL: http://www.infostor.com/index/articles/display/3214572139/articles/infostor/volume-14/issue-1/special-report/does-mlc_flash_belong.html>; February 1, 2010; retrieved on January 24, 2012.	
	78	OHTSUKI, T.; "Rate adaptive indoor infrared wireless communication systems using repeated and punctured convolutional codes," Communications, 1999. ICC '99. 1999 IEEE International Conference on, vol.1, no., pp.609-613 vol.1, 1999.	
	79	SANDFORCE; Product Brief SF-1500 Enterprise SSD Processors; retrieved from the Internet <URL: http://www.sandforce.com/userfiles/file/downloads/LSI_SandForce_1500ENT_PB_120104.pdf>; 2012; retrieved January 24, 2012; LSI Corporation; Milpitas, CA.	
	80	SANDFORCE; Product Brief SF-2500 & SF-2600 Enterprise SSD Processors; retrieved from the Internet <URL: http://www.sandforce.com/userfiles/file/downloads/LSI_SandForce_2500-2600ENT_PB_120104.pdf>; 2012; retrieved on January 24, 2012; LSI Corporation; Milpitas, CA.	
	81	SANDFORCE; RAISE Improves Total SSD Reliability; retrieved from the Internet <URL: http://sandforce.com/index.php?id=174&parentId=3>; 2012; retrieved on January 24, 2012.	
	82	SANDFORCE; SandForce SSD Processors Transform Mainstream Data Storage; retrieved from the Internet <URL: http://www.sandforce.com/userfiles/file/downloads/SFI_Launch_PR_Final.pdf> April 13, 2009; retrieved on Jan. 24, 2012; Saratoga, CA.	
	83	ZHOU, et al.; "High rate turbo code using unevenly punctured convolutional constituent code," Communications, 1999. APCC/OECC '99. Fifth Asia-Pacific Conference on... and Fourth Optoelectronics and Communications Conference, vol. 1, no., pp.751-754 vol.1, 1999.	
	84	STEC; Engineering MLC Flash-Based SSDs to Reduce Total Cost of Ownership in Enterprise SSD Deployments; retrieved from the Internet <URL: http://www.stec-inc.com/downloads/whitepapers/SAFE_WP.pdf>; 2011; STEC, Inc.; retrieved on January 24, 2012.	
	85	STEC; STEC Secure Array of Flash Elements Technology Improves Reliability of Enterprise-Class SSDs; retrieved from the Internet <URL: http://www.stec-inc.com/downloads/whitepapers/SAFE_WP.pdf>; 2011; STEC, Inc.; retrieved on January 24, 2012.	

13056094_1
052212

Examiner Signature	Date Considered
***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T[1] - Place a check mark in this area when an English language Translation is attached.

Electronic Acknowledgement Receipt

EFS ID:	12832576
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/Sandra Autry
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	22-MAY-2012
Filing Date:	
Time Stamp:	15:31:12
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	ADS_DATUMTE076A1.pdf	244930 126381ac7090a720231 3245df516f0c1dd41 185d	no	4
Warnings:					
Information:					

This is not an USPTO supplied ADS fillable form					
2		Spec_DATUMTE076A1.pdf	778898 8b85dda2ac84015b92d1f0d7d1f5e22aae959454	yes	98

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Specification	1	90
	Claims	91	95
	Abstract	96	96
	Appendix to the Specification	97	98

Warnings:					
Information:					
3	Drawings-only black and white line drawings	Drawings_DATUMTE076A1.pdf	588686 f9d55ece118a042bb95118823fe30485a2fe82f7	no	16
Warnings:					
Information:					
4	Foreign Reference	FP1_DATUMTE076A1.pdf	830825 f12e7758a52c65e92bc32c009ff596d4baac10de	no	13
Warnings:					
Information:					
5	Foreign Reference	FP2_DATUMTE076A1.pdf	30157 f42707629389cd0ebc86a8793e8f6aa629aab14d	no	1
Warnings:					
Information:					
6	Foreign Reference	FP3_DATUMTE076A1.pdf	1229741 4405f0f52087531b59325d97907aeda3f3eddd82	no	27
Warnings:					
Information:					
7	Foreign Reference	FP4_DATUMTE076A1.pdf	2806477 d6bffe2623a7f32e7dd5c8b25020ec378ee26973	no	67
Warnings:					
Information:					
8	Foreign Reference	FP5_DATUMTE076A1.pdf	2101133 1d445ea9132dddcd46cf6fe449cb2036d4fba9c6	no	49

Warnings:					
Information:					
9	Non Patent Literature	NPL01_DATUMTE076A1.PDF	795682 5471e9616a9a06468979dcf2fdc0e7a50a8ba815	no	10
Warnings:					
Information:					
10	Non Patent Literature	NPL02_DATUMTE076A1.PDF	976537 b8da8795c88dcebccbf70ced29ce2e99a8fba945	no	9
Warnings:					
Information:					
11	Non Patent Literature	NPL03_DATUMTE076A1.PDF	89709 578a3b5bea49033934c5d7935273b19b62d1a959	no	20
Warnings:					
Information:					
12	Non Patent Literature	NPL04_DATUMTE076A1.PDF	566694 357aed218b5b7cf713c002613b6db051b88738d8	no	3
Warnings:					
Information:					
13	Non Patent Literature	NPL05_DATUMTE076A1.PDF	378767 195738fa6c3644f3926ae46599ce238f87a2ad36	no	5
Warnings:					
Information:					
14	Non Patent Literature	NPL06_DATUMTE076A1.PDF	659802 68e2c678754dcd561daa4b0020806cf395e1e97a	no	2
Warnings:					
Information:					
15	Non Patent Literature	NPL07_DATUMTE076A1.PDF	780390 7b6bfd9cd1d549ab4fce3cc0113316af67126395	no	2
Warnings:					
Information:					
16	Non Patent Literature	NPL08_DATUMTE076A1.PDF	515732 d295c47dd80c7924759d6a5ce38cce91b564f816	no	2
Warnings:					
Information:					
17	Non Patent Literature	NPL09_DATUMTE076A1.PDF	181054 d4ee92334145c0663f0214ca3c81e69c9ff14019	no	6

Warnings:					
Information:					
18	Non Patent Literature	NPL10_DATUMTE076A1.PDF	237800 93ab2fffdaf6a0301d34303fdcf48cc62060c9d8	no	4
Warnings:					
Information:					
19	Non Patent Literature	NPL11_DATUMTE076A1.PDF	1190109 4dcc22e13ab4548abf784dc49366adaf63d1ecbe	no	5
Warnings:					
Information:					
20	Non Patent Literature	NPL12_DATUMTE076A1.PDF	2161507 333cc067211ed8a22c8356f406c5a70081c29242	no	5
Warnings:					
Information:					
21		IDS_DATUMTE076A1.pdf	305304 3392817919cbef66c18ba6a64c44311d0390f3ff	yes	5

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Transmittal Letter	1	1
	Information Disclosure Statement (IDS) Form (SB08)	2	5

Warnings:	
Information:	
Total Files Size (in bytes):	17449934

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DATUMTE.076A1
	Application Number	
Title of Invention	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Applicant Information:

Applicant 1

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Philip	L.	Northcott	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Coquitlam	Country Of Residence	CA
Citizenship under 37 CFR 1.41(b)	CA		

Mailing Address of Applicant:

Address 1	815 Northview Place		
Address 2			
City	Coquitlam	State/Province	BC
Postal Code	V3J 3R4	Country	CA

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	20995		
Email Address	efiling@kmob.com	Add Email	Remove Email

Application Information:

Title of the Invention	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE		
Attorney Docket Number	DATUMTE.076A1	Small Entity Status Claimed	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Suggested Class (if any)		Sub Class (if any)	
Suggested Technology Center (if any)			
Total Number of Drawing Sheets (if any)	16	Suggested Figure for Publication (if any)	\\

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DATUMTE.076A1
	Application Number	
Title of Invention	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	

Publication Information:

☐	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☒	**Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S. C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	20995		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.

Prior Application Status		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button.

Foreign Priority Information:

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).

			Remove
Application Number	Country[1]	Parent Filing Date (YYYY-MM-DD)	Priority Claimed
			◉ Yes ○ No

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Assignee Information:

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37 of the CFR to have an assignment recorded in the Office.

Assignee 1

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	DATUMTE.076A1
	Application Number	
Title of Invention	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	

If the Assignee is an Organization check here. ☒			
Organization Name	PMC-Sierra, Inc.		
Mailing Address Information:			
Address 1	Mission Tower One, 3975 Freedom Circle		
Address 2			
City	Santa Clara	**State/Province**	CA
Country	US	Postal Code	95054
Phone Number		Fax Number	
Email Address			
Additional Assignee Data may be generated within this form by selecting the **Add** button.			

Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.					
Signature	Michael Okamoto			Date (YYYY-MM-DD)	2012-05-22
First Name	Michael	Last Name	Okamoto	Registration Number	47831

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE

Cross-Reference to Related Application

Appendix A

[0001] Appendix A, which forms a part of this disclosure, is a list of commonly owned copending U.S. patent applications. Each one of the applications listed in Appendix A is hereby incorporated herein in its entirety by reference thereto.

Background

Field of the Invention

[0002] Embodiments of the invention generally relate to electronics, and in particular, to flash drive memory controllers, such as to solid-state drive memory controllers.

Description of the Related Art

[0003] Flash memory is a form of non-volatile memory. A memory cell in flash memory can be a single-level cell (SLC), which encodes one bit of information per cell, or a multi-level cell (MLC), which encodes two or more bits of information per memory cell. Typically, a flash memory implementation using MLC is much cheaper than a flash memory implementation with SLC. Further, a flash memory device is arranged into pages and blocks. Data can be written to and read from flash memory in pages. A group of pages known as a block corresponds to the smallest erasable unit of flash memory.

[0004] Over time, programming and erasing flash memory causes a variety of defects that degrade the performance of the flash memory cell. In particular, MLC memory cells have much lower program/erase cycle lifetimes than SLC memory cells, which can be a problem in an enterprise application. This degradation, along with other noise effects, cause the signal-to-noise ratio of the memory cell to change over time. After the signal-to-noise ratio has fallen to a certain level, the flash memory device is typically no longer reliable.

Manufacturers typically specify a number of program/erase cycles over which the properties of their flash devices are guaranteed.

[0005] As flash memory technologies become denser with decreasing process technology, the amount of charge stored on a floating gate of a memory cell tends to fall, crosstalk between cells tends to rise, insulation material between memory cells become thinner, and so on. Taken together, these effects tend to cause the signal-to-noise ratio of flash memory to decrease with each passing generation.

[0006] Flash memory devices require the use of a form of Error Correction Coding (ECC) to detect and correct the errors that inevitably occur. ECC, and in particular the ubiquitous Reed-Solomon or the Bose, Chaudhuri, and Hocquenghem (BCH) hard-decision codes, are widely used in electronics as a way of mitigating low signal-to-noise ratio in communications and storage media. With ECC, redundant information is stored or transmitted alongside the regular information bearing data, to permit an ECC decoder to deduce the originally transmitted or stored information even in the presence of errors.

[0007] A conventional approach to error management for MLC flash memories has been for the flash memory chip manufacturer to specify a particular strength of ECC code, typically a one-dimensional BCH code, capable of correcting a certain number of bits per certain size of sector, for example 24 bits per 1024 bytes. Examples of allocations of flash bytes in a typical flash page using vendor-specified BCH are shown in Figures 1A and 1B. Figure 1A illustrates a standard layout for a page of flash memory. A region is provided for user data 101 or information data in the main area of the flash page, and a spare area 102 is provided to provide room for ECC parity and other data. In practice, the user data 103, metadata 104, such as data integrity feature (DIF) and Journaling Engine management data, and the manufacturer's recommended amount of ECC parity data 105 may be stored interleaved within the page as illustrated in Figure 1B.

[0008] So long as the specified error correction is provided by the flash controller, the flash memory chip manufacturer guarantees a certain number of Program/Erase (P/E) cycles over which the flash memory chips will store and retain data, with no more errors than the ECC can correct, with a probability of uncorrectable errors occurring less than some acceptable risk for the end user. For example, consumer grade flash-based drives may tolerate a relatively high uncorrectable error rate. However, in an

enterprise storage environment, a relatively low uncorrectable error rate is applicable, for example, $1x10^{-16}$ (1E-16).

[0009] However, conventional approaches of applying ECC to flash memory can be inefficient at achieving relatively low uncorrectable error rates over the total service life and traffic loads that can be required in an enterprise storage environment, such as in a server.

Summary

[0010] Disclosed techniques provide relatively low uncorrectable bit error rates for flash memory, low write amplification, long life, fast and efficient retrieval, and efficient storage density such that a solid-state drive (SSD) or flash drive can be implemented using relatively inexpensive MLC flash for an enterprise storage application.

[0011] In one embodiment, a three-dimensional turbo code, built out of three partially orthogonal ECC components is as follows. In a first dimension, a set of "primary" error correction codes, such as BCH codes, provides error correction for a flash page. In a second dimension, a set of "secondary" error correction codes, which are at least partly orthogonal to the primary codes, provides error correction for a "page stripe," which can be a group of pages that are associated by secondary error correction codes. A page stripe corresponds to a set of *Ws* flash pages (a page stripe of width *Ws*), wherein each page stripe includes one page on each of the *Ws* flash pages. Preferably, each flash page of a page stripe is from a different semiconductor die or different flash memory array. A value for the width *Ws* can vary in a very broad range. In one embodiment, the width *Ws* is 4, however, other applicable widths will be readily determined by one of ordinary skill in the art. For example, the value can be 2, 3, 5-8, or the like. In one embodiment, the *Ws* flash dice are arranged so that they can be efficiently accessed in parallel.

[0012] In a third dimension, a set of tertiary Reed-Solomon (RS) codes having codewords orthogonal to the codewords of both the primary code and secondary code, provides error correction for a "page grid," which can be a group of pages that are associated by tertiary error correction codes. A page grid corresponds to a set of *Wg* page stripes (a page grid of width *Wg*), wherein each page grid includes *Wg* page stripes with each page of the page grid preferably on different sets of flash dice or flash memory arrays, for a total of

one flash page from each of the product of width *Ws* and width *Wg* flash dice. In one embodiment, the width *Wg* is 4, however, other applicable widths will be readily determined by one of ordinary skill in the art. For example, for the case in which each of the width *Ws* and the width *Wg* are 4, then there would be 16 flash pages in the page grid. The primary, secondary, and tertiary codes can be considered to be an outer code, a first inner code, and a second inner code, respectively.

[0013] In one embodiment, a journaling filing system or flash translation layer is used to reduce write amplification, thereby extending flash memory cell life. As or after the journaling file system receives data (information) for storage in flash memory cells, that data is used to calculate primary parity, secondary parity, and tertiary parity. For efficiency, the calculation of the parity bits is preferably performed in parallel with the writing of the data to the flash memory cells.

[0014] Data may be received by the journaling file system in a relatively large data stream. When stored, this data stream is allocated among a plurality of payloads of codewords. In the illustrated embodiment, an ECC encoder/decoder generates the parity bits for the primary parity, the secondary parity, and the tertiary parity. As discussed earlier, the terms primary, secondary, and tertiary refer to orthogonal dimensions over which block data is taken for calculation of ECC parity bits. In its simplest form, the block data can correspond to user data, that is, information, but as explained below, with other parity bits, can include parity bits of earlier calculated ECC.

[0015] In one embodiment, primary parity information is inserted into the stored data stream at the end of each primary codeword such that a primary codeword typically has, but does not need to have, contiguous symbols. In the illustrated embodiment, secondary parity is calculated over and inserted at the end of each page stripe (but before the tertiary parity). Generally, the secondary parity symbols are not contiguous with the secondary information symbols. A second primary parity that is calculated on the secondary parity bits is inserted at the end of the secondary parity. Tertiary parity is calculated over a page grid (groups of *Wg* page stripes), and is inserted into a reserved tertiary space at the end of the last page stripe. In this context, "last" means the last in time page stripe of the page grid to store data. The last page stripe can also be termed the tertiary page stripe. Generally, the tertiary

parity symbols are not contiguous with the tertiary information symbols. Primary parity is calculated on the tertiary parity bits (PoTP) and inserted after the tertiary parity bits.

[0016] The result is that the flash memory is programmed as a set of page grids, with each piece of user data being protected by a three-dimensional grid of error correction codes. Each error correction code requires the addition of redundant data, which reduces the amount of user data that can be stored in the grid. The stronger the correction code, the more redundant data is required. The strength of the error correcting code can be selected on a per-page stripe basis. Preferably, the selected strength for the code can be the weakest one that is still certain to be able to correct the bit error rate anticipated on that page stripe.

[0017] When reading data, the data of interest is usually less than a page stripe. In accordance with one embodiment of the invention, the data of interest is read from the flash memory along with enough surrounding data to make up a set of complete primary codewords (a partial page read). If the primary decode process alone is sufficient to correct the data with high confidence, the operation is complete. If not, the entire page stripe containing the data of interest will be read and processed.

[0018] When decoding a page stripe, the primary and secondary codes can be used in an iterative fashion as necessary. If this is sufficient to correct the data with high confidence, the ECC decoding operation is complete and the data is released to the file system. If not, then the remaining three page stripes of the page grid are read.

[0019] When decoding a page grid, the primary and secondary parity of the *Wg*, for example, 4, page stripes are used to correct those page stripes to the extent practical, according to the principles of turbo decoding, which can include iterative decoding. After this is complete, the tertiary codewords over the page grid will be decoded iteratively with the primary and secondary, according to the principles of three-dimensional turbo decoding. If this is sufficient to correct the data with high confidence, the decoding operation is complete and the data is released to the file system. If not, then a final procedure can be attempted, which allows the algorithm to recover from the massive failure of one page in the grid. If that fails, then ECC decoding has failed, and a read response is sent indicating that the user data was uncorrectable.

[0020] One embodiment includes a method of selecting an error correction coding (ECC) scheme, wherein the method includes: determining a bit error rate associated with a

region associated with a plurality of flash memory pages or integer fractions thereof; comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein the first gear has a different data payload size and correction capability than the second gear; and selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds; wherein determining, comparing, and selecting are performed by an integrated circuit. Determining, comparing, and selecting can be performed concurrently with at least one of flash initialization, read, or write operations for an end user. However, these determining, comparing, and selecting activities do not need to be done each time there is an initialization, read, or write operation.

[0021] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to determine a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof and to compare the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes, wherein the first gear has a different data payload size and correction capability than the second gear; and a second circuit configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

[0022] One embodiment includes an apparatus for selecting an error correction coding (ECC) scheme, wherein the apparatus includes: a means for determining a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof; a means for comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein the first gear has a different data payload size and correction capability than the second gear; and a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

[0023] One embodiment includes an method of selecting among gears, wherein different gears correspond to different error correction coding (ECC) schemes for regions of

a flash drive, wherein the method includes: performing a memory test to determine raw bit error rates among the regions of the flash drive; retiring regions having a raw bit error rate (RBER) above a selected threshold; and for remaining regions, allocating one or more regions to one of at least a first gear or a second gear based at least partly on the RBER for the region, wherein the first gear and the second gear vary by data payload size and correction capability; wherein the memory test, retiring regions, and allocating are performed by an integrated circuit; wherein the memory test, retiring regions, and allocating are performed concurrently with initialization operations for an end user.

[0024] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to run a memory test to determine raw bit error rates among regions of a flash drive; and a second circuit configured to: retire regions having a raw bit error rate (RBER) above a selected threshold; and allocate one or more remaining regions to one of at least a first gear or a second gear based at least partly on the RBER for the region, wherein the first gear and the second gear vary by data payload size and correction capability, wherein different gears correspond to different error correction coding (ECC) schemes for regions of the flash drive; wherein the first circuit is configured to run the memory test concurrently with initialization operations for an end user.

[0025] One embodiment includes an apparatus for selecting among gears, wherein different gears correspond to different error correction coding (ECC) schemes for regions of a flash drive, wherein the apparatus includes: a means for performing a memory test to determine raw bit error rates among the regions of the flash drive, wherein the memory test performing means is configured to run the memory test concurrently with initialization operations for an end user; a means for retiring regions having a raw bit error rate (RBER) above a selected threshold; and a means for allocating one or more remaining regions to one of at least a first gear or a second gear based at least partly on the RBER for the region, wherein the first gear and the second gear vary by data payload size and correction capability.

[0026] One embodiment includes a method of managing one or more gears, wherein different gears correspond to different error correction code (ECC) schemes, wherein the method includes: receiving a request for user data stored in flash memory devices, wherein the user data can be in stored in a scrambled from or in an unscrambled

form; retrieving one or more codewords containing the requested user data from a region of the flash memory devices, wherein the region is associated with a first gear; correcting the one or more codewords to detect and correct errors with the first gear; and deciding whether or not to select a different gear for the region or retiring the region based on at least one of an analysis of an approximate raw bit error rate (RBER) of the region or a number of decoding iterations used to correct the requested user data, wherein the different gear has greater error correction capacity than the first gear; wherein at least correcting and deciding are performed by an integrated circuit.

[0027] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to: receive a request for user data stored in flash memory devices, wherein the user data can be in stored in a scrambled from or in an unscrambled form; and retrieve one or more codewords containing the requested user data from a region of the flash memory devices, wherein the region is associated with a first gear, wherein different gears correspond to different error correction code (ECC) schemes; and a second circuit configured to correct the one or more codewords to detect and correct errors with the first gear; wherein the first circuit is configured to decide whether or not to select a different gear for the region or retire the region based on at least one of an analysis of an approximate raw bit error rate (RBER) of the region or a number of decoding iterations used to correct the requested user data, wherein the different gear has greater error correction capacity than the first gear.

[0028] One embodiment includes an apparatus of managing one or more gears, wherein different gears correspond to different error correction code (ECC) schemes, wherein the apparatus includes: a means for receiving a request for user data stored in flash memory devices, wherein the user data can be in stored in a scrambled from or in an unscrambled form; a means for retrieving one or more codewords containing the requested user data from a region of the flash memory devices, wherein the region is associated with a first gear; a means for correcting the one or more codewords to detect and correct errors with the first gear; and a means for deciding whether or not to select a different gear for the region or retiring the region based on at least one of an analysis of an approximate raw bit error rate (RBER) of the region or a number of decoding iterations used to correct the requested user data, wherein the different gear has greater error correction capacity than the first gear.

[0029] One embodiment includes a method of programming data to a plurality of flash memory devices, wherein the method includes: receiving data to be programmed; arranging the received data for distribution among a group of pages such that sequentially received data is to be programmed among the group of pages, wherein the group of pages comprises two or more flash memory pages or two or more integer fractions of flash memory pages; error correction encoding data to be stored within a particular page or integer fraction to generate primary parity symbols for primary codewords; generating information portions of secondary codewords by selecting portions of primary codewords, wherein secondary codewords are at least partially orthogonal to primary codewords; error correction encoding the information portions of the secondary codewords to generate secondary parity symbols for the secondary codewords; arranging the secondary parity data for distribution among the group of pages such that the secondary parity data is to be programmed among the group of pages; and programming the group of pages with the arranged received data, the primary parity data, and the arranged secondary parity data; wherein at least error correction encoding data, generating information portions, and error correction encoding the information portions are performed by an integrated circuit; wherein at least error correction encoding data, generating information portions, and error correction encoding the information portions are performed concurrently with write operations for an end user.

[0030] One embodiment includes an apparatus, wherein the method includes: a first circuit configured to: receive data to be programmed; and arrange the received data for distribution among a group of pages such that sequentially received data is to be programmed among the group of pages, wherein the group of pages comprises two or more flash memory pages or two or more integer fractions of flash memory pages; a second circuit configured to error correction encode data to be stored within a particular page or integer fraction to generate primary parity symbols for primary codewords; the first circuit further configured to select portions of primary codewords to generate information portions of secondary codewords, wherein secondary codewords are at least partially orthogonal to primary codewords; the second circuit further configured to error correction encode the information portions of the secondary codewords to generate secondary parity symbols for the secondary codewords; the first circuit further configured to arrange the secondary parity data for distribution among the group of pages such that the secondary parity data is to be

programmed among the group of pages; and the first circuit further configured to program the group of pages with the arranged received data, the primary parity data, and the arranged secondary parity data; wherein the first circuit and the second circuit are configured to error correction encode data, to generate information portions, and to error correction encode the information portions concurrently with write operations for an end user.

[0031] One embodiment includes an apparatus for programming data to a plurality of flash memory devices, wherein the apparatus includes: a means for receiving data to be programmed; a means for arranging the received data for distribution among a group of pages such that sequentially received data is to be programmed among the group of pages, wherein the group of pages comprises two or more flash memory pages or two or more integer fractions of flash memory pages; a means for error correction encoding data to be stored within a particular page or integer fraction to generate primary parity symbols for primary codewords; a means for generating information portions of secondary codewords by selecting portions of primary codewords, wherein secondary codewords are at least partially orthogonal to primary codewords; the encoding means further configured to error correction encode the information portions of the secondary codewords to generate secondary parity symbols for the secondary codewords; the arranging means further configured to arrange the secondary parity data for distribution among the group of pages such that the secondary parity data is to be programmed among the group of pages; and a means for programming the group of pages with the arranged received data, the primary parity data, and the arranged secondary parity data; wherein the arranging means and the error correction means are configured to operate concurrently with write operations for an end user.

[0032] One embodiment includes a method of decoding data with low latency, wherein the method includes: receiving a request for data stored in at least one of a plurality of flash memory devices; identifying a page stripe that includes the requested data, wherein the page stripe comprises a grouping of one or more equally-sized flash pages or integer fractions of flash pages, wherein each page or each integer fraction of the page stripe contains primary parity symbols for error correction of data stored within the page or integer fraction, and wherein the pages or integer fractions of the page stripe contain secondary parity symbols for error correction of data stored throughout the page stripe; determining a portion of the page stripe that contains an integer number of a first set of primary codewords

containing the requested data and associated primary parity symbols such that retrieval and decoding of the requested data is performed with a first amount of latency; retrieving from flash memory the first set of primary codewords containing the requested data; performing error correction on the first set of primary codewords to determine if the raw requested data contains errors and if the errors, if any, are correctable, to generate first corrected requested data; releasing the first corrected requested data based at least partly on a determination that the first corrected requested data has been decoded with a likelihood of error below a predetermined acceptable uncorrectable bit error rate (aUBER) when the first corrected requested data cannot be considered to be error free to within the aUBER, performing the following: reading the page stripe containing the requested data such that a second set of primary codewords is retrieved, wherein reading of page stripe is performed with a second latency longer than the first latency; performing error correction on the second set of primary codewords of the page stripe to generate an updated page stripe; interleaving data from the updated page stripe to arrange secondary codewords; performing error correction on the secondary codewords to generate corrected secondary codewords, wherein execution of error correction on the secondary codewords is not performed in parallel with execution of error correction on the primary codewords; de-interleaving the corrected secondary codewords to generate a further updated page stripe having an updated second set of primary codewords; performing error correction on at least a portion of the updated second set of primary codewords to generate updated corrected requested data; and releasing the second corrected requested data based at least partly on a determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; wherein at least performing error correction is performed by an integrated circuit; wherein at least performing error correction is performed concurrently with a read operation for an end user.

[0033] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to receive a request for data stored in at least one of a plurality of flash memory devices; the first circuit configured to identify a page stripe that includes the requested data, wherein the page stripe comprises a grouping of one or more equally-sized flash pages or integer fractions of flash pages, wherein each page or each integer fraction of the page stripe contains primary parity symbols for error correction of data stored within the page or integer fraction, and wherein the pages or integer fractions of the page stripe contain

secondary parity symbols for error correction of data stored throughout the page stripe; the first circuit configured to determine a portion of the page stripe that contains an integer number of a first set of primary codewords containing the requested data and associated primary parity symbols such that retrieval and decoding of the requested data is performed with a first amount of latency; the first circuit configured to retrieve from flash memory the first set of primary codewords containing the requested data; a second circuit configured to perform error correction on the first set of primary codewords to determine if the raw requested data contains errors and if the errors, if any, are correctable, to generate first corrected requested data; the first circuit configured to release the first corrected requested data based at least partly on a determination that the first corrected requested data has been decoded with a likelihood of error below a predetermined acceptable uncorrectable bit error rate (aUBER), and when the first corrected requested data cannot be considered to be error free to within the aUBER: the first circuit is configured to read the page stripe containing the requested data such that a second set of primary codewords is retrieved, wherein reading of page stripe is performed with a second latency longer than the first latency; the second circuit is configured to perform error correction on the second set of primary codewords of the page stripe to generate an updated page stripe; the first circuit is configured to interleave data from the updated page stripe to arrange secondary codewords; the second circuit is configured to perform error correction on the secondary codewords to generate corrected secondary codewords, wherein execution of error correction on the secondary codewords is not performed in parallel with execution of error correction on the primary codewords; the first circuit is configured to de-interleave the corrected secondary codewords to generate a further updated page stripe having an updated second set of primary codewords; the second circuit is configured to perform error correction on at least a portion of the updated second set of primary codewords to generate updated corrected requested data; and the first circuit is configured to release the second corrected requested data based at least partly on a determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; wherein the second circuit is configured to perform error correction concurrently with a read operation for an end user.

[0034] One embodiment includes an apparatus for decoding data with low latency, wherein the apparatus includes: a means for receiving a request for data stored in at

least one of a plurality of flash memory devices; a means for identifying a page stripe that includes the requested data, wherein the page stripe comprises a grouping of one or more equally-sized flash pages or integer fractions of flash pages, wherein each page or each integer fraction of the page stripe contains primary parity symbols for error correction of data stored within the page or integer fraction, and wherein the pages or integer fractions of the page stripe contain secondary parity symbols for error correction of data stored throughout the page stripe; a means for determining a portion of the page stripe that contains an integer number of a first set of primary codewords containing the requested data and associated primary parity symbols such that retrieval and decoding of the requested data is performed with a first amount of latency; a means for retrieving from flash memory the first set of primary codewords containing the requested data; a means for performing error correction on the first set of primary codewords to determine if the raw requested data contains errors and if the errors, if any, are correctable, to generate first corrected requested data; a means for interleaving; a means for de-interleaving; a means for releasing the first corrected requested data based at least partly on a determination that the first corrected requested data has been decoded with a likelihood of error below a predetermined acceptable uncorrectable bit error rate (aUBER) when the first corrected requested data cannot be considered to be error free to within the aUBER: the retrieving means is configured to read the page stripe containing the requested data such that a second set of primary codewords is retrieved, wherein reading of page stripe is performed with a second latency longer than the first latency; the error correction means is configured to perform error correction on the second set of primary codewords of the page stripe to generate an updated page stripe; the interleaving means is configured to interleave data from the updated page stripe to arrange secondary codewords; the error correction means configured to perform error correction on the secondary codewords to generate corrected secondary codewords, wherein execution of error correction on the secondary codewords is not performed in parallel with execution of error correction on the primary codewords; the de-interleaving means is configured to de-interleave the corrected secondary codewords to generate a further updated page stripe having an updated second set of primary codewords; the error correction means configured to perform error correction on at least a portion of the updated second set of primary codewords to generate updated corrected requested data; and the releasing means configured to release the second

corrected requested data based at least partly on a determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; wherein at least performing error correction is performed concurrently with a read operation for an end user.

[0035] One embodiment includes a method of reading data, wherein the method includes: receiving a request for data stored in at least one of a plurality of flash memory devices; identifying a page stripe that includes at least a portion of the requested data, wherein the page stripe comprises a grouping of one or more equally-sized flash pages or integer fractions of pages of flash pages, wherein each page or each integer fraction of the page stripe contains primary parity symbols for error correction of data stored within the page or integer fraction, and wherein the pages or integer fractions of the page stripe contain secondary parity symbols for error correction of data stored throughout the page stripe; retrieving contents of the page stripe in raw form; performing error correction on the primary codewords of the page stripe to generate an updated page stripe containing corrected requested data; determining whether or not at least the corrected requested data contains error; releasing the corrected requested data based at least partly on a determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; when the first corrected requested data is not released, performing the following: interleaving data from the updated page stripe to arrange secondary codewords; performing error correction decoding on the secondary codewords to generate updated secondary codewords, wherein execution of error correction on the secondary codewords is not performed in parallel with execution of error correction on the primary codewords; de-interleaving the corrected secondary codewords to generate a further updated page stripe having an updated primary codewords; performing error correction on at least a portion of the updated primary codewords to generate updated corrected requested data; and releasing the updated corrected requested data based at least partly on the determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; wherein at least performing error correction is performed by an integrated circuit; wherein at least performing error correction is performed concurrently with a read operation for an end user.

[0036] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to receive a request for data stored in at least one of a plurality of flash memory devices; the first circuit configured to identify a page stripe that includes at least a portion of the requested data, wherein the page stripe comprises a grouping of one or more equally-sized flash pages or integer fractions of pages of flash pages, wherein each page or each integer fraction of the page stripe contains primary parity symbols for error correction of data stored within the page or integer fraction, and wherein the pages or integer fractions of the page stripe contain secondary parity symbols for error correction of data stored throughout the page stripe; the first circuit configured to retrieve contents of the page stripe in raw form; a second circuit configured to perform error correction on the primary codewords of the page stripe to generate an updated page stripe containing corrected requested data; the first circuit configured to determine whether or not at least the corrected requested data contains error; the first circuit configured to release the corrected requested data based at least partly on a determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; when the first corrected requested data is not released: the first circuit is configured to interleave data from the updated page stripe to arrange secondary codewords; the second circuit is configured to perform error correction decoding on the secondary codewords to generate updated secondary codewords, wherein execution of error correction on the secondary codewords is not performed in parallel with execution of error correction on the primary codewords; the first circuit is configured to de-interleave the corrected secondary codewords to generate a further updated page stripe having an updated primary codewords; the second circuit is configured to perform error correction on at least a portion of the updated primary codewords to generate updated corrected requested data; and the first circuit is configured to release the updated corrected requested data based at least partly on the determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; wherein the second circuit is configured to perform error correction concurrently with a read operation for an end user.

[0037] One embodiment includes an apparatus for reading data, wherein the apparatus includes: a means for receiving a request for data stored in at least one of a plurality of flash memory devices; a means for identifying a page stripe that includes at least

a portion of the requested data, wherein the page stripe comprises a grouping of one or more equally-sized flash pages or integer fractions of pages of flash pages, wherein each page or each integer fraction of the page stripe contains primary parity symbols for error correction of data stored within the page or integer fraction, and wherein the pages or integer fractions of the page stripe contain secondary parity symbols for error correction of data stored throughout the page stripe; a means for retrieving contents of the page stripe in raw form; a means for performing error correction on the primary codewords of the page stripe to generate an updated page stripe containing corrected requested data; a means for determining whether or not at least the corrected requested data contains error; a means for interleaving; a means for de-interleaving; a means for releasing the corrected requested data based at least partly on a determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; when the first corrected requested data is not released: the interleaving means is configured to interleave data from the updated page stripe to arrange secondary codewords; the error correction means is configured to perform error correction decoding on the secondary codewords to generate updated secondary codewords, wherein execution of error correction on the secondary codewords is not performed in parallel with execution of error correction on the primary codewords; the de-interleaving means is configured to de-interleave the corrected secondary codewords to generate a further updated page stripe having an updated primary codewords; the error correction means is configured to perform error correction on at least a portion of the updated primary codewords to generate updated corrected requested data; and the releasing means is configured to release the updated corrected requested data based at least partly on the determination that the likelihood of the requested data containing an error is less than a predetermined acceptable failure rate; wherein the error correction means is configured to perform error correction concurrently with a read operation for an end user.

[0038] One embodiment includes a method of storing data to an array of flash memory devices, wherein the method includes: maintaining one or more lists of page stripes indicating at least which page stripes are available to be programmed, the one or more lists of page stripes associating, for an available page stripe, the following: one or more physical addresses of pages or integer fractions of pages for the page stripe; and a gear setting for the page stripe, the gear setting indicating an error correction scheme and corresponding payload

size assigned to the page stripe; maintaining one or more lists associating at least virtual addresses with physical addresses for data stored within the array of flash memory devices, wherein a virtual address is associated with at least one physical address; receiving an allocation block, a virtual address identifying the allocation block, and a request for storage of the allocation block into the array of flash memory devices; determining, based at least partly on the gear assigned to a current page stripe and an amount of data, if any, already assigned to the current page stripe, an extent to which the allocation block can be stored within an available payload of the current page stripe, wherein the current page stripe corresponds to a page stripe selected from the one or more lists of page stripes that are available to be programmed; assigning at least a first subset of the allocation block to the payload of the selected page stripe; storing an association of the virtual address with (a) the physical address of the current page stripe, and with (b) a starting location of the allocation block within the current page stripe; when enough data has been collected to consider the payload of the current page stripe as full: performing or having error correction encoding performed for the data of the payload in accordance with the gear setting associated with the current page stripe; programming the current page stripe to the array; associating the current page stripe with programmed page stripes; selecting a next page stripe from the one or more lists of available page stripes for subsequent programming; if the allocation block had not been fully assigned to the payload of the current page stripe: assigning a remainder of the allocation block to a payload of the next page stripe; and storing an additional association of the virtual address with the physical address of the next page stripe; wherein the method is performed by an integrated circuit.

[0039] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to maintain one or more lists of page stripes indicating at least which page stripes are available to be programmed, the one or more lists of page stripes associating, for an available page stripe, the following: one or more physical addresses of pages or integer fractions of pages for the page stripe; and a gear setting for the page stripe, the gear setting indicating an error correction scheme and corresponding payload size assigned to the page stripe; the first circuit further configured to maintain one or more lists associating at least virtual addresses with physical addresses for data stored within an array of flash memory devices, wherein a virtual address is associated with at least one physical address; the first

circuit further configured to receive an allocation block, a virtual address identifying the allocation block, and a request for storage of the allocation block into the array of flash memory devices; the first circuit further configured to determine, based at least partly on the gear assigned to a current page stripe and an amount of data, if any, already assigned to the current page stripe, an extent to which the allocation block can be stored within an available payload of the current page stripe, wherein the current page stripe corresponds to a page stripe selected from the one or more lists of page stripes available for programming; the first circuit further configured to assign at least a first subset of the allocation block to the payload of the selected page stripe; the first circuit further configured to store an association of the virtual address with (a) the physical address of the current page stripe, and with (b) a starting location of the allocation block within the current page stripe; when enough data has been collected to consider the payload of the current page stripe as full: a second circuit is configured to perform error correction encoding performed for the data of the payload in accordance with the gear setting associated with the current page stripe; the first circuit is configured to program the current page stripe to the array; the first circuit is configured to associate the current page stripe with programmed page stripes; the first circuit is configured to select a next page stripe from the one or more lists of available page stripes for subsequent programming; if the allocation block had not been fully assigned to the payload of the current page stripe: the first circuit is configured to assign a remainder of the allocation block to a payload of the next page stripe; and the first circuit is configured to store an additional association of the virtual address with the physical address of the next page stripe.

[0040] One embodiment includes an apparatus for storing data to an array of flash memory devices, wherein the apparatus includes: a means for maintaining one or more lists of page stripes indicating at least which page stripes are available to be programmed, the one or more lists of page stripes associating, for an available page stripe, the following: one or more physical addresses of pages or integer fractions of pages for the page stripe; and a gear setting for the page stripe, the gear setting indicating an error correction scheme and corresponding payload size assigned to the page stripe; the maintaining means further maintaining one or more lists associating at least virtual addresses with physical addresses for data stored within the array of flash memory devices, wherein a virtual address is associated with at least one physical address; a means for receiving an allocation block, a

virtual address identifying the allocation block, and a request for storage of the allocation block into the array of flash memory devices; a means for determining, based at least partly on the gear assigned to a current page stripe and an amount of data, if any, already assigned to the current page stripe, an extent to which the allocation block can be stored within an available payload of the current page stripe, wherein the current page stripe corresponds to a page stripe selected from the one or more lists of page stripes that are available to be programmed; a means for assigning at least a first subset of the allocation block to the payload of the selected page stripe; a means for storing an association of the virtual address with (a) the physical address of the current page stripe, and with (b) a starting location of the allocation block within the current page stripe; when enough data has been collected to consider the payload of the current page stripe as full: a means for performing or having error correction encoding performed for the data of the payload in accordance with the gear setting associated with the current page stripe; a means for programming the current page stripe to the array; a means for associating the current page stripe with programmed page stripes; a means for selecting a next page stripe from the one or more lists of available page stripes for subsequent programming; if the allocation block had not been fully assigned to the payload of the current page stripe: the assigning means further assigning a remainder of the allocation block to a payload of the next page stripe; and the association storing means further storing an additional association of the virtual address with the physical address of the next page stripe.

[0041] One embodiment includes a method of reading data from an array of flash memory devices, wherein the method includes: receiving a request for data stored in the flash array, the requested data being identified by a virtual address; looking up the following, which are associated with the virtual address: a physical address or addresses of one or more page stripes; a starting location for the data within the first page stripe of the one or more page stripes of the flash array; and one or more gear settings for the one or more page stripes, wherein the gear settings are indicative of a strength of error correction assigned to a page stripe of the one or more page stripes; requesting one or more portions of the payload of the one or more page stripes associated with the virtual address, and providing the one or more gear settings of the one or more page stripes along with the request; receiving, from an ECC decoder, corrected retrieved portions of the one or more page stripes, and an indication of

whether each portion was successfully corrected; and if all portions of the requested data were successfully corrected by the ECC decoder, reassembling the corrected portions into the requested data and returning the same; wherein the method is performed by an integrated circuit.

[0042] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to receive a request for data stored in a flash array, the requested data being identified by a virtual address; a second circuit configured to look up the following, which are associated with the virtual address: a physical address or addresses of one or more page stripes; a starting location for the data within the first page stripe of the one or more page stripes of the flash array; and one or more gear settings for the one or more page stripes, wherein the gear settings are indicative of a strength of error correction assigned to a page stripe of the one or more page stripes; the second circuit further configured to request one or more portions of the payload of the one or more page stripes associated with the virtual address, and providing the one or more gear settings of the one or more page stripes along with the request; the second circuit further configured to receive, from an ECC decoder, corrected retrieved portions of the one or more page stripes, and an indication of whether each portion was successfully corrected; and if all portions of the requested data were successfully corrected by the ECC decoder, the second circuit is further configured to reassemble the corrected portions into the requested data and return the same.

[0043] One embodiment includes an apparatus for reading data from an array of flash memory devices, wherein the apparatus includes: a means for receiving a request for data stored in a flash array, the requested data being identified by a virtual address; a means for looking up the following, which are associated with the virtual address: a physical address or addresses of one or more page stripes; a starting location for the data within the first page stripe of the one or more page stripes of the flash array; and one or more gear settings for the one or more page stripes, wherein the gear settings are indicative of a strength of error correction assigned to a page stripe of the one or more page stripes; a means for requesting one or more portions of the payload of the one or more page stripes associated with the virtual address; a means for providing the one or more gear settings of the one or more page stripes along with the request; a means for receiving, from an ECC decoder, corrected retrieved portions of the one or more page stripes, and an indication of whether

each portion was successfully corrected; and a means for reassembling; wherein if all portions of the requested data were successfully corrected by the ECC decoder, the reassembling means further reassembling the corrected portions into the requested data and returning the same.

[0044] One embodiment includes a method of reclaiming flash blocks of a flash array, wherein the method includes: selecting a block grid for reclamation, wherein block grids contain complete page grids; for valid data stored in the block grid, reading, performing error correction, and journaling into available page stripes of the flash array, wherein valid data includes valid allocation blocks, wherein an ECC decoder used for performing error correction generates an indication as to whether or not correction for at least the valid data from the flash array has met criteria indicating that a different gear to be selected; erasing the blocks of the block grid; updating gear settings for the page stripes of the block grid based at least partly on corresponding indications from the ECC decoder; and making the page stripes of the block grid available for writing; wherein at least selecting and making are performed by an integrated circuit.

[0045] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to select a block grid for reclamation, wherein block grids contain complete page grids; an ECC decoder; for valid data stored in the block grid, the first circuit is further configured to read, the ECC decoder is configured to perform error correction, and the first circuit is further configured to journal into available page stripes of a flash array, wherein valid data includes valid allocation blocks, wherein the ECC decoder generates an indication as to whether or not correction for at least the valid data from the flash array has met criteria indicating that a different gear to be selected; the first circuit further configured to erase the blocks of the block grid; the first circuit further configured to update gear settings for the page stripes of the block grid based at least partly on corresponding indications from the ECC decoder; and the first circuit further configured to make the page stripes of the block grid available for writing.

[0046] One embodiment includes an apparatus for reclaiming flash blocks of a flash array, wherein the apparatus includes: a means for selecting a block grid for reclamation, wherein block grids contain complete page grids; a means for reading; a means for ECC decoding; a means for journaling; for valid data stored in the block grid, the reading

means further reading, the ECC decoding means further performing error correction, and the journaling means further journaling into available page stripes of the flash array, wherein valid data includes valid allocation blocks, wherein the ECC decoding means is configured to generate an indication as to whether or not correction for at least the valid data from the flash array has met criteria indicating that a different gear to be selected; a means for erasing the blocks of the block grid; a means for updating gear settings for the page stripes of the block grid based at least partly on corresponding indications from the ECC decoder; and a means for making the page stripes of the block grid available for writing.

[0047] One embodiment includes a method of programming data to a plurality of flash memory devices, wherein the method includes: associating two or more flash memory pages or integer fractions of flash memory pages from one or more integrated circuits or planes with a page stripe, wherein a page is associated with a set of primary ECC codewords, and a page stripe can optionally be associated with a set of secondary and primary over secondary parity (PoSP) ECC codewords; associating two or more flash memory stripes from different integrated circuits or planes with a page grid, wherein the page grid is associated with a group of tertiary ECC codewords, wherein the tertiary ECC is usable as an erasure code, and wherein a last page stripe of the page grid has a reduced payload capacity; receiving data to be programmed; error correction encoding the received data to be stored within each page stripe to generate at least primary ECC codewords for the pages or integer fractions of the page stripe; calculating tertiary ECC parity across the page grid in a sequence such that a tertiary ECC codeword includes a number of symbols residing on every page no greater than the erasure capability of the tertiary ECC codeword; arranging the tertiary parity data such that the tertiary parity data is broken up and spread for programming across a last page stripe; and programming the pages of the page grid; wherein at least performing error correction encoding is performed in an integrated circuit.

[0048] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to associate two or more flash memory pages or integer fractions of flash memory pages from one or more integrated circuits or planes with a page stripe, wherein a page is associated with a set of primary ECC codewords, and a page stripe can optionally be associated with a set of secondary and primary over secondary parity (PoSP) ECC codewords; the first circuit is further configured to associate two or more flash memory

stripes from different integrated circuits or planes with a page grid, wherein the page grid is associated with a group of tertiary ECC codewords, wherein the tertiary ECC is usable as an erasure code, and wherein a last page stripe of the page grid has a reduced payload capacity; the first circuit is further configured to receive data to be programmed; a second circuit configured to error correction encode the received data to be stored within each page stripe to generate at least primary ECC codewords for the pages or integer fractions of the page stripe; the second circuit is further configured to calculate tertiary ECC parity across the page grid in a sequence such that a tertiary ECC codeword includes a number of symbols residing on every page no greater than the erasure capability of the tertiary ECC codeword; and the first circuit is further configured to arrange the tertiary parity data such that the tertiary parity data is broken up and spread for programming across a last page stripe; and the first circuit is further configured to program the pages of the page grid such that data is programmed to a plurality of flash memory devices.

[0049] One embodiment includes an apparatus for programming data to a plurality of flash memory devices, wherein the apparatus includes: a means for associating two or more flash memory pages or integer fractions of flash memory pages from one or more integrated circuits or planes with a page stripe, wherein a page is associated with a set of primary ECC codewords, and a page stripe can optionally be associated with a set of secondary and primary over secondary parity (PoSP) ECC codewords; a means for associating two or more flash memory stripes from different integrated circuits or planes with a page grid, wherein the page grid is associated with a group of tertiary ECC codewords, wherein the tertiary ECC is usable as an erasure code, and wherein a last page stripe of the page grid has a reduced payload capacity; a means for receiving data to be programmed; a means for error correction encoding the received data to be stored within each page stripe to generate at least primary ECC codewords for the pages or integer fractions of the page stripe; a means for calculating tertiary ECC parity across the page grid in a sequence such that a tertiary ECC codeword includes a number of symbols residing on every page no greater than the erasure capability of the tertiary ECC codeword; a means for arranging the tertiary parity data such that the tertiary parity data is broken up and spread for programming across the last page stripe; and a means for programming the pages of the page grid.

[0050] One embodiment includes a method of recovering data from a failed page of flash memory, wherein the method includes: reading contents of flash pages of a page grid containing the failed page, wherein tertiary parity symbols are distributed among the pages of a last page stripe of the page grid, wherein each of the pages of the page grid have ECC codewords stored therein; (a) performing error correction on each page stripe of the page grid according to the gear with which the page stripe was encoded; (b) arranging the data of the page grid to form tertiary codewords; (c) marking symbols of the tertiary codewords that are associated with the failed page as erasures; (d) performing error correction decoding using mixed error and erasure decoding on the tertiary codewords to update the tertiary codewords; (e) arranging the updated tertiary codewords to generate corrected versions of the page stripes; (f) performing error correction on page stripes of the page grid according to the gear with which each page stripe was encoded; performing one or more iterations of (b) through (f) until a stopping criterion is reached or a determination that the recovered data has been recovered without error; and releasing the recovered data to the requester based at least partly on the determination that the recovered data has been recovered without error; wherein at least (a) is performed in an integrated circuit.

[0051] One embodiment includes an apparatus, wherein the apparatus includes: a first circuit configured to read contents of flash pages of a page grid containing a failed page of flash memory, wherein tertiary parity symbols are distributed among the pages of a last page stripe of the page grid, wherein each of the pages of the page grid have ECC codewords stored therein; a second circuit configured to (a) perform error correction on each page stripe of the page grid according to the gear with which the page stripe was encoded; the first circuit configured to (b) arrange the data of the page grid to form tertiary codewords; the first circuit configured to (c) mark symbols of the tertiary codewords that are associated with the failed page as erasures; the second circuit configured to (d) perform error correction decoding with mixed error and erasure decoding on the tertiary codewords to update the tertiary codewords; the first circuit configured to (e) arrange the updated tertiary codewords to generate corrected versions of the page stripes; the second circuit configured to (f) perform error correction on page stripes of the page grid according to the gear with which each page stripe was encoded; wherein the first circuit and the second circuit are configured to iterate performance of (b) through (f) until a stopping criterion is reached or a determination that the

recovered data has been recovered without error; and the first circuit configured to release the recovered data to the requester based at least partly on the determination that the recovered data has been recovered without error.

[0052] One embodiment includes an apparatus for recovering data from a failed page of flash memory, wherein the apparatus includes: a means for reading contents of flash pages of a page grid containing the failed page, wherein tertiary parity symbols are distributed among the pages of a last page stripe of the page grid, wherein each of the pages of the page grid have ECC codewords stored therein; a means for recovering data, the recovering data means further: (a) performing error correction on each page stripe of the page grid according to the gear with which the page stripe was encoded; (b) arranging the data of the page grid to form tertiary codewords; (c) marking symbols of the tertiary codewords that are associated with the failed page as erasures; (d) performing error correction decoding using mixed error and erasure decoding on the tertiary codewords to update the tertiary codewords; (e) arranging the updated tertiary codewords to generate corrected versions of the page stripes; (f) performing error correction on page stripes of the page grid according to the gear with which each page stripe was encoded; wherein the recovering data means further performing one or more iterations of (b) through (f) above until a stopping criterion is reached or a determination that the recovered data has been recovered without error; and a means for releasing the recovered data to the requester based at least partly on the determination that the recovered data has been recovered without error.

Brief Description of the Drawings

[0053] These drawings and the associated description herein are provided to illustrate specific embodiments of the invention and are not intended to be limiting.

[0054] Figures 1A and 1B illustrate a conventional flash page arrangement.

[0055] Figure 2 illustrates a flash programming process according to an embodiment of the invention.

[0056] Figure 3 illustrates an overview of a decoding process.

[0057] Figure 4 illustrates an example of storage of user data in a flash page according to an embodiment of the invention.

[0058] Figure 5 illustrates an example of an arrangement for flash page stripe other than the last page stripe according to an embodiment of the invention.

[0059] Figure 6 illustrates an example of a flash page grid arrangement according to an embodiment of the invention.

[0060] Figure 7 illustrates an example of a flash page stripe arrangement for the last page stripe of a page grid according to an embodiment of the invention.

[0061] Figure 8 illustrates an example of a decoding process according to an embodiment of the invention.

[0062] Figure 9 illustrates an example of decoding a supporting page stripe according to an embodiment of the invention.

[0063] Figure 10 illustrates an example of a process for recovery of a failed page.

[0064] Figure 11 illustrates an example of a flash page ECC gear progression.

[0065] Figure 12 illustrates the operation of a virtual/physical translation table, which is used by the journaling engine of Figure 13.

[0066] Figure 13 illustrates an example of a block diagram for a solid-state drive controller according to an embodiment of the invention.

[0067] Figure 14 illustrates a process that a journaling engine can perform to execute write requests.

[0068] Figure 15 illustrates a process that a journaling engine can perform to execute read requests.

[0069] Figure 16 describes a process that a journaling engine can perform to reclaim regions of flash, making them available for writing

[0070] In this description, reference is made to the drawings in which like reference numerals indicate identical or functionally similar elements.

Detailed Description of Specific Embodiments

[0071] Although particular embodiments are described herein, other embodiments of the invention, including embodiments that do not provide all of the benefits and features set forth herein, will be apparent to those of ordinary skill in the art. For example, one embodiment of the invention corresponds to an SSD memory controller or a flash drive controller with one or more of the following features: (1) translation of logical or virtual

addresses and real addresses via a journaling file system or flash translation layer to reduce the number of program/erase cycles to the memory cells (see Figure 12); (2) selection of a strength of error correction coding based on a raw bit error rate (see Figure 11); (3) provision of one or more additional error correction codes so that lower latency reads for small numbers of errors are possible as well additional error correction for larger numbers of errors (see Figures 8 and 9); and/or (4) rebuilding of an entire failed page from redundant data (see Figure 10). Each of these features will be described in more detail in the following.

[0072] Embodiments of the invention advantageously provide relatively low uncorrectable bit error rates for flash memory, low write amplification, long life, fast and efficient retrieval, and efficient storage density such that a solid-state drive (SSD) can be implemented using relatively inexpensive MLC flash for an enterprise storage application. MLC flash can be half or less the price of the SLC flash per bit of storage. Thus, the relatively low price of a sophisticated SSD controller can result in a substantial cost savings to an SSD implemented with MLC versus an SSD implemented with SLC. In addition, while described generally in connection with solid-state drives, the principles and advantages disclosed herein are applicable to other flash drives.

[0073] The techniques disclosed herein can be implemented by software/firmware instructions executed by a processor, by dedicated hardware, or by a combination of software/firmware and hardware. Instructions for the software/firmware can be stored in a tangible, non-transitory computer-readable medium.

Definitions

[0074] The following definitions may be helpful in understanding the specification.

[0075] *x*E*y*: *x* times 10 raised to the *y*-th power, that is $x(10^y)$. Thus 2.1E6 is 2,100,000, and 5E-3 is 0.005.

[0076] RBER: Raw Bit Error Rate. RBER refers to the number of differences (errors) between the data written to a region of flash, and the data read from the flash, divided by the total number of bits in that region.

[0077] UBER: Uncorrected Bit Error Rate. This refers to the number of differences (errors) between a set of data blocks sent to a flash file system, and the contents

of those data blocks when read back from the flash file system, divided by the total number of bits in the data block(s).

[0078] ECC: Error Correction Coding is a class of techniques in which redundant information (parity) is added to information (information bits) in such a way that if errors are subsequently introduced, the original information bits can be recovered. ECC can also stand for error correction code, corresponding to the parity symbols themselves. An ECC has a correction capability, which represents its ability to correct errors. In the simplest form, this may be a certain number of bits *T* per ECC codeword length *N*, but in complex codes the correction capability can be hard to represent succinctly, and is often expressed in statistical terms, such as the RBER (for random errors) that can be reduced to a desired UBER.

[0079] Metadata: Information stored by the file system for the purpose of managing the storage and retrieval of user information from the storage array. This information may include the mapping of user allocation blocks to physical locations, as well as information about the flash array itself. Metadata is a broad term to describe information that is stored in flash, but is not user data, ECC parity, or unused space.

[0080] Codeword: a set of *K* information symbols, plus *P* parity symbols calculated from the information symbols by a predetermined formula. In the case of BCH codes, a symbol is 1 bit; in the case of RS codes it is *m* bits, where *m* is the order of the code. The information bits can correspond to a portion of user data 401 (Figure 4), 501a-501d (Figure 5), 701a-701d (Figure 7) or to metadata, which will be considered user data herein. However, for extra redundancy, a portion of the parity of an earlier-in-time ECC code, such as the Primary ECC 702 can be part of the "block of data" for a subsequent ECC, such as a secondary ECC block code 703. The information block and the parity ECC of a codeword do not need to be stored adjacent to one another.

[0081] Page Stripe: A grouping of one or more equally-sized flash pages (or integer fractions of pages, such as 1/2 a page) which can be read more-or-less simultaneously, and which can be managed as a single region for the purpose of journaling. A page stripe has a unique gear setting at any given time, and has a payload capacity which is an integer number of journaling cells. A page stripe should be made up of pages on different flash planes or, preferably, different dies. For both latency and reliability purposes, it is preferable to use pages residing on different dice and attached to different data buses. In one

embodiment, a page stripe comprises 4320 bytes on each of four pages, the four pages being on separate dice attached to separate ONFI buses.

[0082] Page Grid: A grouping of associated page stripes (or pages when page stripes are not used), wherein tertiary error protection exists to protect the pages of the page grid. The tertiary error correction is preferably an erasure code such as RS or LDPC, capable of rebuilding data lost when a page of the page grid becomes unreadable.

[0083] Block Stripe: A set of flash blocks from one or more dies in a flash array that contains a set of complete page stripes.

[0084] Block Grid: A set of flash blocks from one or more dies in a flash array that contains a set of complete page grids.

[0085] Flash die: A monolithic piece of semiconductor material containing flash memory and control circuitry. A flash package typically contains between one and eight flash dies. Flash dies are often referred to as logical units (LUNs).

[0086] Gear: In the context of this disclosure, a "gear" is a set of parameter(s) specifying an error correction coding scheme, including one or more of the payload capacity, the type and parameters of each component code, the interleaving scheme, and parameters used for determining whether the gear is appropriate for a particular situation. For example, the "Gear 1" sections of Tables III, IV, and V, plus the Threshold G1, collectively define Gear 1.

[0087] Block: Depending on context, a "block" can refer to: the smallest erasable unit within a flash memory, can refer to an amount of data over which an error correction code is calculated, can refer to a block within a flowchart or process, or can refer to an addressable unit of input/output (and corresponding to a particular virtual address or logical block address or LBA) of a data storage device, such as a hard disk, which is emulated by a solid-state drive or a flash drive. It will be apparent to one of ordinary skill in the art which is intended based on context.

[0088] Journaling Cell: The minimum unit of data for which the journaling engine or flash translation layer has a logical-to-physical mapping. In one embodiment, a journaling cell is 536 bytes, comprising 512 user data bytes, 8 bytes of data integrity field (DIF), 8 bytes of data integrity extension (DIX), and 8 bytes of journaling engine metadata.

[0089] High confidence: the probability of the opposite result is less than a predetermined acceptable error probability, such as, 1E-16.

Problems Recognized by Applicant

[0090] A conventional approach to error management for MLC-based SSD can exhibit a number of weaknesses. In one example of a conventional approach, the SSD controller is revised for each brand and technology generation of flash, in order to support the specified ECC. In another example, ECC codes being specified by the vendors tend to be ones that are easy to implement, but which have performance far below the theoretical maximum defined by the Shannon limit.

[0091] In another example, one conventional approach applies the same degree of error correction to all parts of the flash memory. However, flash memory is available in a NOR type and in a NAND type, with the NAND type having more storage density and thus lower costs. An SSD is typically implemented with the NAND type. Experimental results show that some areas of a NAND flash, particularly the upper pages of an MLC flash, and memory cells residing farthest from the sense amplifiers can exhibit many times the error rates of other areas. In another example, a conventional approach applies the same degree of error correction to the flash memory throughout its lifespan. Experimental results show that the error rate of a flash memory typically increases with increased number of program/erase cycles.

Journaling File Systems or Flash Translation Layer

[0092] Flash memory based storage systems can suffer from write amplification. In flash memory, the minimum erasable block size, typically 1 MB or more, is much larger than on hard disks using platters. A file system, which is typically part of an operating system of a host, exchanges data with a data storage device in different kinds of blocks, which should not be confused with storage blocks or erase blocks, which correspond to the minimum erasable unit in a flash device. Data is transferred to and from a host and a block device, such as a hard disk drive or other storage device, in units of data termed allocation blocks herein. These allocation blocks can correspond to "disk sectors" in a hard disk drive. Each of these allocation blocks has a virtual address. A flash translation layer is used so that a flash drive can emulate a hard disk drive by transferring data to and from the host in

allocation blocks as expected by the file system. An example of a flash translation layer is described in U.S. Patent No. 5,404,485 to Amir Ban, the disclosure of which is incorporated by reference in its entirety herein.

[0093] An allocation block is a sequence of words, such as bits or bytes or multiples thereof, of a certain block size, and is the minimum unit of data that is transferred to or from a host and a data storage device, such as a flash drive or a hard disk drive. While the allocation block size can vary among data storage devices, the allocation block size is always fixed for a particular data storage device. Examples of an allocation block size that have been used in the past include 512, 520, 528, 4096, and 4224 bytes. However, other allocation block sizes are possible. The data storage device can communicate its allocation block size and number of available allocation blocks to the host in response to low-level commands from the host.

[0094] The number of address bits can vary in a very wide range. For example, the ATA-1 standard has 28-bit addresses and the ATA-6 standard has 48-bit addresses. Typically, an interface standard requires that the virtual address start at 0 and be contiguous up to the last allocation block available on the drive.

[0095] To modify a piece of data in flash memory, an entire storage block of the flash memory containing the data is read, the entire storage block erased, and then some or all of the data can be modified and written back. If the flash memory is mapped into a conventional file system without a flash translation layer, in which the physical address of data on the flash memory is determined by the apparent address or logical block address in the user file system, this disadvantage can lead to relatively large amounts of reading or writing when files are being altered, which slows down write operations. Other problems also arise. For instance, if a few files are regularly updated, one part of the flash memory may wear out long before others. Even when wear leveling is utilized, such erase and programming operations can drastically shorten the life of a solid-state drive or flash drive.

[0096] Write amplification describes a ratio between user data written, for example, one 512 byte disk sector, and the total amount of data that is actually written to the flash memory. For example, when an entire block, for example, 256 pages of 8 sectors each, is rewritten to alter one sector, then the write amplification is 2048. Block sizes and write amplification can vary from the example shown. However, this very large value for write

amplification means that writes to the SSD will take a relatively long time, and due to the write amplification, wear out the drive quite relatively quickly, which makes physically addressed SSDs impractical for any job that involves modifying small amounts of data on a regular basis.

[0097] To solve these and other problems, a journaling file system or flash translation layer (FTL) can be used, in which data to be written is stored to the flash in essentially the order it arrives, and a mapping table between the "apparent address" or virtual address and the "actual physical address" is maintained by the flash controller. The foregoing is a simplified description, but should serve to help understand the principles and advantages of certain embodiments of the invention.

[0098] Under a journaling file system, write amplification factors approaching of $1/(2u-u^2)$ can be achieved, in which u is the proportion of the drive that is not currently full of data. Maintaining write amplification at a relatively low value throughout the life of a device can help achieve high throughput, good flash life expectancy, and low thermal dissipation.

Error Correction Coding

[0099] One way of improving the performance of an ECC solution is to expand the size of the error correction block (information portion of a codeword) over which the ECC is applied. Two-dimensional "Turbo Product Codes", with iterative decode are a way of creating very large block codes while maintaining a reasonable circuit size and power.

[0100] However, if the information portion of a turbo code block is substantially larger than a typical user data request, the resulting solution will suffer from high latency and wasted throughput. This occurs because in order to decode even a small portion of a codeword, the entire codeword must still be read from flash storage. The latency goes up because the ECC decode operation cannot be completed until the entire block is read. Also, this additional data access can lower throughput for the SSD.

[0101] The illustrated embodiment replaces a conventional 1-dimensional BCH ECC code with a 3-dimensional turbo code formed from block ECC codes. Alternatively, 2 dimensions can be used and other forms of block codes can be used. In one embodiment, the 2 or 3 dimensions (for primary, secondary, and/or tertiary ECC) are each orthogonal to each other such that a codeword of one dimension shares at most one symbol with a codeword of

another dimension. In another embodiment, the primary and secondary codewords are either disjoint or at least partially orthogonal to each other. Table I summarizes primary, secondary, and tertiary error correction characteristics according to one embodiment of the invention.

Type of Error Correction	Data Protection
Primary	within a page or integer fraction of a page
Secondary	within a page stripe
Tertiary	within a page grid

Table I

[0102] Orthogonality refers to the way in which one set of codewords and another set of codewords relate. In particular, given a codeword from a first set and a codeword from a second set, orthogonality refers to the number of symbols at which the codewords intersect. Two codeword sets are exactly orthogonal if they intersect at exactly one symbol. To the extent that 2 codewords intersect at more than one symbol, they are only partially orthogonal. When two codewords do not share any symbols, they are disjoint. Symbol sets are orthogonal if all members from a first set are orthogonal or disjoint with respect to a second set. Note that a symbol for a BCH code is a bit, while for an RS code a symbol is m bits, where m is the order of the code.

[0103] A high degree of orthogonality reduces the chances that a small pattern of errors is uncorrectable by the ECC scheme. In one embodiment, the orthogonality of the primary, secondary, and tertiary dimensions are implemented by selecting data for each primary, secondary, and tertiary codeword based on a predetermined interleave pattern.

[0104] For example, in the illustrated embodiment, the block data portion of the primary codewords can be taken sequentially as is common practice for ECC on flash pages. The block data portion of the secondary codewords can be taken by, for example, taking a bit of data, then skipping an amount of data at least equal to the block data size of a primary codeword to take the next bit of data for the block data portion of the secondary codeword, and so on, until the block data portion of the secondary codeword is filled. The skipping of data can be implemented by utilizing an appropriately sized skip factor in interleaving. The

subsequent secondary codeword can retain the pattern with a one bit shift to capture new data, and so on. Similarly, the block data portion for the tertiary codewords uses another pattern, preferably one with even larger amounts of spacing between selections of bits than for the secondary codewords because for efficiency, it is desirable to access fewer pages of data for the decoding of secondary codewords than it would be to access tertiary codewords.

[0105] As will be explained later, decoding of the foregoing structure can be quite efficient. For the usual case in which the data is relatively uncorrupted, primary ECC decoding alone can confirm good data or correct the correctable errors such that data can usually be retrieved with relatively low latency. However, additional error correction is available to be invoked as needed, giving the best of low latency and more robust error correction.

[0106] One embodiment of invention avoids wasted-throughput and latency problems by choosing the primary ECC such that it can be decoded with only part of the total page stripe available. Most of the time, only the primary codewords containing the user-requested data need to be fetched, and the user data can be released after only a primary decoding operation. Only when the primary decoding operation is unable to perform the correction with high certainty does a higher latency decode operation need to be invoked. The primary ECC and its decode algorithm should be carefully chosen so that, even in the presence of relatively high bit error rate, the primary ECC can correct the errors present with a relatively high probability, for example, in one embodiment, at least 99% of the time, and almost never performs a "false decode." Other values are applicable, such as, but not limited to, at least 95% of the time or at least 90% of the time. Typically, it will be desirable to keep an expected amount of latency relatively low. The average amount of latency can be estimated by summing (a) the length of time that it takes to read the data and perform primary decoding; (b) the probability that the data is not released after primary decoding times the length of time that it takes to read a full page stripe and perform the primary/secondary iterative decoding operation; and (c) the probability that the data is not released after primary or primary/secondary decoding times the length of time that it takes to read a full page grid and perform tertiary/secondary/primary iterative decoding operation.

[0107] A false decode occurs when the primary ECC decoding process deduces that it has properly corrected the data, when in fact it has added errors. If a false decode

occurs without detection, incorrect data would be released, which is an occurrence that should be kept to a very low level. For example, the false decode rate should be less than 1E-16.

[0108] The desired rate of false decodes can be guaranteed by choosing a primary quick-release threshold Qp ≤ Tp, such that Eq. 1 is satisfied.

$$\left(\frac{2Tp}{Np}\right)\cdot\left(\frac{1}{Tp!}\right)\cdot\prod_{i=1}^{Tp-Qp-1}\left(\frac{Tp-i+1}{Np}\right) < aUBER \qquad \text{Eq. 1}$$

[0109] In Eq. 1, the "!" indicates factorial, *aUBER* corresponds to a predetermined acceptable uncorrected bit error rate, *Tp* corresponds to the correction capability of the primary ECC, and *Np* corresponds to the size of a primary codeword. The expression of Eq. 1 is solved for the value of *Qp*. The calculations described in Eq. 1 can be performed during a design stage, during production test or the like, and the value obtained for the parameter *Qp* can be stored or programmed into an SSD controller 1300 (Figure 13). The quick-release parameter *Qp* can vary based on gear selection.

[0110] If ECC correction is performed, *Qp* or fewer bits are corrected, the data may be immediately released. If more than *Qp* bits are corrected, or the codeword is uncorrectable, then the result of the error correction is not certain. In this case, a page stripe should be fetched, and an iterative decoding operation using secondary codewords performed. The decoding operation of a page stripe can takes a great deal longer time than the primary decoding operation, but so long as the likelihood of requiring a page stripe decode can be kept sufficiently low (e.g. 1% or less), the impact on the average latency and throughput is advantageously relatively small.

[0111] The desired low likelihood of full stripe decodes can be achieved by choosing a maximum operating raw bit error rate *G* such that the safe correction capability (*Qp*) is sufficient to correct the targeted Bit Error Rate at least 99% of the time, as calculated by Eq. 2.

$$P_{corrected} \approx \sum_{i=0}^{Qp}\binom{N}{i}G^i \geq 0.99 \qquad \text{Eq. 2}$$

[0112] Figure 2 illustrates a flash programming process, which includes error correction encoding, according to an embodiment of the invention. The process can be implemented by software or firmware executed by a processor, by hardware, or by a combination of both software or firmware and hardware. For example, the process can be performed by a flash drive controller, which will be described later in connection with Figure 13. It will be appreciated by the skilled practitioner that the illustrated process can be modified in a variety of ways. For example, in one embodiment, various stages can be implemented in a pipelined fashion and/or performed in parallel. For example, in another embodiment, various portions of the illustrated process can be combined, can be rearranged in an alternate sequence, can be removed, or the like. At the start of the process, it is assumed that, when variable, a particular strength of ECC has been selected for the various ECC encoders.

[0113] The process begins in a stage 202 in which the process receives a write command and the data (information) to be written. The write command and the data can be received from a journaling engine. As will be explained later, data can also correspond to error correction code from earlier-in-time encoding. The process advances from the stage 202 to a stage 204. The amount of data for each page stripe can be pre-arranged based on the type and strength of the ECC currently selected for the page stripe.

[0114] In the stage 204, the process distributes the data among the *n* lanes of the SSD to which the data will ultimately be written. The lanes can be parallel data buses to which the flash devices of the SSD are connected. In one embodiment, each lane is an 8 bit wide open NAND flash interface (ONFI) bus or a TOGGLE bus. The distribution of data can be done a variety of ways. For example, bit interleaving can be used to distribute the data. In another example, a first received byte can be allocated to a first lane, a second received byte to a second lane, a third received byte to a third lane, a fourth received byte to a fourth lane, a fifth received byte back to the first lane, and so on, to distribute the data among the lanes in a manner similar to a card dealer dealing cards to players. The distribution of the data can be by, for example, splitting a 32-bit data bus into four separate 8-bit buses, one to each of four engines implementing the stages 206. In another embodiment, sequentially received data can be used to fill one entire page (or integer fraction) of a page stripe, and then continue on to fill the next page (or integer fraction) of the page stripe, and so forth. The

process advances from the stage 204 to the stages 206. Stage 206 and stage 208 are performed for each lane.

[0115] In the stages 206, for a particular lane, the process can be considered to break up the data for the lane into block data sized portions for codewords of that lane. For example, start of block and end-of-information-block markers can be added, and data flow paused until the parity insertion in the stage 208 is complete. The process advances from the stages 206 to stages 208.

[0116] In the stages 208, the process calculates the ECC parity (the Primary Parity) corresponding to the information portion of the codeword, and inserts said parity into the stream. The resulting data stream is illustrated in Figure 4. A skilled practitioner will recognize that the illustrated process can be modified in a number of ways, including the use of shortened codes, non-systematic codes, and the like. The process can perform the stages 206, 208 in a pipelined fashion, or in a large batch.

[0117] Outputs of the stages 208 are provided to the stage 210 for secondary processing, to the stage 216 for tertiary processing, and to the stage 222 for writing to the flash. It should be noted that data can be written sequentially to the data or cache register of a flash device, so long as the previously written data is not being changed. Alternately, a complete page stripe of data can be accumulated in a data buffer in the controller prior to any write commands being issued. In addition, when writing relatively large amounts of data, it can be desirable to perform write operations in parallel. For example, a flash memory die can have multiple planes that can be written at the same time. Thus, there can be multiple areas of memory set aside for buffers for page stripes and for tertiary parity calculation.

[0118] In the stage 210, data from the stages 208 is observed. Each bit is assigned to one of a set of secondary codewords, in accordance with the principles of turbo coding. In a preferred implementation, in which there are *Cs* secondary codewords for a particular gear, data from each primary codeword is assigned on a round-robin, bitwise basis to the secondary codewords, such that each secondary codeword has a substantially equal number of bits (plus or minus 1) from each primary codeword. The rearranged bits are passed to the stage 212. After the user data and primary parity on user data has been processed, the stage 210 is complete for the page stripe. Secondary parity, tertiary parity, PoSP and PoTP, as will be described later, can be used in the stage 210.

[0119] In the stage 212, the assembled bits are treated as the information bits of a secondary ECC code. In one embodiment, the secondary ECC code is a BCH code. In the stage 212, the ECC parity bits corresponding to the information bits of the assembled secondary codewords are calculated. In one embodiment, the operation of the stage 212 is as follows. A scratchpad memory of sufficient size for the parity of all the secondary codewords of a stripe is provided. This memory is set to zero when a page stripe encode is initiated upon entry to the stage 202. Whenever new information bits are assigned to a secondary codeword in the stage 210, then the current value of the scratchpad for that codeword is read and loaded as the state of a linear feedback shift register (LFSR). The new data is shifted into the LFSR, and the revised state written back to scratchpad memory. These LFSR operations can be performed in a single cycle or multiple cycles. After all the secondary codewords for the page stripe have been fully calculated, the process proceeds to the stage 214.

[0120] In the stage 214, the secondary parity is read from the scratchpad memory, and formed into a data stream. The secondary parity is passed back to the stage 204, where the secondary parity is treated as if it were user data. In the stage 206, the secondary parity is broken up into sections forming the information bits of primary codewords (which can be of different lengths than the ones used for user data) and in the stage 208, primary parity (termed "primary on secondary parity or PoSP") is calculated over these secondary parity bits. The amount of primary parity calculated over the secondary parity can be different than for the user data. The output of the stage 208 is provided to the stage 216 for tertiary processing, and to the stage 222 for writing to the flash memory.

[0121] For the stage 210, in one embodiment, each primary codeword should be covered roughly equally by the applicable secondary codewords, and likewise the information bits of each secondary codeword should contain as even a distribution of bits from primary codewords as practical. This can be achieved by, for example, ensuring that the number of secondary codewords *Cs* in a page stripe is prime relative to the size (total length) of each primary codeword *Np*, or by rotating the order of bit assignments to secondary codewords by one position every *Cs* bits, every *Cs* times *n* bits, wherein *n* is one or more. In one embodiment, the rotation is performed by one position every *Cs* bytes, that is, with *n* equal to 8.

[0122] The states from the stage 216 through the stage 220 apply to tertiary processing, which operates across a span of two or more page stripes, such as four page stripes. In the stage 210, the stream of primary codewords from the stage 208, including bits originating from both the stage 202 and the stage 214, are observed. Each bit is assigned to a tertiary codeword. In one embodiment, in which each grid contains *Wt* tertiary codewords based on a Reed-Solomon error correction code defined over Galois field gf(2^Mt) for a particular gear, data from each primary codeword is grouped into symbols of size 2^Mt, and assigned on a symbol-wise round-robin basis to the tertiary codewords, such that each tertiary codeword has a substantially equal number of bytes (plus or minus 1) from each primary codeword, and a roughly equal number of bytes covered by each secondary codeword. The rearranged bits are passed to the stage 218. Padding of implied zeroes may be used to reach an integer number of symbols, according to the known techniques of shortening RS codewords by partial symbols.

[0123] In the stage 218, the assembled bits are treated as the information bits of a tertiary ECC code. In a preferred implementation, the tertiary ECC code is an RS code. In the stage 218, the ECC parity bits corresponding to the information bits of the assembled secondary codewords are calculated. In one embodiment, the operation of the stage 212 is as follows. A scratchpad memory of sufficient size for the parity of the tertiary codewords of the page stripes of one or more grids is provided. The scratchpad memory for each page grid 600 (Figure 6) is set to zero when the first page stripe encode of the grid is started (stage 202 for that page stripe). Whenever new information bits are assigned to a tertiary codeword in the stage 216, the current value of the scratchpad for that codeword is read and loaded as the state of a Reed-Solomon encoding LFSR. The new data is shifted into the LFSR, and the revised state written back to scratchpad memory. These LFSR operations can be done in a single cycle or in multiple cycles. After the tertiary codewords for the page grid have been fully calculated, the process proceeds to the stage 220.

[0124] In the stage 220, the tertiary parity is read from the scratchpad memory, and formed into a data stream. The tertiary parity is passed back to the stage 204, in which the tertiary parity is treated as if it were user data. In the stage 206, the tertiary parity is broken up into sections forming the information bits of the primary codewords (which may be of different lengths than the ones used for user data) and in the stage 208, ECC parity

(termed "primary on tertiary parity" or PoTP) is calculated over these tertiary parity bits. The amount of primary parity calculated over the tertiary parity may be different than for the user data. The output of the stage 208 is provided to the stage 222 for writing to the flash memory, but not to the stage 210 or the stage 216.

[0125] In the operation of the stage 210, each flash page should be covered roughly equally by the corresponding tertiary codewords, and each primary codeword should be covered roughly equally by the tertiary codewords, and likewise the information bits of each tertiary codeword should reflect as even a distribution of bits from the primary codewords as practical. This can be achieved by, for example, ensuring that the number *Wt* of tertiary codewords in a page grid is prime relative to the size (total length) *Np* of a primary codeword, or by rotating the order of bit assignments to tertiary codewords by one position every *Wt* symbols, each tertiary codeword symbol having a size of *Mt* bits.

[0126] In one example in which the tertiary symbol size *Mt* is 8 bits or one byte, tertiary codeword size *Nt* is 250 bytes, block data size *Kt* is 226 Bytes, and each tertiary codeword can correct up to 12 symbol errors, up to 24 erasures, or some mix thereof such that the number of erasures plus twice the number of symbol errors is equal to 24. In one embodiment, the strength of the tertiary ECC is the same for all gears.

[0127] Figure 3 illustrates an overview of a decoding process for decoding of information encoded by the process described earlier in connection with Figure 2. The decoding process can be implemented by software or firmware executed by a processor, by hardware, or by a combination of both software or firmware and hardware. For example, the decoding process can also be performed by a solid-state drive controller, which will be described later in connection with Figure 13.

[0128] Figure 3 provides a high level overview of one embodiment of a process of retrieving and recovering data from the flash memory. It will be appreciated by the skilled practitioner that the illustrated process can be modified in a variety of ways. For example, in another embodiment, various portions of the illustrated process can be combined, can be rearranged in an alternate sequence, can be removed, or the like. In the stage 302, the process receives a request to read certain data from the flash memory. In the stage 304, an attempt is made to recover the data with minimal latency. The stage 304 will be described in greater detail later in connection with a partial page stripe decoding portion 304 of the

process of Figure 8, with stages 802–804. If that fails, a second attempt to recover data is made with a stage 308, using additional data stored in the flash memory, and additional error correction processing. The stage 308 will be described later in greater detail later in connection with a page stripe decoding portion 308 of the process of Figure 8. If the stage 308 fails, then a third attempt is made with a stage 310, using still further add additional data previously stored in the flash memory, and further additional processing. The stage 310 will be described greater detail later in connection with the page grid decoding portion 310 of the process of Figure 8. If the stage 310 fails, it can be because one particular page, block, or die in the flash array has failed entirely. Thus, a fourth attempt is made with a stage 312, using special recovery techniques that will be described in greater detail later in connection with Figure 10.

[0129] If the recovery process of the stage 312 fails, then the user data is uncorrectable, and a report to that effect is made in the stage 314 and in a stage 1034 (Figure 10). If any of the stages 304, 308, 310, 312 is successful, the requested data is returned to the journaling engine in a stage 306, along with various status information. The stage 306 is represented in more detail by the stages 805, 830, 832 (Figure 8) and the stage 1034 (Figure 10).

[0130] Figure 4 illustrates an arrangement for user data stored in a flash page having the same size as that of Figures 1A and 1B along with primary ECC data. An arrangement of secondary ECC and tertiary ECC will be described later in connection with Figures 5 and 7. Rather than storing the user data in relatively large portions such as shown in the user space 101 (Figure 1A) and the user data 103 (Figure 1B) illustrating 4096 byte portions and 1024 byte portions, respectively, the illustrated embodiment preferably stores the user data in much smaller portions. In one embodiment, the user data 401 (information) to be stored to a flash page 400 is divided into portions of a length *Kp* convenient to or efficient for the primary BCH. In an alternative embodiment, the user data 401 is divided into a portion having a length corresponding to a multiple of *Kp* payload symbols. After each *Kp* payload symbols, a number of BCH parity bits *Pp* are added 402, for a total of *Np* parity symbols. The values of *Kp* and *Pp* are different for each ECC gear, and can be selected for the desired correction capability, balanced with ease of implementation. ECC gearing will be discussed in greater detail later in connection with Figure 11. Example

values for *Kp* and *Pp* are illustrated in Table III. The block data for the primary data includes user data and sometimes metadata. The block data size *Kp* is not required to have a simple arithmetic relationship to the size of the block used by the file system.

[0131] After the information bits for that page and their primary parity are added, a region 403 is reserved for the storage of secondary parity, which in the illustrated embodiment, is calculated over a page stripe, which will be described later in connection with Figures 5 and 7. A region 404 is also reserved for primary parity on secondary parity (PoSP), that is, ECC parity information for the data to be stored in the region 403.

[0132] Preferably, space is also reserved for a fingerprint 407 or check code, which can be derived or determined from the physical address of the flash page. The fingerprint 407 can be used to verify whether or not an address fault has occurred, that is, determine whether or not the retrieved data came from the correctly addressed memory space. In one embodiment, the fingerprint 407 is derivable from a physical location, such as address, of the page 400. Anything can be used for the fingerprint 407 so long as it identifies the page 400 uniquely, at least for a particular lane. Two pages on different lanes can use the same fingerprint 407 because they are physically separated and there is no risk of retrieving data across physically distinct lanes. Many different techniques can be used. In one embodiment, a hash code generated from the physical address or at least a portion of the physical address is used. The hash code preferably has a relatively large hamming distance such that one or more errors can be tolerated. The hash code can be based on a BCH calculation, on CRCs or the like. Alternatively, the address itself and corresponding ECCs can be used, but there is no need to store the actual address itself and it would thus be a relatively inefficient use of memory space. The use of the address by itself without corresponding ECC could be used, but an address line failure can result in a change of just one address bit, which in the data could be subject to a random bit error and then be erroneously reported as correct. All that is needed is a unique identifier such that the page can be identified in some repeatable manner such that the SSD controller can compare the fingerprint 407 retrieved from flash memory to an expected value for the fingerprint.

[0133] Figure 5 illustrates an example of an arrangement for a first type of flash page stripe 550 according to an embodiment of the invention. An example of an arrangement for a second type of flash page stripe 640 used for the last page stripe will be described in

greater detail later in connection with Figure 7. The stripe width *Ws* of the flash page stripe 550 can be within a broad range of numbers of one or more, but for efficiency should match the number of lanes. In the illustrated embodiment, four flash pages 400a-400d each in its own lane are used such that the stripe width *Ws* is 4 with lanes 0 to 3. Other values for the stripe width *Ws* will be readily determined by one of ordinary skill in the art.

[0134] In the illustrated embodiment, the flash pages 400a-400d of the page stripe 550 share the same address, but belong to different flash lanes, such as to different bits of a data bus. Each of the flash pages 400a-400d can be as described earlier for the flash page 400 of Figure 4. In addition, note the following. In the illustrated embodiment, user data 501a-501d (information) can be distributed among the four flash pages 400a-400d on a byte interleaved round-robin basis as described earlier in connection with the distribution of data among 4 lanes in the stage 204 of Figure 2. In one embodiment, the user data 501a-501d is scrambled before being stored and unscrambled after it has been decoded as will be described in greater detail later in connection with a decision block 804 of Figure 8. Of course, as more and more data is stored, it will eventually become necessary to write data to different pages of the lanes. Primary parity 502a-502d is added on each flash page 400a-400d as described earlier in connection with the stage 208 of Figure 2.

[0135] As each bit (whether information bit or parity bit) is written to a user data portion 501a-501d or to a primary parity portion 502a-502d of the page stripe 550, the bit is also used in the calculation of one of the *Cs* secondary ECC codewords, preferably on an individual bit interleaved basis for robustness. The secondary ECC codewords can be maintained in scratchpad memory, such as a register or RAM, while the secondary ECC codewords are being computed. The secondary parity calculation for a particular codeword of the *Cs* secondary codewords is updated as new bits are covered by the particular codeword. After the incoming user data (information) and primary parity bits have been processed by the secondary ECC encoding process, the secondary parity calculation can be finalized, and the secondary parity can be written to the secondary parity portion 503a-503d of the flash page stripe 550. In one embodiment, the secondary parity is stored on the flash page stripe 550 on a bit-interleaved manner across the entire flash page stripe 550.

[0136] Primary parity is calculated over the secondary parity bits written to each flash page, that is, over the data stored in the portions 503a-503d, and stored in the PoSP

portions 504a-504d. A fingerprint 507a-507d for each flash page 400a-400d is preferably added to determine whether or not the retrieved data came from a read of the correct memory address as described earlier in connection with the fingerprint 407 for Figure 2 for further robustness. The same fingerprint 507a-507d can be reused across the pages 400a-400d of a particular page stripe 550 provided that the pages 400a-400d are each in separate lanes because each lane is physically a different set of conductive lines, and thus, unique identification of data is preserved.

[0137] Figure 6 illustrates a page grid 600. The address space of the flash memory is divided into many page grids 600. Preferably, a page grid 600 includes an associated set of pages, each from a different flash memory die. However, it will be understood that the principles and advantages described herein will be applicable to systems in which a page grid uses more than one page each or less than one page each from each flash memory die. For example, it can be advantageous to build grids from multiple partial pages, each from a separate die, or to build grids from multiple pages on separate planes of the same die, or both. In addition, while the illustrated page grid 600 has a "square" arrangement with 4 pages across (from 4 lanes) and 4 pages down (from 4 page stripes), the page grid 600 does not have to be square, nor is it required that the page stripes making up the page grid 600 be encoded with the same gear.

[0138] The pages of the page grid 600 are already associated in a page stripe. In the illustrated embodiment, a page grid 600 is made up of four page stripes 550a, 550b, 550c, 640 having a rank of 4 each, thus the page grid 600 has 16 pages 612, 614, 616, 618, 622, 624, 626, 628, 632, 634, 636, 638, 642, 644, 646, 648 from 16 different die. Of these page stripes 550a, 550b, 550c, 640, the first three page stripes 550a, 550b, 550c are of the first type of flash page stripe 550 described earlier in connection with Figure 5, and the last page stripe 640 is of the second type of flash page stripe 640 that will be described later in connection with Figure 7. The number of lanes and the amount of rank can vary in a very broad range and do not have to be the same. For example, 4 lanes and a rank of 8 will also work. A relatively larger number of lanes will typically have more throughput, but can be more complicated to implement with wider data buses and so forth. Thus, the selection of the number of lanes to have and the deepness of rank is subject to an engineering tradeoff.

[0139] In the illustrated embodiment, data fills the page grid 600 as follows. Initially, data to be written is written to the first page stripe 550a. When the first page stripe 550a is filled with data, then data is written to the second page stripe 550b. When the second page stripe 550b is filled, then data is written to the third page stripe 550c. When the third page stripe 550c is filled, then data is written to the last page stripe 640 of the page grid 600. When the last page stripe 640 is filled, then subsequent data is stored to a different page grid. In addition, within each page stripe, data is distributed among the pages of the page stripe in a round-robin manner. For example, with respect to the first page stripe 550a, a symbol of data is stored in the first page 612, then the next symbol of data is stored in the second page 614, then the next symbol of data is stored in the third page 616, then the next symbol of data is stored in the fourth page, then the next symbol of data is stored in the first page 612, and so on until the first page stripe 550a is filled. In the illustrated embodiment, the symbol size used for storage in the pages corresponds to a byte or 8 bits, but other sized symbols can be used.

[0140] The data within each page 612, 614, 616, 618, 622, 624, 626, 628, 632, 634, 636, 638, 642, 644, 646, 648 in the page grid 600 is covered by its own set of primary parity data 402 (Figure 4), 502a-502c (Figure 5). The data within a page stripe 550a, 550b, 550c, 640 is also protected by a secondary parity 503a-503d (Figure 5), which in turn is protected within the individual pages by the PoSP 504a-504d. The pages 642, 644, 646, 648 of the last page stripe 640 of the page grid 600 also carry the tertiary parity 705a-705d and the primary over tertiary parity (PoTP) 706a-706d bits as will be described in greater detail later in connection with Figure 7. The tertiary parity bits 705a-705d protect the page grid 600 as a whole and are stored in the last page stripe 640.

[0141] In one embodiment of the invention, the allocation of bits from allocation blocks to codewords is as follows. For the purposes of this calculation, the bits of a page stripe are numbered starting from 0, such that the first bit of the first byte of lane 0 is bit 0; the last bit of the first byte of lane 3 is bit 31; the first bit of the second byte of lane 0 is bit *Ws**8, and so on. The endianness of the bits, for example, byte order, is not important, so long as consistency is maintained. For the purposes of these equations, "div" represents integer division rounding down, and "mod" represents the modulo operator. For example, 11

div 4 equals 2, and 11 mod 4 equals 3. In the illustrated embodiment, the page stripe width *Ws* is 4.

[0142] In the illustrated embodiment, user data from an allocation block is allocated on a bytewise basis to one of *Ws* primary codewords at a time, that is, one primary codeword per lane). The size or length of each primary codeword is represented by *Np*. After each primary codeword has (*Np* div 8) bytes, the next set of *Ws* codewords are used. Mathematically, the primary codeword P to which bit *x* is allocated can be as expressed in Eq. 3.

$$P(x) = 4(x \operatorname{div} \{Np \cdot Ws\}) + (x \operatorname{div} 8) \bmod Ws \qquad \text{Eq. 3}$$

[0143] Data is also allocated on a bytewise basis to one of *Cs* secondary codewords, where *Cs* corresponds to the number of secondary codewords for the page stripe. Every *Cs* bytes, a 1-lane rotation is applied to the secondary codewords, to avoid the same secondary codeword falling in the same lane. Mathematically, the secondary codeword to which bit *x* is allocated can be as expressed in Eq. 4.

$$S(x) = s'(x) - [s'(x) \bmod Ws] + [\{s'(x) - i(x)\} \bmod Ws] \qquad \text{Eq. 4}$$

[0144] The variables *s*'(*x*) and *i*(*x*) can be solved as expressed in Eq. 5 and Eq. 6, respectively.

$$s'(x) = (x \text{ div } 8) \bmod Cs \qquad \text{Eq. 5}$$

$$i(x) = (x \text{ div } 8) \text{ div } Cs \qquad \text{Eq. 6}$$

[0145] Data is also allocated on a bytewise basis to one of *Ct* tertiary codewords. Every *Ct* bytes, a 1-lane rotation is applied to the tertiary codewords, to avoid the same tertiary codeword falling in the same lane. Mathematically, the tertiary codeword to which bit *x* is allocated can be as expressed in Eq. 7.

$$T(x) = [t'(x) \bmod Ws] + [\{t'(x) - i(x)\} \bmod Ws] \qquad \text{Eq. 7}$$

[0146] The variables *t*'(*x*) and *i*(*x*) can be solved as expressed in Eq. 8 and Eq. 9, respectively.

$$t'(x) = (x \text{ div } 8) \bmod Ct \qquad \text{Eq. 8}$$

$$i(x) = (x \text{ div } 8) \text{ div } Ct \qquad \text{Eq. 9}$$

[0147] Primary ECC on the secondary parity (PoSP) and primary ECC on the tertiary parity (PoTP) can follow the same rules as the primary, but the values of the codeword length *N*, the number of information bits in a codeword *K*, the number of codewords *C* per page stripe, and the correction capability of the ECC or FEC *T* can be adjusted to make the bits fit into the available space.

[0148] The bytewise allocation of data to the secondary means that a given primary codewords and secondary codeword tend to share an integer number of bytes. This is convenient from an implementation point of view, but it can mean that the maximum overlap will tend to be greater, for example 16 bits, than the minimum possible, for example 14 bits. This condition can lead to "killer pattern" situations in which a small number of errors (for example, 11 errors within a particular 16 bit overlap, in which $Ts < Tp < 11$). However, these patterns inherently require multiple bits errors within a single byte. A Reed-Solomon based tertiary is very efficient at fixing such "killer patterns."

[0149] Various alternatives exist. For example, in an alternative embodiment of the invention, the allocation of bits to codewords can be as follows. Data is allocated on a bytewise basis to one of *Ws* Primary codewords at a time, that is, one primary codeword per lane). Once each codeword has (*Np* div 8) bytes, the next set of *Ws* codewords are used. Mathematically, the primary codeword to which bit x is allocated can be as expressed in Eq. 10.

$$P(x) = 4\left[x \operatorname{div}(Np \cdot Ws)\right] + (x \operatorname{div} 8) \bmod Ws \qquad \text{Eq. 10}$$

[0150] Data is allocated on a bitwise basis to the *Cs* secondary codewords. *Cs* is a prime number, or at least relatively prime to both *Cp* and *Ws*.

$$S(x) = x \bmod Cs \qquad \text{Eq. 11}$$

[0151] Data is allocated on a bytewise basis to one of *Ct* tertiary codewords. *Ct* is a prime number, or at least relatively prime to *Cp*, *Cs*, and *Ws*. Mathematically, the tertiary codeword to which bit *x* is allocated can be as expressed in Eq. 12.

$$T(x) = (x \text{ div } 8) \bmod Ct \qquad \text{Eq. 12}$$

[0152] In yet another embodiment of the invention, the allocation of bits to codewords can be as follows. Data is allocated to each primary codeword until it is full. Thus, in this embodiment, every primary codeword is across all *Ws* lanes. Data is allocated to the *Cs* secondary codewords on a round-robin basis.

[0153] Tertiary codewords need not be present. If tertiary codewords are present, data can be allocated on a bytewise basis to one of *Ct* tertiary codewords. *Ct* should be a prime number, or at least prime relative to *Cp*, *Cs*, and *Ws*. Mathematically, the tertiary codeword to which bit *x* is allocated can be as expressed in Eq. 13.

$$T(x) = (x \text{ div } 8) \bmod Ct \qquad \text{Eq. 13}$$

[0154] Figure 7 illustrates a structure for the last page stripe 640 (Figure 6) of the page grid 600. The illustrated last page stripe 640 of the page grid 600 includes four flash pages 700a-700d. However, the stripe width *Ws* can vary as described earlier in connection with the flash pages 400a-400d of Figure 5.

[0155] The structure of the last page stripe 640 can be similar to the page stripe 550 described earlier in connection with Figure 5, but has additional tertiary ECC and less user data space. The last page stripe 640 holds user data 701a-701d, primary parity on user data 702a-702d, secondary parity 703a-703d, primary parity on secondary parity (PoSP) 704a-704d, and fingerprints 707a-707d, which are analogous to the user data 401, 501a-501d, primary parity on user data 402, 502a-502d, secondary parity 503a-503d, primary parity on secondary parity (PoSP) 504a-504d, fingerprints 407, 507a-507d respectively, described earlier in connection with Figures 4 and 5.

[0156] The total user data space is smaller, holding less user information than other page stripes 550 (Figure 5) 550a, 550b, 550c (Figure 6) because the last page stripe 640 additionally holds tertiary error correction information for the corresponding page grid. The ECC encoding parameters, such as the number of primary codewords *Wp*, number of information bits *Kp* per primary codeword, number of bits *Np* per primary codeword, number of information bits *Ks* per secondary parity, number of bits *Ns* per secondary codeword, number of secondary codewords *Cs* per page stripe can vary somewhat from the ECC encoding parameters used for the other page stripes 550a, 550b, 550c to allow for efficient packing. An example of sample ECC parameters will be described later in tables following a discussion of Figure 7. The ECC encoding parameters of this last page stripe 640 are

preferably selected so that the error correction capability is at least as good as for the other page stripes.

[0157] The last page stripe 640 holds the parity bits of the tertiary codewords, that is, the "tertiary parity" 705a-705d, which are calculated bytewise over the user data 501a-501d, 701a-701d, the primary parity 502a-502d, 702a-702d, the secondary parity 503a-503d, 703a-703d, and the PoSP 504a-504d, 704a-704d portions of all the page stripes 550a, 550b, 550c, 640. Thus, every bit of the user data 501a-501d, 701a-701d is covered by at least one primary codeword, one secondary codeword, and one tertiary codeword. The allocation can be performed such that each tertiary codeword has an approximately equal number of bytes from each page of the page grid 600.

[0158] The correction capability *Tt* of the tertiary parity 705a-705d should be chosen to be high enough to be able to find the values of all the bytes in any one page (if that page is marked as an erasure, that is, marked as unknowns) of the page grid 600, which then permits the rebuilding of a lost page. For example, a page of memory can be lost due to the failure of a row in a flash memory. The correction capability *Tt* of the tertiary parity 705a-705d should additionally have three-quarters of the correction capability of the secondary parity 503a-503d, 703a-703d on one page stripe 550a, 550b, 550c, 640 for the case of 4 pages or portions of pages per page stripe. This latter requirement is based on the assumption that, if one page of the page grid 600 fails entirely, the secondary parity 503a-503d, 703a-703d for the corresponding page stripe 550a, 550b, 550c, 640 will also be inoperative without correction provided by the tertiary parity 705a-705d. The tertiary parity 705a-705d is itself covered by an appropriately resized primary ECC parity over tertiary parity (PoTP) 706a-706d. Further details of the correction capability *Tt* of the tertiary parity 705a-705d can be as follows. Generally, the extra correction capability can be *Ws*-1 / *WsWg* of the secondary parity, wherein *Ws* is the number of pages in a page stripe, and *WsWg* is the number of pages in a page grid. In the illustrated example, *Ws* is 4 and *WsWg* is 16, such that correction capability *Tt* of the tertiary parity should further include 3/16 of the correction capability of the secondary error parity such that, in the illustrated example, the correction capability *Tt* for the tertiary parity should be able to correct for a missing page and 3/16 of the correction capability of the secondary parity. It will be understood that the 3/16 figure

will vary depending on the number of pages or portions of pages per page stripe and the number of page stripes per page grid.

[0159] In one embodiment, the tertiary ECC has a correction capability equal to (a) half of the maximum number of symbols that each tertiary codeword has on any one page of the page grid, plus( ) at least $TTs * (Ws - 1) / (Wt*Wg*Ws)$ symbols, wherein *TTs* corresponds to the total correction capability of ECC codewords calculated across multiple pages of a page stripe, *Ws* corresponds to the number of pages in a page stripe, *Wt* corresponds to the number of tertiary codewords, and *Wg* corresponds to the number of page stripes in a page grid.

[0160] The tertiary code is preferably an erasure code, that is, a code that is tolerant of missing symbols. Such a code has improved decoding capability when certain portions of the codewords are known to be highly errored. The property of being an erasure code can permit the recovery of data even if an entire page of data were to become unreadable as can happen when a die goes bad or when there is a problem with a row in a memory chip. In one embodiment, the tertiary code is a Reed-Solomon (RS) code. In alternative embodiments, the tertiary code can be a Low density parity check (LDPC) code, Tornado code, Online code, Luby transform (LT) code, Raptor code, or the like. Other applicable erasure codes will be readily determined by one of ordinary skill in the art. In one embodiment, the strength of the tertiary code does not vary with the "gearing" of ECC, which will be discussed in greater detail later in connection with Figure 11. The tertiary code can deal with a total failure, which can happen at any time, whereas the primary and secondary codes are intended to cover wear out, which happens gradually.

[0161] It is not necessary to wait for a complete page grid to be available before writing data to flash. Interim state values for the tertiary parity calculation can be held in scratchpad memory after each page stripe, and the calculation resumed on arrival of the next page stripe. In one embodiment of the invention, a plurality of page grids are programmed in an interleaved fashion. Each page stripe is identified by a thread ID, associated with a region of scratchpad RAM and a set of interim tertiary values. As each page stripe arrives, an encoder retrieves any interim tertiary values, initializes the tertiary encoders, processes the page stripe, and then either stores the new interim values back to scratchpad, or writes out the completed tertiary parity, as appropriate.

[0162] The illustrated embodiment has 4 pages per page stripe, and 4 page stripes per page grid 600. An example of physical capacities of each is described in connection with Table II, with corresponding ECC parameters in Tables III to VII.

	Page	Stripe	Grid
No. of Pages	1	4	16
Capacity in Bytes	4320	17280	69120

Table II

[0163] Table III illustrates an example of data allocation for a first gear, second gear, and third gear amount of ECC strength for the primary ECC with reference to page stripe 0 550a, page stripe 1 550b, page stripe 2 550c, and page stripe 3 640 (Figure 6). As illustrated in the table, increasing ECC strength corresponds to decreasing ratio *R*. As illustrated in Table III, in the illustrated embodiment, the primary payload of a page stripe is not an integer multiple of 512, 530, 1024, or 2048 bytes, which can be examples of allocation block sizes. However, the primary payload can vary in a very broad range and in alternative embodiments a particular page stripe and ECC strength can combine to generate a payload having a multiple of the allocation block size.

	First Gear		Second Gear		Third Gear	
Page Stripe	0, 1, or 2	3	0, 1, or 2	3	0, 1, or 2	3
Primary Payload (536-byte 'Journaling Cells')	30	16	27	14	21	11
Primary Payload (Information Bytes)	16080	8576	14472	7504	11256	5896
Primary codeword information (bits) (*Kp*)	1792	1560	1816	1504	808	848
Primary parity bits per codeword (*Pp*)	110	110	198	198	264	264
Rp = Kp/Np	0.942	0.934	0.902	0.884	0.754	0.763
Correction capability (*Tp*)	10	10	18	18	24	24
Primary codewords per page	18	11	16	10	28	14
Primary codewords per page stripe (*Wp*)	72	44	64	40	112	56
Fractional Shortening	FStFP by Page	FStFP by Page	FStFP by Page	FStFP by Page	FStFP by Page	FStFP by Page
Total primary parity bits per page stripe	7920	4840	12672	7920	29568	14784

Table III

[0164] Table IV illustrates an example of data allocation for the secondary ECC.

	First Gear		Second Gear		Third Gear	
Page Stripe	0, 1, or 2	3	0, 1, or 2	3	0, 1, or 2	3
Secondary Payload (Primary Codeword Bytes)	17072	9184	16056	8496	14952	7744
Secondary codeword length (bits) (*Ns*)	1018	498	956	692	990	734
Secondary codeword information (bits) (*Ks*)	1008	488	896	632	880	624
Secondary parity bits per codeword (*Ps*)	10	10	60	60	110	110
Rs = *Ks*/*Ns*	0.990	0.980	0.937	0.913	0.889	0.850
Secondary Correction capability (*Ts*)	1	1	6	6	11	11
Secondary codewords per page stripe (*Ws*)	136	152	144	108	136	100
Fractional Shortening	FStFP by PgStr	FStFP by PgStr	FStFP by PgStr	FStFP by PgStr	FStFP byPgStr	FStFP by PgStr
Total secondary parity bits per page stripe	1360	1520	8640	6480	14960	11000

Table IV

[0165] Table V illustrates an example of data allocation for the primary over secondary parity (PoSP) ECC.

	First Gear		Second Gear		Third Gear	
Page Stripe	0, 1, or 2	3	0, 1, or 2	3	0, 1, or 2	3
PoSP Payload (Secondary Parity bytes from 1 stripe)	172	192	1080	812	1872	1376
PoSP codeword length (bits) (*Np*)	388	428	1190	926	1512	1640
PoSP codeword information (bits) (*Kp*)	344	384	1080	816	1248	1376
PoSP parity bits per codeword (*Pp*)	44	44	110	110	264	264
$R = K/N$	0.887	0.897	0.908	0.881	0.825	0.839
Correction capability (*Tp*)	4	4	10	10	24	24
PoSP codewords per page stripe (Wposp)	4	4	8	8	12	8
Fractional Shortening	FStFP by PgStr	FStFP by PgStr	FStFP by PgStr	FStFP by PgStr	FStFP by PgStr	FStFP by PgStr
Total PoSP parity bits per page stripe	138112	75160	137968	75328	137760	75072

Table V

[0166] Table VI illustrate an example of data allocation for the tertiary ECC.

	First Gear	Second Gear	Third Gear
Page Stripe	3	3	3
Tertiary Payload (Primary CW, Secondary Parity, PoSP Parity Bytes)	61208	61208	61208
Tertiary codeword length (*Nt*) RS (bytes)	250	250	250
Tertiary codeword information (bits) (*Kt*)	226	226	226
Tertiary parity bits per codeword (*Pt*)	192	192	192
$Rt = Kt/Nt$	0.904	0.904	0.904
Correction capability (*Tt*) bytes	12	12	12
Tertiary (RS) symbols (bytes) per page per codeword (Wt)	16	16	16
Fractional Shortening	FStFP by Grid	FStFP by Grid	FStFP by Grid

Table VI

[0167] Table VII illustrates an example of data allocation for the PoTP ECC.

	First Gear	Second Gear	Third Gear
Page Stripe	3	3	3
PoTP Payload (Tertiary Parity Bytes)	6528	6528	6528
PoTP codeword length in bytes (*N*)	197	197	197
PoTP codeword information in bytes (*K*)	164	164	164
PoTP parity bits per codeword (*P*)	264	264	264
R = K/N	0.832	0.832	0.832
Correction capability (*T*)	24	24	24
PoTP codewords per page stripe (Wpotp)	40	40	40
Fractional Shortening	FStFP by Page	FStFP by Page	FStFP by Page
Total PoTP parity bits per page stripe	10560	10560	10560

Table VII

[0168] When allocating bits to ECC dimensions, such as those set out in tables III, IV, V, VI, and VII, for a given dimension payload *Ix*, codeword payload *Kx*, and number of codewords *Wx*, the total payload *Wx*Kx* is typically not an integer multiple of the codeword payload. That is, *Kx*Wx – Ix* >= 0. In this case "Fractional Shortening to Fit Payload" or FStFP, is applied. First, *Ix* is padded to be an integer number of bytes. Then, a

prearranged set of (*Kx***Wx* - *Ix*)/8 of the codewords are each shortened by one additional byte. It will be appreciated that a granularity different from one byte can be used. In this context, *x* can be any of *p*, *s*, *t*, *PoSP*, or *PoTP*.

[0169] When a read operation from the flash memory is performed by the SSD, it is a function of the ECC decoder, for example, the decoder portion of an ECC encoder/decoder 1310 (Figure 13), to detect and correct errors that may exist in the data read, and return the corrected data of interest to a journaling engine 1308 (Figure 13).

[0170] Figure 8 illustrates a procedure that can be used to recover data including user data (information) 401 (Figure 4), 501a-501d (Figure 5), 701a-701d (Figure 7) in the presence of a relatively large bit error rate. The reference characters will be reused from those described earlier in connection with Figures 5-7, and it will be understood that these reference characters may refer to where the original data came from and can instead be copied to and processed in scratchpad memory, such as the RAM 1306 (Figure 13), rather than in flash memory. In addition, in one embodiment, primary error correction can be performed between the reading of information from the flash memory and the temporary storage of the information to the scratchpad memory. Other data can also be recovered as will be explained below. It will be appreciated by the skilled practitioner that the illustrated process can be modified in a variety of ways. For example, in another embodiment, various portions of the illustrated process can be combined, can be rearranged in an alternate sequence, can be removed, or the like. For example, in an alternate embodiment without secondary ECC, or without tertiary ECC, the states and decision blocks associated with the secondary ECC can be eliminated from the process. At the start of the process, it is assumed that data and various primary, secondary, and tertiary ECC has been stored to the corresponding flash page grid 600 as described earlier in connection with Figures 4-7.

[0171] The illustrated process can begin either at a stage 801 or a stage 806. For the sake of efficiency and minimum latency, one embodiment of the process preferably calculates the minimum set of primary codewords (the "partial page") for recovering the requested data operating under the assumption that secondary and tertiary decoding will not be necessary. Typically, a page will have no errors to relatively few errors that can be corrected via decoding of primary codewords, so this assumption makes for good efficiency and low latency for most situations. To implement this assumption, the process starts at the

stage 801 in which the process receives a request for user data. It will be understood that due to journaling activities / flash translation layer activities, that the process may need to translate an address associated with the requested data with an actual address as stored in the flash memory. The process advances from the stage 801 to a stage 802.

[0172] In the stage 802, the process reads a partial page stripe, which can be less than a page stripe depending on how much data is requested. The partial page stripe contains the primary codewords corresponding to the requested data. For example, if only one byte of data is requested, then a single page may hold the corresponding user data. In another example, a range of data can be requested, which can take less time to retrieve from the page stripe than reading the entire page stripe. At least one page from the flash memory is read. The process advances from the stage 802 to a stage 803.

[0173] In the stage 803, the process performs primary error detection and correction, as necessary, using the primary parity on user data. For example, the primary codewords can be retrieved from a combination of reading the user data 501a-501d, 701a-701d and the primary parity on user data 502a-502d, 702a-702d. It will be understood that if the foregoing refers to reading a partial page stripe of the page stripes 550a, 550b, 550c, 640. For example, the process can use conventional ECC decoding techniques to detect and possibly correct errors in the user data 501a-501d, 702a-702d using the primary parity on user data 502a-502d, 702a-702d, but new forms of ECC decoding that are yet to be developed may also be applicable. Of course, the decoding process should match the encoding process. The process advances from the stage 803 to the decision block 804.

[0174] In the decision block 804, the process evaluates whether the data decoded in the stage 803 is reasonably likely to be error-free. In one embodiment, the criterion for establishing that the data was decoded in an error-free manner with high confidence includes the following: (a) every primary codeword containing requested data is correctable; and (b) the number of bits corrected in every primary codeword containing requested data is less than or equal to a parameter *Qp* (see Figure 11) for the current gear; and optionally (c) no primary codeword containing requested data contains all-zeroes after correction. A codeword with all zeroes information bits and all zeroes parity is a valid BCH codeword, but it can be produced by a physical error. Thus, in one embodiment, a codeword of all zeroes is initially assumed to be erroneous. Sometimes, the actual user data to be stored will correspond to all zeroes. To

differentiate all-zeroes user data from a physical failure, in one embodiment, the incoming user data is passed through a scrambler prior to storage in the flash memory. Scrambling can be implemented by exclusive-ORing (XORing) a pseudo-random sequence in a known, predictable pattern with the incoming data such that long runs of ones or zeroes are avoided. The XORing is performed prior to storing the user data and prior to calculating parity symbols. The original user data can simply be recovered by XORing the same pseudo random sequence and the recovered (corrected) data from the flash memory.

[0175] A problem in MLC flash storage is that certain data patterns can cause rapid flash wear out as well as increased error rates on other pages. An example of a troublesome data pattern is all ones on one page, and all zeroes on a "matching" page. In a situation in which many users have data on a flash array, a malicious or unfortunate user could damage the flash array by deliberately storing data with these patterns, thus increasing the risk of data loss for all users of the flash array. There are a number of ways of eliminating this problem. One, known in the art, is to encrypt the data before writing it to the flash. However, the cost, power, and latency impact of encryption and key management is not always desirable. In the absence of encryption, it is known in the art that large data patterns can be removed using a scrambler (for example, XORing the data with the output of a pseudorandom pattern generator). However, if a malicious user is able to guess the scrambler sequence, the malicious user can submit data in the form of the desired malicious pattern XORed with the scrambler sequence, with the result that the malicious pattern is written to flash.

[0176] In accordance with one embodiment of the invention, prior to ECC parity calculation, data to be written is XORed with the output of a pseudorandom bit sequence. The initial value (seed) of the pseudorandom bit sequence is determined from information available to the algorithm, but which the end user, including a malicious user, cannot know. In one embodiment of the invention, the seed is generated by taking a digest (hash) of the physical address at which the data is to be stored. In a specific embodiment, the seed is 16 bits, is generated separately once for every 536-byte scrambler or journaling "cell" of data stored by the journaling engine, and can be generated as expressed in Eq. 14.

$$\begin{aligned} &\text{Seed}[15:0] = (\text{block\_address}[7:0]; \text{page\_address}[7:0]) \\ &\text{XOR}(7'b0; \text{chip\_enable}[2:0]; \text{cell\_num}[4:0]); \text{odd\_parity} \end{aligned} \qquad \text{Eq. 14}$$

[0177] In Eq. 14, ";" denotes bitwise concatenation, "7'b0" indicates seven bits of value 0, "XOR" denotes a bitwise exclusive OR operation, cell_num is the address of the journaling cell of data within a page stripe, and odd_parity represents a bit calculated so that there are an odd number of ones in the 16-bit vector. Table VIII illustrates the number of journaling cells per page stripe for each of the 4 page stripes and 3 gears of the illustrated embodiment. The rightmost column of Table VIII contains the total number of journaling cells per page grid.

Page Stripe	0	1	2	3	Total (Page Grid)
First Gear	30	30	30	16	106
Second Gear	27	27	27	14	95
Third Gear	21	21	21	11	74

Table VIII

[0178] The precise allocation of bits to the seed is not particularly important, so long as the seed is never zero; the seed is different for most pages; the seed is readily calculated with data that is available even after a sudden loss of power; and a user does not have any way of knowing or routinely guessing the entire seed. For example, when the journaling engine 1308 (Figure 13) is used, the user should not be able to know, for example, the physical page address because of the translation by the journaling engine 1308.

[0179] If the correction is performed with high confidence, process proceeds from the decision block 804 to a stage 805 in which the data is provided to the host and is indicated as correct. If the correction is not performed with high confidence, then the process proceeds to the stage 807 in which a page stripe is read from the flash memory to begin decoding of the page stripe. For example, any of the page stripe 550a, the page stripe 550b, the page stripe 550c, or the page stripe 640 can be decoded. In an alternative embodiment, the process can begin in the stage 806 and initiate a read of the page stripe 550a, 550b, 550c, 640 (Figure 6). When this occurs, the process reads the entire page stripe first, without an initial partial page stripe read. The process advances from the stage 807 to a stage 808.

[0180] In the stage 808, each primary codeword in the page stripe is decoded to perform primary error correction for the page stripe. The stage 808 is similar to the stage 803, except that the primary error detection and correction is performed over the page stripe in the stage 808. A page stripe can be of the first type exemplified by the page stripes 550a, 550b, 550c or the second type exemplified by the last page stripe 640. If a page stripe happens to be of the second type exemplified by the last page stripe 640 of the page grid 600, decoding of the primary ECC codewords is performed on the user data 701a-701b with the primary parity on user data 702a-702b, on the secondary parity 703a-703d using the PoSP 704a-704d, and optionally the tertiary parity over page grid 705a-705d with the PoTP 706a-706d. For efficiency, the PoTP 706a-706d need not be decoded in the stage 808 because depending on how the page stripe decoding portion 308 of the process fares, it may not be needed. The PoTP 706a-706d can be used later in the page grid decoding portion 310 of the process.

[0181] If a page stripe happens to be of the first type exemplified by the page stripes 550a, 550b, 550c of the page grid 600, then there are no tertiary parity bits so decoding of the primary ECC codewords is performed on the user data 501a-501b with the primary parity on user data 502a-502b and on the secondary parity 503a-503d using the PoSP 504a-504d. The process advances from the stage 808 to a decision block 809.

[0182] In the decision block 809, the process determines whether the error correction, if any, was performed with high degree of confidence that the error correction is without error. The criterion used in the decision block 809 to establish the high degree of confidence can include the following: (a) every primary codeword is correctable; and (b) the number of bits corrected in every Primary codeword is Qp or less; and (c) previous secondary decoding was completed without corrections, and with no more than one uncorrectable.

[0183] If the correction of the codewords containing requested data has been performed with high confidence, the process proceeds from the decision block 809 to a stage 830 for evaluation of a possible change in gear (strength of ECC) and then to the stage 805 in which the data is provided to the host. See, for example, the decision blocks 1132, 1134, 1136, the set next gear flag stages 1142, 1144, 1146, and the decision blocks 1172, 1174, 1176, which will be described later in connection with Figure 11. If the codewords have not

been decoded with high confidence, then the process proceeds from the decision block 809 to a decision block 810.

[0184] In the decision block 810, the process evaluates a stopping criterion and determines if an impasse has been reached. In one embodiment, a stopping criterion is used to halt the iteration of the loop from the stage 808 to the stage 813 if at least one of the following is true: (a) a loop counter has reached a predetermined limit, that is, a maximum iteration limit; or (b) no new corrections were made for the stage 808 in which primary decoding is performed or for the stage 812 in which secondary decoding is performed, or (c) the changes performed in two successive executions of the stage 808 were the same. In one embodiment, the predetermined limit is 12 iterations. Other limits can be used and will be readily determined by one of ordinary skill in the art. If an impasse has been reached, the process proceeds from the decision block 810 to a stage 814 to begin decoding of a page grid. For example, the page grid 600 can be decoded. If no impasse has been reached, the process proceeds from the decision block 810 to a stage 811.

[0185] In the stage 811, the corrected, but with low confidence, data (not exclusively user data) is re-interleaved into a set of secondary codewords. The secondary codewords can correspond to, for example, the user data 501a-501d (Figure 5) and the secondary parity 503a-503d (Figure 5) or to the user data 701a-701d (Figure 7) and the secondary parity 703a-703d (Figure 7). The process advances from the stage 811 to a stage 812.

[0186] In the stage 812, error correction is performed on the secondary codewords according to the particular ECC used for encoding. The secondary codewords can include, for example, the user data 501a-501d, 701a-701d and the secondary parity 503a-503d, 703a-703d or the primary parity on user data 502a-502d, 702a-702d and the secondary parity 503a-503d, 703a-703d. For example, the process can utilize the secondary parity 503a-503d, 703a-703d to correct the user data 501a-501d, 701a-701d and the primary parity 502a-502d, 702a-702d. The decoding of the secondary codewords should be performed sequentially after and not in parallel with the stage 808 in which decoding of the primary codewords is performed. For example, in the stage 808, PoSP 504a-504d, 704a-704d is used to detect and correct errors in the secondary parity 503a-503d, 703a-703d. The process advances from the stage 812 to a stage 813.

[0187] In the stage 813, the process de-interleaves the secondary codewords and returns the corrected data to the format or order used for decode of the primary ECC. The process returns to the stage 808 to continue iteration of the loop.

[0188] If the page stripe stopping criterion of the decision block 810 has been met, the process proceeds to perform error correction on the page grid 600, that is, perform the page grid decoding portion 310 of the process. The various decoding processes can be performed in hardware, software/firmware, or combinations of hardware and software/firmware. The relatively rarity of execution of the page grid decoding portion 310 indicates that it can be practical to implement the partial page stripe decoding portion 304 and the page stripe decoding portion 308 in hardware, and implement the page grid decoding portion 310 in software/firmware. The page grid decoding portion 310 of the process depicted in Figure 8 proceeds along the assumption that none of the pages of the page grid 600 have failed. A different process for page grid decoding will be described later in connection with Figure 10 for the decoding of a page grid 600 with a failed page. Returning now to the page grid decoding portion 310 of the process, in the stage 814, the other page stripes or "supporting page stripes" of the page grid are read. The supporting page stripes are the other page stripes from the same page grid 600 such that the supporting page stripes are page stripes from which user data is not specifically requested, but yet contain relevant secondary and/or tertiary ECC. The process advances to a stage 815.

[0189] In the stage 815, the supporting page stripes are decoded. In one embodiment, the stage 814 and the stage 815 can be performed iteratively. In the stage 815, each supporting page stripe is preferably individually ECC decoded at the page stripe level similar to the previously described page stripe decoding process described in the stages 807, 808, 811, 812, 813 and the decision blocks 809, 810. One embodiment of decoding a supporting page stripe process will be described later in connection with Figure 9. The process of Figure 9 can be repeated until all the supporting page stripes have been decoded. The process advances from the stage 815 to the stage 816.

[0190] In the stage 816, the process performs interleaving to rearranges the user data 501a-501d, 701a-701d and tertiary ECC 705a, 705d into tertiary codewords. The process advances from the stage 816 to a stage 817.

[0191] In the stage 817, the process performs error correction of the tertiary codewords using the tertiary parity 705a-705d. For example, when encoded according to Reed-Solomon (RS) codes, conventional techniques for decoding RS codes can be used. The process advances from the stage 817 to a stage 818.

[0192] In the stage 818, the process returns the corrected block data of the tertiary codewords to the native format of the page grid 600 or order of the user data 501a-501d, 701a-701d, primary parity on user data 502a-502d, secondary parity 503a-503d, PoSP 504a-504d by de-interleaving the tertiary codewords and stores them in scratchpad memory. Of course, the original page structure need not be precisely reproduced to recover the data, but it is convenient to explain and implement the process is this manner. The process advances from the stage 818 to a stage 819.

[0193] In the stage 819, the process interleaves the recovered secondary parity 503a-503d into a format convenient for secondary decoding. The convenient format can be to arrange the secondary parity 503a-503d with other data into secondary codewords. The PoSP 504a-504d, 704a-704d can also be used later during the stage 822 to detect and/or correct errors in the secondary parity 503a-503d, 703a-703d. The process advances from the stage 819 to a stage 820.

[0194] In the stage 820, the process performs error correction of the secondary codewords using the secondary parity 503a-503d, 703a-703d. The result of the decoding of the secondary codewords yields corrected data for the user data 501a-501d, 701a-701d and primary parity on user data 502a-502d, 702a-702d. The process advances from the stage 820 to a stage 821.

[0195] In the stage 821, the corrected data is de-interleaved to return it to the native format of the user data 501a-501d, 701a-701d and primary parity on user data 502a-502d, 702a-702d. The process advances from the stage 821 to a stage 822.

[0196] In the stage 822, the process performs error correction using the primary codewords, that is, using the data of the user data 501a-501d, 701a-701d and the primary parity on user data 502a-502d, 702a-702d. It should be noted that the codewords do not have to be specifically formed but can be constructed as needed based on, for example, pointers to scratchpad memory locations. The process advances from the stage 822 to a decision block 823.

[0197] In the decision block 823, the process determines whether or not the results of the error correction in the stage 817, the stage 820, and the stage 822 were completed with a relatively high degree of confidence. The criterion used in the decision block 823 to establish the high degree of confidence includes the following queries, which taken together are sufficient to exit the decision block 823 with high confidence: (a) every primary codeword is correctable; and (b) the number of bits corrected in every Primary codewords is *Qp* or less; and (c) previous secondary decoding was completed without corrections, and with no more than one uncorrectable, and (d) previous tertiary decoding was completed without uncorrectable codewords being detected. In some cases, such as for very high values of Qp, it may be acceptable to release the data without (c) or (d) being true.

[0198] If the process determines that a high confidence of a correct decoding operation has been achieved, then the process proceeds from the decision block 823 to a stage 832 to set the next gear flag (see the next gear flag indication 1142, 1144, 1146, Figure 11) and then to the stage 805 to release the requested data. In an alternative embodiment, the stage 832 to set the next ear flag is unconditionally performed when the page grid decoding portion 310 of the process is invoked. If the error correction has not been performed with high confidence, then the process proceeds from the decision block 823 to a decision block 824.

[0199] In the decision block 824, the process evaluates a stopping criterion and determines if an impasse has been reached. In one embodiment, a stopping criterion is used to halt the iteration of the loop from the stage 816 to the stage 822 if at least one of the following is true: (a) a loop counter has reached a predetermined limit, that is, a maximum iteration limit; (b) no new corrections were made for any of the stages 817, 820, 822 in which tertiary, secondary, or primary decoding is performed; or (c) two successive iterations of stage 822 were identical. In one embodiment, the predetermined limit is 12 iterations. Other limits can be used and will be readily determined by one of ordinary skill in the art. If an impasse has been reached, the process proceeds from the decision block 824 to a stage 825 to attempt page failure recovery. One embodiment of the stage 825 will be described in greater detail later in connection with Figure 10. If no impasse has been reached, the process returns from the decision block 824 to the stage 816 to proceed with another iteration of tertiary decoding.

[0200] Figure 9 illustrates an example of decoding a supporting page stripe according to an embodiment of the invention. Figure 9 shows greater detail of the stage 815 described earlier in connection with Figure 8. The supporting page stripes of a page grid 600 are the page stripes of the page grid 600 that do not contain the data requested in the stage 801 or in the stage 806. However, the supporting page stripes are of the page grid 600 that does contain the data requested in the stage 801 or in the stage 806. The process 900 is similar to that of the page stripe decoding process represented by a dashed block 308 (Figure 8), except that the data is not released as it would be in the stage 805 (Figure 8). Instead, the data can be temporarily stored in the scratchpad memory for later use by the stage 816 (Figure 8).

[0201] In the stage 902, each of the supporting page stripes in the page grid 600 is read from the flash memory in a manner similar to that of the stage 807 (Figure 8). In the stage 903, primary error correction is performed similar to that of the stage 808 (Figure 8). For the last page stripe, the stage 903 also includes decoding of the primary codewords that cover the tertiary parity bits 705a-705d. In the decision block 904, the criteria used to determine if the page stripe decoding has been successful or not can be the same as that used in the decision block 809 (Figure 8). In the decision block 905, the criterion used to determine if further decode is futile or not can be the same as that used in the decision block 810 (Figure 8). If not futile, then the data is re-interleaved in the stage 906 into a set of secondary codewords in a manner similar to that performed in the stage 811 (Figure 8). Error correction is performed on the secondary codewords according in the stage 907 in a manner similar to that of the stage 812 (Figure 8). After error correction, in a stage 908, the process de-interleaves the data to the format used for primary decode in a manner similar to that of the stage 813 (Figure 8). The process flow returns from the stage 908 to the stage 903.

[0202] When the decode operation is determined to have been successful in the decision block 904 or futile in the decision block 905, then the data for the page stripe is stored in the stage 909 for page grid decoding.

[0203] The page stripe decoding process 900 can be repeated for the remaining supporting page stripes of the page grid. When the process has reached the stage 909 for all

supporting page stripes, then the stage 815 (Figure 8) is complete, and the decoded page stripes are passed to the stage 816 as described earlier in connection with Figure 8.

[0204] If iterative decoding including primary, secondary, and tertiary decoding have failed, it is possible that the reason for the failure is the total failure of a flash page within the grid. This failure may be the result of the total failure of a flash die, or of part of a die (such as a plane, block, or page), or simply a very high BER on one page. In one embodiment, the pages of a page grid 600 are organized such that the page grid 600 contains only one page from a given die, so that even if a die failure occurs, the die failure affects only one page on each page grid 600.

[0205] Figure 10 illustrates an example of a process for attempting a page failure recovery. It will be appreciated by the skilled practitioner that the illustrated process can be modified in a variety of ways. For example, in another embodiment, various portions of the illustrated process can be combined, can be rearranged in an alternate sequence, can be removed, or the like. The process is illustrated using primary, secondary, and tertiary error correction. As discussed earlier, primary error correction provides error correction for a page 400, secondary error correction provides error correction for a page stripe 550, 640, and tertiary error correction provides error correction for a page grid 600. Since the failure of a page can be a relatively rare event as compared to the correction of errors within pages or page stripes, it is practical to implement the tertiary decoding in firmware/software with instructions executed by a processor, or in a block of hardware shared by several decoders.

[0206] While illustrated in the context of an SSD controller using primary, secondary, and tertiary error correction, the principles and advantages described herein are also applicable to systems using fewer error correction levels or even more error correction levels. For example, in an alternative embodiment, the secondary error correction and page stripe arrangement are not utilized and rather, the tertiary error correction is used with the primary error correction and the states in the dashed box 1030 are not used and can thus be considered optional. In such an alternative configuration, the page grid will then contain a collection of pages or integer fractions of pages rather than a collection of page stripes. In another alternative arrangement, tertiary parity bits can be distributed among the pages of a page grid in a manner similar to that of the last page stripe 640 (Figure 7).

[0207] Returning now to the process depicted in Figure 10, at the start of the process, it is assumed that, data has been stored to the flash memory and that the process described earlier in connection with Figure 8 has not resulted in recovered data and has ended in the stage 825. Since the process of Figure 8 has failed to recover data, the possibility is relatively high that there is a failed page in the page grid 600. The failed page will usually be the page for which data is being requested, but can also be another page of the page grid that is needed to complete error correction decoding for the requested page. When a failed page of the page grid 600 can be identified, the error correction capability for the page grid can be dramatically improved. As discussed earlier in connection with the stages 216-220 of Figure 2, in one embodiment, a Reed-Solomon (RS) code is used for the tertiary error correction. A RS code is both an error correction code and an erasure code.

[0208] When correcting errors with RS codes and certain other codes, there is a difference in the ability of the RS ECC decoder to correct errors when the location of the errors is known. For example, an RS ECC decoder can typically correct twice as many erasures than it can correct errors of unknown location because there is information in the location of the errors. As discussed earlier in connection with Figure 2, in one example in which the tertiary symbol size *Mt* is 8 bits or one byte, tertiary codeword size *Nt* is 250 bytes, block data size *Kt* is 226 Bytes, and each tertiary codeword can correct up to 12 symbol errors, up to 24 erasures, or some mix thereof such that the number of erasures plus twice the number of symbol errors is equal to 24. Thus, for example, when blindly correcting errors such as in the page grid decoding portion 310 of the process of Figure 8, the RS ECC decoder is correcting errors of unknown location because the RS ECC decoder does not know which symbols need correction. However, if portions of the data to be corrected can be identified or marked as unknowns or erasures, the RS ECC decoder can then correct more errors than previously done in the page grid decoding portion 310 of the process of Figure 8, thereby recovering a failed page. While illustrated in connection with RS codes, the above-described principle applies to other erasure codes as well, such as to LDPC codes. Additionally, it should be noted that an RS ECC is applicable to symbols and not to individual bits. Typically, each symbol of an RS ECC is a byte (8 bits) of data, but the symbols can be other than bytes. In addition, the process can temporarily store data in scratchpad memory, such as in the RAM 1306 (Figure 13) and can further rearrange the data

as needed. In one embodiment, the process copies data from the page grid 600 (Figure 6) to the RAM 1306 and performs the page recovery processing described in connection with Figure 10 using the contents copied over to the RAM 1306.

[0209] The process of Figure 10 starts in a stage 1001. As described earlier in connection with the fingerprints 407 of Figure 4, the fingerprints 507a-507d, 707a-707d stored for each page can be determined from the physical address of the flash location of that page. For example, a hash code based on the physical address can be used for the fingerprints. The physical address of the page will be known to the SSD controller. Accordingly, the correct or expected values for the fingerprints 507a-507d, 707a-707d is known or is at least determinable prior to data retrieval. In the stage 1001, the process compares the retrieved value for the fingerprints 507a-507d, 707a-707d on each page to its original value. In one embodiment, the page of the page grid 600 having the retrieved fingerprint with the largest magnitude difference from the expected value is identified as the failed page, so long as that difference is greater than a predetermined threshold, such as 5 bits. If this fingerprint comparison is unable to determine a unique failed page, then in one embodiment, the page with the most primary codeword failures can be identified as the failed page. The comparison of primary codeword failures can also be used as a tiebreaker if the fingerprint differences are equal.

[0210] The method is intended to detect many different kinds of failure. For example, an address line can fail, which can result in data from the wrong address being returned. Other forms of failure might return fixed byte patterns (such as all-ones), totally random data, or simply a very high bit error rate. Also, note that the page identified as having failed does not have to be the particular page of the page grid 600 from which data is requested. Alternate forms of identifying a bad page can also be applicable. For example, in another embodiment, a brute force procedure is used and the corresponding data from various pages of the page grid 600 are successively erased for error correction. However, the brute force technique would typically be quite time consuming to execute. The process advances from the stage 1001 to a stage 1016a.

[0211] In the stage 1016a, the tertiary codewords with the erasures can be re-interleaved substantially as described earlier in connection with the stage 816 (Figure 8). The process advances from the stage 1016a to a stage 1002.

[0212] In the stage 1002, the process marks as an erasure or as an unknown, the symbols of the tertiary codewords that are based at least partly on the failed page. In the illustrated embodiment, each symbol is a byte of data, but other symbol sizes can be used. In the illustrated embodiment, the tertiary codewords are interleaved so that the contents of each page are evenly distributed among the tertiary codewords of a grid. The SSD controller will be aware of the particular pattern used for interleaving bits for the various codewords. With knowledge of the pattern, the SSD controller determines which of the symbols making up the tertiary codewords are based on data stored in the failed page. These identified symbols are then marked as unknowns or erasures by the SSD controller for decoding such that for a given tertiary codeword, each symbol, for example, in one embodiment, each byte of a Reed-Solomon ECC defined over Galois Field (GF)(2^8 or 2^8), coming from the failed page is marked as an erasure for the RS ECC decoder. This marking as an erasure can correspond to an instruction to the RS ECC decoder identifying the erased symbols. It should be noted that erasure in this context does not refer to the erasure of a block of flash memory. The process advances from the stage 1002 to a stage 1017.

[0213] In the stage 1017, the process performs tertiary error correction on each tertiary codeword, treating each "erasure" or unknown byte as an assumed error location to perform mixed error and erasure decoding; otherwise 1017 is substantially as described earlier in connection with the stage 817 (Figure 8). The process advances from the stage 1017 to a stage 1018 in which the process de-interleaves the tertiary codewords to return the now corrected block data to the native format of the page grid 600 substantially as described earlier in connection with the stage 818 (Figure 8). The result of the performance of the stages 1016a, 1017, 1018 can differ from that of the stages 816, 817, 818 because the data being corrected is different due to the erasure of the symbols of the failed page performed in the stage 1002. The process advances from the stage 1018 to a stage 1019.

[0214] In the stages 1019, 1020, 1021, the process performs page stripe decoding for the page grid with the secondary error correction substantially as described earlier in connection with the stages 819, 820, 821 of Figure 8. If the outcome of the stage 1017 was improved compared with the stage 817, then the secondary decoding can be expected to perform better than in the process of Figure 8. The process advances from the stage 1021 to a stage 1022. In the stage 1022, the process performs primary error correction substantially

as described earlier in connection with the stage 822 of Figure 8. The process advances from the stage 1022 to a decision block 1023.

[0215] In the decision block 1023, the process evaluates the results of the primary, secondary, and tertiary decoding operations. If the release criteria for the decision block 823 are met, or if all of the following are true: (a) the maximum number of iterations has been reached, such as 12 iterations, (b) the primary decoding has no uncorrectable errors, and (c) the tertiary decoding has no uncorrectable errors. If release criteria are met, the decoding process is complete, and the process proceeds from the decision block 1023 to a stage 1033 in which the data is released.

[0216] If the correction has not been performed with high confidence and data should not be released, then the process proceeds from the decision block 1023 to a decision block 1024 to evaluate a stopping criterion. In one embodiment, the stopping criterion is the same as for the decision block 824. If the stopping criterion is not satisfied, another iteration of the loop is initiated, with the process proceeds from the decision block 1024 to the stage 1016b to perform tertiary error correction without taking erasures into account. With additional symbols corrected from secondary and primary error correction, a subsequent tertiary error correction can be more successful than an initial or an earlier pass. In the stage 1016b, the at least partially corrected tertiary codewords can be re-interleaved substantially as described earlier in connection with the stage 816 (Figure 8).

[0217] The process proceeds from the decision block 1024 to an optional decision block 1032, which determines whether or not the stopping criterion of the decision block 1024 had been satisfied for the first time or the second time for the page grid. If the stopping criterion had been reached for only the first time, the symbols of the failed page are considered to be unknown a second time and the process optionally returns from the decision block 1032 to the stage 1016a to execute a second time. Iteration counters are also set back to zero except for the stopping criterion of the decision block 1024. The reason for performing the optional second execution, which appears to perform an apparently identical process, is that the primary and secondary correction steps may have cleaned up random errors to a sufficient extent that a second erasure pass can fix the remaining problems. The second pass also tends to eliminate problems arising from Tertiary false-corrections occurring during the first execution of 1002 and 1017.

[0218] If, on the other hand, the stopping criterion has been satisfied for a second time, then the decoding operation has been a failure and the process proceeds from the decision block 1032 to a stage 1034 in which a read response indicating an uncorrectable block is returned to the host. Further attempts to repeat can be attempted, but the number of attempts should be limited to avoid an infinite loop.

[0219] Flash page failures can take many forms. The data may be returned in the form of all-zeroes or all-ones. The data can be totally random. The data may also, due to an address decoder failure, be data from the wrong page.

[0220] In at least two of these cases (all-zeroes and data from the wrong page) the primary codewords are likely to indicate that the data is correctable. A rule of the decision block 804 forbidding quick release of an all-zeroes codeword is intended to avoid mistaking an all-zeroes page due to a flash failure for a correct page. An all-zeroes codeword can have an all zeroes checksum such that failure of the page can be difficult to detect. The likelihood of a genuine all-zeroes codeword can be minimized through the use of a scrambler as discussed earlier in connection with the decision block 804 of Figure 8.

Adaptive ECC

[0221] Figure 11 illustrates an example of flash page ECC gear progression. An ECC gear is selected upon initialization and upon use. Figure 11 illustrates gear progression for both scenarios, which can be considered two separate processes. The processes can be performed by the SSD controller 1300 (Figure 13). It will be appreciated by the skilled practitioner that the illustrated process can be modified in a variety of ways. For example, in another embodiment, various portions of the illustrated process can be combined, can be rearranged in an alternate sequence, can be removed, or the like. In the illustrated example, three levels (gears) of ECC capability are used. In the illustrated embodiment, each gear or strength level has a different ratio R of information bits K to total bits N. In alternative embodiments, two levels are used or more than three levels are used. A set of criteria associated with the gear can be used to select the next gear or to retire a page, a page stripe, or a page grid. Advantageously, the retiring of a page, a page stripe, or a page grid can correspond to a relatively small amount of memory that is retired at a time and can be used to retire, for example, upper pages earlier than lower pages, rather than entire blocks of pages. In one embodiment, the next gear is selected if more than $gx\_thresh = Gx*stripe\_size$ bits

were corrected during a decode of a page stripe, or if more than a predetermined number *gx_iter_thresh* iterations of primary error correction and secondary error correction were required for a successful decode at the page stripe decoding portion 308 of the process. In one embodiment, *stripe_size* = 4320*8*4=138240 bits, *G1* is 1.3E-3, and thus *g1_thresh* = 180 bits; *g1_iter_thresh* = 3.

[0222] In an alternate embodiment, the gear progression can include a change of an ECC type to increase the ECC strength instead of or in addition to a change in the ratio *R*. For example, ECC type can include, but are not limited to: Hamming codes, BCH codes, Reed-Solomon codes, LDPC codes, Hadamard codes, Expander codes, Golay codes, and Reed-Muller codes. For example, the type of ECC can be changed from a Hamming code to a BCH code to implement a gear change. The change in type can also be accompanied by a change in ratio *R*. Preferably, when a next ECC technique is selected for a gear change, whether implemented by having a lower ratio *R* and/or a change in type of type, the next ECC technique uses the least overhead, for example, the least number of parity bits of the ECC techniques having a corrective capacity within the RBER observed for the page stripe.

[0223] In addition, extra safety margin can be provided by generation of a safety factor that covers expected variations in RBER. The memory testing techniques can be imperfect and the memory cells themselves are be subject to variability; furthermore error rates can increase if data is retained in the flash for a long period. For example, at the production level, the characteristics of the flash memory cells can be tested to see how much the RBER varies among the page stripes and among a sample of SSDs with memory cells of the same type. The observed RBER can be compared against historical RBER including, for example, the worst-case RBER *Bworst* ever observed on that page or group of pages. An "excursion factor *Ex*" can be calculated based on a cumulative probability density function analysis of the comparison data to compensate for the variability. *Ex* is chosen such that the probability of observing an RBER > *Bworst* is less than aUBER. The excursion factor *Ex* is then multiplied by a "retention factor" *RETx* sufficient to account for increased BER over the desired retention period, to get the safety factor *SFx*. The safety factor is always greater than or equal to 1. The safety factor is used to determine the operational RBER limits for each gear. For example, operational RBER limits *G*1, *G*2, *G*3, for three gears can be selected by dividing the theoretical ECC RBER (*ECC_CAP*) at which UBER < 1E-16 is achieved in

each gear by the safety factor. Thus $Gx \leq ECC\_CAPx / SFx = ECC\_CAPx / (Ex * RETx)$, wherein *Gx* is one of *G*1, *G*2, or *G*3. A safety factor is typically calculated during system test and can be programmed into the SSD controller 1300 or into other non-volatile memory or can, for example, be programmed into the threshold limits. The safety factor *SFx*, the excursion factor *Ex*, and the retention factor *RETx*, can vary in a very broad range based on the flash used and the operational conditions the SSD is required to withstand. Non-limiting examples of values for the safety factor *SFx* include about 2.5, a range of 2 to 3, and a range of 1.5 to 4 for a page grid, an excursion factor *Ex* range of about 4-8 for a page, a range of about 2 to 3 for a page stripe, a range of about 1.4-1.8 for a page grid, and a retention factor *RETx* range of about 1.2 to 2.2. Other values are applicable and will be readily determined by one of ordinary skill in the art. The value *Gx* may be chosen smaller than *ECC_CAPx /SFx* for other reasons, such as to ensure a sufficiently high probability of quick-release.

[0224] Another way of describing the use of the safety factor is that the gear should be selected based on an estimate of the future RBER that will exist when the region of flash is decoded. The safety factor accounts for the uncertainties inherent in that estimate, including retention time, variability of the flash medium, and inaccuracies in measurement.

[0225] The process can maintain one or more tables that maintain the selected gears of the page stripes. The table(s) can be stored in DRAM or scratchpad memory. The table(s) or bins can alternatively be stored in other memory, such as within the flash array itself, or other flash memory within or external to a chip embodying the SSD controller 1300 (Figure 13). The processing of assigning page stripes or other groupings or one or more pages to a particular amount of ECC gearing can be referred to as "binning." In one embodiment, a first table 1102 maintains a list of page stripes that are available, that is, are erased, and have pages and page stripes associated with the first gear of ECC. The list can store a reference to the applicable page stripes, such as, but not limited to, upper address bits of the page stripe. A second table 1104 maintains a list of page stripes that are available and associated with a second gear of ECC. A third table 1106 maintains a list of page stripes that are available and associated with a third gear of ECC. A fourth table 1108 maintains a list of unusable or bad page stripes.

[0226] The payload capacity of a page stripe can be large enough to hold multiple allocation blocks. As discussed previously, in the illustrated embodiment, the payload

capacity of a page stripe depends on the gear level and the type of page stripe. In one embodiment, the process temporarily holds, such as in scratchpad RAM 1306, until there is enough data to fill the page stripe to be programmed or "current" page stripe, and then the process programs the current page stripe with the particular gear level as indicated by the stages 1111, 1113, 1115. In addition, an allocation block can also be programmed into portions of two different page stripes, such as, a first subset of the allocation block can be programmed at the end of one page stripe and a second subset of the allocation block can be programmed to the beginning of another page stripe. Much of the time, however, the entire allocation block can be programmed into a single page stripe. It should be noted that a subset can include the whole. The mapping of particular allocation blocks to page stripes is handled by the journaling engine 1308, which will be discussed later in connection with Figure 12.

[0227] A fifth table 1112 maintains a list of page stripes holding data and are associated with the first gear. A sixth table 1114 maintains a list of page stripes holding data and are associated with the second gear. A seventh table 1116 maintains a list of page stripes holding data and are associated with the third gear. In an alternative embodiment, the data of the tables 1102, 1104, 1106, 1108, 1112, 1114, 1116 can be merged into fewer tables. In the illustrated embodiment, since a page stripe and the pages of the page stripe change gears at the same time, a separate table to track the gear of a page is not needed. However, in alternate embodiments with different gears for pages and their corresponding page stripe, additional tables can be used.

[0228] While the primary, secondary, and tertiary ECC features described earlier in connection with Figures 3-10 can be used with a non-varying strength of error correction, preferably, the SSD controller can vary the ECC strength to accommodate different quality levels of underlying flash memory, to compensate for wearing out from erase cycles over time, to compensate for the difference in RBER based on page index, and the like. The gears for the four page stripes of a grid do not need to be the same. In one embodiment, a process performed by the SSD controller allocates to each page stripe a particular gear (ECC strength) that will, with relatively high certainty (for example, no more than one failure in 1E16 reads), correct all errors that should arise during that program/erase cycle. In the illustrated embodiment, a page stripe can have 1 of 3 ECC levels for its primary and secondary ECC, but the tertiary ECC level is always maintained the same, to ease

implementation of cases in which the four page stripes of a page grid have different gears or ECC levels. In an alternative embodiment, the tertiary ECC level or gear can be determined by the page stripe of the page grid with the highest gear. In the illustrated embodiment, the ECCs gears are changed on a page stripe basis, such that the secondary ECC and the primary ECC strengths are changed at the same time. However, in alternative embodiments, a stripe can be just one page, or gears assigning different ECC strengths to individual pages within a stripe can be defined. In the terminology used in the following, first gear denotes the smallest amount of ECC strength (highest ratio *R*), second gear denotes an intermediate amount of ECC strength, and third gear denotes the largest amount of ECC strength. Of course, the number of gears can vary in a very broad range and can generally be 2 or more.

[0229] In a stage 1110, the process initializes or formats the array, that is, all the memory devices of the SSD. The initialization is typically executed during production test when the flash memory cells are fresh, but can also be executed by an end user to return the SSD controller 1300 (Figure 13) to its factory configuration after the flash memory cells have experienced wear. When the flash array is initialized, a memory test is performed such that test data is written to the page stripes of the flash array, and then read back. For example, an example of a memory test is a pattern resulting from scrambling all zeroes using a synchronous scrambler or the output of a pseudo random bit sequence (PRBS) 31 generator. However, any of a variety of test patterns can be used as will be recognized by one of ordinary skill in the art, and the scrambler can be disabled as appropriate for the memory test. After the memory test is performed, the particular raw bit error rate (RBER) or associated error metric should be known. In the illustrated embodiment, the RBER is determined and maintained at a page stripe level, but in alternative embodiments in which page stripes are not present, the RBER can be maintained at the page grid level.

[0230] An appropriate amount of strength of ECC for a page stripe is selected based on the RBER of the page stripe. The RBER of each page stripe is compared against thresholds chosen for each gear. Page stripes with an RBER less than or equal to the 1st gear threshold *G*1 are assigned the first gear ECC strength and listed in the first table 1102. Page stripes with *an* RBER greater than the first gear threshold *G*1 and less than or equal to the second gear threshold *G*2 are assigned to the 2nd Gear ECC strength and listed in the second table 1104. Page stripes with an RBER greater than the second gear threshold *G2* and less

than or equal to the third gear threshold *G*3 are assigned to 3rd Gear ECC strength and listed in the third table 1106. Page stripes with an RBER greater than the threshold *G*3 are considered unusable and listed in the fourth table 1108. The gear setting for each page stripe can be maintained by the journaling engine 1308 (Figure 13) in memory, such as in DRAM. If this information is lost, the gear ratio can be deduced by attempting a decode operation with each gear, in sequence, starting with the highest gear, until a correct decode is achieved.

[0231] Each gear has a different ratio *R*, and with a fixed physical page sizes, thus corresponds to varying amounts of user data that can be stored per physical page. Ordinarily, a change in page size cannot be tolerated by the host because the host expects to transfer data to and from a data storage device in fixed-size storage blocks. However, by utilizing the journaling engine 1308, the host is not aware of the variation in the amount of information data that can be stored in the underlying physical pages as the characteristics of the physical pages are abstracted away by the journaling engine 1308 or flash translation layer. When the journaling engine 1308 is writing data, it selects a page stripe from its pool of empty flash regions as indicated in the tables 1102, 1104, 1106 and looks up the gear setting for that page stripe, which can be associated implicitly per table 1102, 1104, 1106, or stored separately in a table in nonvolatile memory. Based on the gear setting, the journaling engine 1308 instructs the ECC encoder to encode according to the current gear, passes an amount of data that fits in the information portion of the page stripe to write data 1122, 1124, 1126. The journaling engine 1308 can also remove a reference to the selected page stripe from the tables 1102, 1104, 1106 listing availability and place a reference to the page stripe into the appropriate one of the tables 1112, 1114, 1116. The reference can be, but is not limited to, upper address bits (most significant bits). When the number of user sectors or allocation blocks that fit in a page stripe is not an integer, the journaling engine handles splitting the user sector among two page stripes. The tables 1102,1104,1106, 1112, 1114, 1116 need not literally exist; it is usually more convenient to store information related to page stripes in a way that can be indexed by fill status, physical address and/or age, and to store the gear setting as a field in an entry corresponding to that page stripe.

[0232] In one example of a flash translation layer (FTL) implementation, the FTL or equivalent maintains two lists. One of these lists, analogous to the first table 1102, is for in-service flash regions that can be guaranteed correctable by ECC and another list,

analogous to the fourth table 1108, is for retired regions. An overprovisioning factor *u* can be expressed as *u = (fc - dc)/fc*, wherein *fc* is the payload capacity of in-service flash regions, and *dc* is the drive capacity made available to the user. In the presence of random writes, a write amplification factor *wa = user_bytes_written / flash_bytes_written* is proportional to 1 / (2*u* – *u*^2). As regions of the flash memory are retired, *fc* is reduced by the capacity of the regions retired. This in turn causes overprovisioning *u* to fall, which causes the write amplification factor *wa* to rise. The rise in *wa* impacts the drive's write throughput capacity, and increases the rate of wear. As a result, it is advantageous to take an SSD out of service after *wa* reaches a critical level. In one embodiment, the maximum *wa* permissible is 4.

[0233] Advantageously, one embodiment of the invention maintains a plurality of lists 1102, 1104, 1106 of flash regions associated with different gears as well as the list 1108 of retired regions. In one embodiment, these flash regions correspond to page stripes. When a region of flash is moved from one gear to the next, only a fraction of its former capacity is lost. In one embodiment, the transition from 1st gear to 2nd gear reduces the region's capacity by 10%, the 2nd gear to 3rd gear transition reduces it by a further 20%, and the 3rd gear to retired transition reduces it by the remaining 70%. When this method is employed, the decrease in *fc* due to increasing flash RBER is greatly slowed, which in turn delays the decline of overprovisioning *u*, and decreases the increase in write amplification *wa*. The result is a degree of life extension (as measured in total user bytes written) far in excess of a system that simply uses a high-strength ECC equivalent to 3rd gear throughout the life of the system. This aspect of the invention can only be achieved by a flexible-rate ECC engine working in tandem with a flexible-payload-size journaling engine.

[0234] Whenever a read occurs, the ECC encoder/decoder 1310 performs ECC decoding 1122, 1124, 1126 to perform error detection and correction as needed as previously described. As it does so, the ECC encoder/decoder 1310 can count how many corrections, if any, it makes. This count can be a good approximation of the number of errors and related to a bit error rate (BER) actually existing in the page stripe. The ECC encoder/decoder 1310 compares this amount to the threshold for the current gear. In the decision blocks 1132, 1134, 1136, if the number of corrected bits for a page stripe read exceeds the *gx_thresh* for the current gear, or if the number of primary/secondary iterations exceeds *g_iter_thesh*, the ECC encoder/decoder 1310 informs the journaling engine 1308 that the page stripe should be

assigned a higher gear, such as the next higher gear, the next time the page stripe is programmed. The journaling engine 1308 initially stores this set next gear flag indication 1142, 1144, 1146 for later use. If the condition for the decision blocks 1132, 1134, 1136 is not met, there is no change made to the next gear flag 1152, 1154, 1156.

[0235] A flash block is the smallest physical unit that is erasable in flash memory. In an embodiment of the invention, the journaling engine 1308 will consider the smallest erasable unit to be one or more block grids (a set of flash blocks containing only complete page grids). For example, if grids are constructed of 16 pages, one from each of 16 dies, then a block grid would be 16 blocks, one from each of 16 dies, containing those page grids. When the journaling engine 1308 erases 1162, 1164, 1166 a block grid of flash, the journaling engine 1308 reviews the flags for the corresponding pages of that block in the decision blocks 1172, 1174, 1176. Any page stripes which have the BER flag set are assigned the next higher gear by listing in the appropriate table 1104, 1106, or, in the case of a page stripe already assigned to the third gear, are marked as unusable or listed in the fourth table 1108. Page stripes without the next gear flag set remain in the present gear and are listed in the appropriate table 1102, 1104, 1106. Preferably, the decision blocks 1132, 1134, 1136 and the decision blocks 1172, 1174, 1176 are executed each time they are encountered during execution of the process of Figure 11. However, the execution of the decision blocks can be skipped, for example, executed every other pass, every fourth pass, or the like. This can be useful when the process is implemented in firmware and processor resources are low.

[0236] In an alternative embodiment, the SSD controller 1300 does not use a test pattern generator during initialization but counts corrected errors instead to determine the RBER. Rather than using a test pattern and tabulating errors by comparing the written data and the read data for the memory test, the SSD controller 1300 can instead use virtually any data, such as all zeroes or all ones, which are then scrambled by the scrambler, then encoded at the maximum strength or highest gear ECC, such as third gear, programs the encoded data, reads the encoded data back, and then decodes the retrieve data as the memory test. The maximum strength ECC should be used since it should have the maximum error counting ability. During decoding, the ECC encoder/decoder 1310 counts and correct the errors encountered, if any. The count of the errors is then used to "bin" the page stripe in the

appropriate one of the tables 1102, 1104, 1106, 1108 a based on the thresholds discussed earlier in connection with the decision blocks 1132, 1134, 1136.

[0237] Figure 12 illustrates the operation of the virtual/physical translation table, which is used by the journaling engine 1308 of Figure 13, which will be described in greater detail later.

[0238] Whenever a host wishes to write data to a storage array, the host supplies the data in a fixed-size allocation block along with a virtual address 1201 that identifies that piece of data to the host. The journaling engine 1308 finds an empty region for a write location 1202 to the storage array, and issues commands for the data to be written to that area. The journaling engine 1308 updates the virtual to physical translation table 1204 with the physical address or addresses 1207 corresponding to the virtual address 1201. If the virtual address was previously mapped to a physical address in a full block 1203, that physical address is marked as "invalid." The pointer to the next valid write location 1202 is advanced. When the pointer reaches the end of the current block grid, the pointer is advanced to the next block grid 1205. The journaling engine 1308 maintains a supply of available blocks 1206 by erasing and reclaiming blocks as will be described in greater detail later in connection with Figure 16.

[0239] Figure 14 illustrates a process that the journaling engine 1308 can perform to execute write requests. Whenever a host or other process wishes to write data to a storage array, it supplies the data in a fixed-size allocation block along with a virtual address or logical address 1201 that identifies that piece of data, such as a block stripe, which is made up of the blocks that correspond to a page stripe, to that host. These writes are added to a write command queue 1401. The journaling engine takes write requests 1402 from the write command queue 1401. The journaling engine 1308 maintains a current page stripe buffer 1413 it is journaling into, as well as a next page stripe buffer 1414. The payload capacity of each depends on the gear, which is specified in the page stripe record 1415. Typically, the payload capacity of a page stripe will be much larger than the size of an allocation block, so that several allocation blocks can be written into a page stripe buffer. At the stage 1403, the journaling engine temporarily stores the allocation block in the current page stripe buffer 1413, which can be implemented in volatile memory, such as RAM. If the allocation block runs over the end of the page stripe payload, the remaining portion can be assigned to the

next available page stripe and temporarily stored in the next available page stripe buffer 1414. The journaling engine records the physical location or locations of the data in the virtual (logical) to physical translation table 1204. If the current page stripe is considered to be full 1405, then in a stage 1406 the journaling engine issues a command 202 to the ECC to encode and write the current page stripe to the flash array. It will be understood that the payload portion of a current page stripe can be considered to be full when it is less than full. For example, if the payload portion of current page stripe is nearly full, such as within a few bytes of being full, the performance of the memory will be improved by not splitting a next allocation block across two different page stripes. In addition, when journaling cells are used, it can be desirable not to split a journaling cell across multiple page stripes. Accordingly, there can be a remaining portion of a page stripe that goes unused when it is considered to be full. In a stage 1407, the process moves the record of the current page stripe into the list of full page stripes within the list of full block grids 1411. In a stage 1408, the next page stripe buffer 1414 becomes the current page stripe buffer 1413, and a new next page stripe 1414 is chosen from the list of available block grids 1412. Once the stage 1409 is reached, the write request is complete.

[0240] During writes, the journaling engine 1308 also checks in the stage 1404 to see if there is existing data in the storage array corresponding to the virtual address being overwritten by the operation. If so, it marks the physical address for the previous data as "invalid," as flash cannot be overwritten.

[0241] Figure 15 illustrates a process that the journaling engine 1308 can perform to execute read requests. Whenever data stored in a flash array is to be read, the host or requesting process supplies the virtual address 1201 for the data it wishes to retrieve. The request enters a read command queue 1501. For each read request 1502 drawn from the queue, in a stage 1503, the journaling engine looks up the corresponding physical address in the logical to physical translation table 1204, and issues commands 302 (Figure 3) to the ECC encoder/decoder 1310 for the data at the physical address to be read. The output of the read procedure in Figure 3 is either a corrected read response 306 or an uncorrectable 314. An uncorrectable is a serious failure, and requires exception handling 1504 that is beyond the scope of this disclosure. If a corrected read response 306 is returned, then if the criteria for choosing a higher gear 1505 were met, then a flag is set in the page stripe record 1511 in

thelist of full page stripes within the list of full page stripes 1411. In a stage 1507, the requested data is returned to the requesting process.

[0242] Alternatively, a process can request the read of an entire page stripe, rather than an allocation block. In this case the request enters a queue 1509. Stripe read requests 1510 already have the physical address identified, and proceed immediately to a read request 302.

[0243] In one embodiment, the reading of an allocation block (a partial page stripe) may be small enough that random variation in the measured bit error rate is undesirably large, resulting in an excessive level of "false positives" indicating gear changes at stage 1505, when such gear changes are not required. In this embodiment, BER readings of partial page reads can be ignored, or very high thresholds applied such that the false positive rate is sufficiently low, for example 1E-6. In one embodiment, when a block grid is to be erased, every page stripe is read back whether or not the page stripe contains valid data, and the statistics of the entire page stripe be used in determining whether a different gear needs to be selected.

[0244] In accordance with another embodiment of the invention, the gear setting of each page stripe within a page grid is forced to be identical, such that the gear setting is based on the performance of the worst page stripe of a page grid.

[0245] Figure 16 illustrates a process that the journaling engine can perform to reclaim regions of flash, making them available for writing. Reclamation commands 1601 can be part of garbage collection, scrubbing, and the process performed to maintain the list of available block grids 1412, 1206. In a stage 1602, the journaling engine chooses a block grid to reclaim 1612 from a list 1411 of full block grids. In one embodiment, the block grid chosen for reclamation can be the least recently erased block grid. A page stripe of the block grid is read in a stage 1603 by issuing commands to a stripe read command queue 1509. When the corresponding read response 1507 comes back, the valid allocation blocks stored in the stripe are extracted 1604 and written 1605 back to a flash array via the write command queue 1401. Valid allocation blocks that run over into the next page stripe may be only partially present. Such partial valid allocation blocks are retained in a stage 1606, and are returned to the stage 1604 to be added to the beginning of the next page stripe. A stage 1607 evaluates whether all the page stripes in the block grid have been processed. If not, the

process returns to the stage 1603 to handle a subsequent page stripe. If so, then in a stage 1608, commands are generated to an erase command queue 1610 to erase the blocks of the block grid. After the erase commands are confirmed 1611, the current block grid 1612 and its page stripes are moved to the list of available block grids 1412.

[0246] Figure 13 illustrates an example of a block diagram for a solid-state drive controller 1300 according to an embodiment of the invention. It will be understood that the various blocks or circuits of the SSD controller 1300 can be rearranged, combined, and/or further divided into smaller blocks. In addition, the SSD controller 1300 can be integrated into a single integrated circuit or can be embodied by multiple integrated circuits. The illustrated SSD controller 1300 includes a storage communications interface 1302, a flash management processor 1304, a random access memory (RAM) 1306, the journaling engine 1308, the ECC encoder/decoder 1310, a flash stripe controller 1312, flash protocol controllers 1314, and flash memory devices 1316a-1316d. The SSD controller 1300 is outside of the flash memory devices 1316 and is not the same as the controllers within the flash memory devices 1316a-1316d. In one embodiment, there are at least 16 different flash memory devices 1316a-1316d in the storage array.

[0247] The storage communications interface 1302 handles communication with a host, such as with an input/output circuit of a server. The storage communications interface 1302 can conform to any of a variety of protocols, such as small computer system interface (SCSI), PCI-Express, serial ATA (SATA), external SATA (eSATA), universal serial bus (USB), IEEE-1394 (FireWire), or the like. In one embodiment, the SSD controller communicates with the host at a minimum of 50,000 allocation blocks per second. The flash management processor 1304 can perform firmware functions, translation layer functions, maintains the association between pages and the gear used, initiates initialization bit error rate commands, and the like. The RAM 1306, which can be either SRAM or DRAM, can be used to temporarily hold, that is, cache, data. Other components of the SSD can use the RAM 1306 as needed for scratchpad memory. For example, a page, page stripe, or page grid can be read into the RAM 1306, and the interleaving/de-interleaving and error correction decoding operations can performed on the data transferred to the RAM 1306. The RAM 1306 is typically volatile memory and should accordingly be used only for temporary storage. The journaling engine 1308 performs the journaling or flash translation layer

activities. The ECC encoder/decoder 1310 performs encoding and decoding of information data and ECC parity. The flash stripe controller 1312 provides services, takes a slice of data that has been encoded and distributes it among the *n* lanes and returns an acknowledgement when done. In the illustrated embodiment, 4 lanes 1318a-1318d are shown.

[0248] When performing a read, the flash stripe controller 1312 retrieves the corresponding data and reassembles a page of data. Typically, the data buses for the flash memory devices 1316a-1316d are arranged in lanes, such as lanes of 8 bits wide each. For example, these flash memory devices can correspond to flash memory chips or to packaged flash memory devices. Flash protocol controllers 1314 can provide registers for read and/or write caching, can provide address information, can provide timing signals, read and write signals, and the like. In the illustrated embodiment, a controller from the flash protocol controllers 1314 is instantiated for each bus.

[0249] In one embodiment, the SSD controller 1300 is implemented in an integrated circuit, such as by an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), which can include firmware or software instructions stored in a tangible, non-transitory computer-readable medium executed by a processor, or in a block of hardware shared by several decoders. In particular, portions that are not expected to be used frequently, such as tertiary parity recovery, may advantageously be implemented in firmware or software, or in a block of hardware shared by several decoders. In addition, it should be noted that the disclosed techniques are applicable to use "in the field," such as in a server, and accordingly differ from techniques used by an engineer during a design phase or during manufacturing and test. Further, it should be noted that an integrated circuit for the SSD controller 1300 is separate from the flash memory devices 1316a-1316d.

Overview of a Decoding Process

[0250] The following is an overview of a decoding process according to one embodiment of the invention.

[0251] Partial Page Read Response

1. Perform Primary Decode on received codewords
    a. Decode and perform corrections
2. For each Primary codeword, check whether it can be quick-released.
    a. A codeword may be quick-released if:

i. The codeword is correctable, and
ii. The number of bits corrected for that codeword is less than or equal to *Qp* for the current gear, and
iii. The codeword is not all-zeroes after correction.

b. If all codewords are quick-released:
i. Report a successful partial read response

c. If not all codewords are quick-released
i. Report that the read is being reissued
ii. Reissue a page stripe read.

3. Report Result

[0252] Page Stripe Read Response

1. Perform Primary and PoSP Decode on received codewords
a. Decode and perform corrections
b. Write into Secondary codeword storage
c. Do not Release until Fingerprint has been compared. In practice, this currently means no release during the first pass.
2. Check for Page Kill. (before, during, or after step 1 as convenient)
a. Compare Fingerprint for each page against Expected Fingerprints.
b. If any Fingerprint has more than *Fingerprint_Tolerance* (default = 5) errors, end with *decode_status* = failed.
i. If Tertiary is disabled, *Fingerprint_Tolerance* should be set to maximum, effectively disabling this check.
3. Iterate
a. Secondary Decode
b. Primary and PoSP Decode
i. A codeword may be released if:
1. The secondary passed with zero corrections, and
2. The codeword is correctable, and
3. The number of bits corrected for that codeword is less than or equal to *Qp* for the current gear

c. If all secondary codewords decoded correctly with zero errors, and all Primary codewords were released
    i. End iteration with *Decode_status* = successful
d. (Optional, controlled by flash_initialization_mode bit) If, after primary decode, at least two pages were either all-ones or all-zeroes (including PoSP / PoTP), then end iteration with Decode status = *erased_page*. Do not proceed to Tertiary.
e. (Optional) If there were uncorrectables (not including PoSP or PoTP), but no corrections in Primary or Secondary, we are stuck.
    i. End iteration with *decode_status* = failed
f. (Optional) If all primary (not including PoSP) codewords correct with zero errors, and there are no Secondary corrections, but up to one secondary uncorrectable is seen
    i. End iteration with *decode_status* = successful
g. If max_iterations is reached,
    i. If all Primary codewords (not including PoSP or PoTP) passed, end with *decode_Status* = successful
    ii. End iteration with *decode_status* = failed.

4. Report Result
    a. If Decode = successful, return a success
        i. Set next_gear if
            1. The total # of bits corrected exceeds *stripe_corr_thresh* for the current page, or
            2. *stripe_iteration_thresh* (default = 3) or more iterations were required.
    b. If Decode= *erased_page* then return Uncorrectable, ideally with an *erased_page* flag set. Do not proceed to teritiary.
    c. If Decode = failed, or Fingerprint Tolerance exceeded, reissue page grid. Completion message = reissue.

[0253] Page Grid Read Response

1. Perform Page Stripe Decode on each Page, with the following modifications:
    a. Do not reissue (we already did)
    b. For Last Page Stripe, PoTP should be performed on each Primary pass.
    c. Store the # of Fingerprint errors for each page (to find the largest one later)

2. Perform Tertiary Iteration
    a. Tertiary Decode (no erasures)
    b. Secondary Decode
    c. Primary / PoSP / PoTP Decode
        i. A codeword may be released if:
            1. The Tertiary had no uncorrectables, and
            2. The release conditions for Page Stripe were satisfied.
    d. If no uncorrectable codewords in Tertiary and all Primary codewords (not including PoSP or PoTP) were released
        i. Move to Report Phase with Decode = Success
    e. Optional ; added to match RTL) : If the Secondary or Primary in (b,c) made changes to the page stripe of interest
        i. Secondary decode on stripe of interest
        ii. Primary decode on stripe of interest
        iii. If bit changes made and NOT *max_iterations* reached, return to (i).
    f. If all secondary codewords decoded successfully without corrections, and all Primary codewords (ignoring PoTP) decoded without corrections,
        i. Move to Report Phase with *Decode* = Success
    g. (Optional) If there were uncorrectables in T, S, or P (not including PoSP or PoTP), but no corrections in T, S, or P, we are stuck.

i. End iteration with *decode_status* = failed

h. If max_iterations is reached

i. If no uncorrectable codewords in Tertiary or Primary codewords (not including PoSP or PoTP), end with *decode_status* = Success

ii. End iteration with *decode_status* = failed.

i. Return to top of "Perform Tertiary Iteration"

3. Erase the killed Page

a. If the process has arrived at this point, then the Tertiary iterations have failed. Zero or one erasures have been done.

b. Select a page to erase

i. If at least one page has more than *Fingerprint_Tolerance* errors in its fingerprint, then the one with the most errors in the fingerprint should be erased.

ii. If no page has more than Fingerprint Tolerance, or in case of a tie, then the page with the highest number of Primary Uncorrectables, based on the most recent Primary decode of that page, should be erased

1. Primary Uncorrectables include Pri, PoTP, and (optionally, at the convenience of the RTL) PoSP.

2. For this purpose (only) primary uncorrectables includes primary codewords that were all-zero after correction.

iii. In case of a tie, the one with the lowest index can be arbitrarily picked.

iv. Implementation note: The RTL can arbitrarily choose the one that arrived back first.

c. If *Erasures_done* < 2

i. Increment *Erasures_done*

ii. Erase the page with the most fingerprint errors

iii. Go back to top of "perform tertiary iteration"

d. Else end with *status* = failed.

4. Report Result
    a. If *Decode* = successful, return a success and transfer requested data.
    b. If *Decode* = failed, report permanent failure, with a page kill flag if appropriate, to FSC.
    c. Set *next_gear* flag
        i. If in this state, primary/secondary decoding has failed, so the *next_gear* flag should be set on the original page stripe.

Extensions and Variations

[0254] The choice of page stripes being four lanes wide is a balanced and convenient choice. However, page stripes of other widths for example, as narrow as one page or as wide as 16 pages, are also applicable. Arranging the flash pages on four parallel lanes provides good latency for partial reads, and efficient sharing of the flash interface. However, the principles and advantages described herein are still applicable if the page stripe is distributed over many dice on the same lane, on a random selection of four lanes within a large multi-lane access system, or any combination thereof.

[0255] Reading a minimum amount from the flash memory necessary to decode the user-requested part of the page stripe, and doing a follow-up read of the entire stripe when required is one approach. Another approach that also works is to transfer the entire page stripe during the initial read, starting with that initial minimal set, but continuing on to read the remainder rest of the page stripe, and wrapping around to the beginning of the stripe, and continuing on until the entire stripe is available. In parallel, the initial ECC decode operation is performed. If the quick-release ECC is successful, the transfer of the remaining bytes of the stripe can be aborted. If the quick-release is not successful, then the decoder waits until the rest of the stripe is available, and completes the decode operation.

[0256] The principles and advantages disclosed herein are applicable to flash drives that emulate hard disk drives. Various forms of flash drives include, but are not

limited to: solid-state drives (whether SCSI, SATA, or PCI attached), solid-state caching or tiering devices, enterprise flash drives, hybrid drives (hybrids of hard disk drives and flash drives), flash memory cards, flash memory sticks, thumb drives, backup drives, or the like. For example, various aspects can be incorporated into a solid-state drive controller chip. Alternatively, various aspects can be incorporated with the host, such as, by being incorporated into the same motherboard with other components of the host. A flash translation layer permits a flash drive to emulate a hard disk drive by transferring data to and from the host in allocation blocks. The host can use any of many different file systems, such as, but not limited to: New Technology File System (NTFS), File Allocation Table (FAT), FAT32, High Performance File System (HPFS), Ext2, Ext3, Ext4, Hierarchical File System (HFS), HFS Plus (HFS+), or the like. The host or host computer can be, for example, but not limited to, an enterprise server, a web server, a mobile base station, a host bus adapter, an IP router, a home server, a file server, a workstation, a data logger, a digital video recorder, a smart TV, a media center, a jukebox, a smart phone, a mobile phone, a personal digital assistant, a personal computer, a laptop computer, a tablet computer, a video gaming console, an interactive kiosk, a digital camera, a digital video camera, a digital music player, an ebook reader, or the like.

[0257] Various embodiments have been described above. Although described with reference to these specific embodiments, the descriptions are intended to be illustrative and are not intended to be limiting. Various modifications and applications may occur to those skilled in the art.

WHAT IS CLAIMED IS:

1. A method of selecting an error correction coding (ECC) scheme, the method comprising:

determining a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof;

comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein the first gear has a different data payload size and correction capability than the second gear; and

selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds;

wherein determining, comparing, and selecting are performed by an integrated circuit.

2. The method of Claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds such that the bit error rate is actively monitored.

3. The method of Claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds at least once every 4 program/erase cycles.

4. The method of Claim 1, wherein determining, comparing, and selecting are performed concurrently with at least one of flash initialization, read, or write operations for an end user.

5. The method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

6. The method of Claim 1, wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

7. The method of Claim 6, wherein the aUBER is 1E-16.

8. The method of Claim 1, wherein selecting further comprises selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

9. The method of Claim 1, further comprising:

receiving a command to perform formatting;

performing a memory test to determine raw bit error rates among the regions of the flash drive; and

allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

10. The method of Claim 1, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

11. The method of Claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor *SFx* is in a range from 1.5 to 4.

12. The method of Claim 10, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx = ECC\_CAPx / (Ex * RETx).$$

13. The method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

14. The method of Claim 1, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

15. An apparatus comprising:

a first circuit configured to determine a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof and to compare the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes, wherein the first gear has a different data payload size and correction capability than the second gear; and

a second circuit configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

16. The apparatus of Claim 15, wherein the ECC encoder is configured to determine the bit error rate and the processor is configured to compare and to select concurrently with at least one of flash initialization, read, or write operations for an end user.

17. The apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region.

18. The apparatus of Claim 15, wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

19. The apparatus of Claim 15, wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

20. The apparatus of Claim 15, wherein the processor is further configured to:

receive a command to perform formatting;

perform a memory test to determine raw bit error rates among the regions of the flash drive; and

allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

21. The apparatus of Claim 15, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

22. The apparatus of Claim 15, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performance of ECC decoding with the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

23. The apparatus of Claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

24. An apparatus for selecting an error correction coding (ECC) scheme, the apparatus comprising:

a means for determining a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof;

a means for comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein the first gear has a different data payload size and correction capability than the second gear; and

a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

25. The apparatus of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

26. The apparatus of Claim 24, wherein the selecting means further comprises means for selecting a gear having a smaller data payload size and a higher amount of

correction capability based at least partly a number of decoding iterations used to correct the requested user data.

27. The apparatus of Claim 24, further comprising:

means for receiving a command to perform formatting;

means for performing a memory test to determine raw bit error rates among the regions of the flash drive; and

means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

28. The apparatus of Claim 24, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

29. The apparatus of Claim 24, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

30. The apparatus of Claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

Abstract of the Disclosure

Apparatus and methods provide relatively low uncorrectable bit error rates, low write amplification, long life, fast and efficient retrieval, and efficient storage density such that a solid-state drive (SSD) can be implemented using relatively inexpensive MLC Flash for an enterprise storage application.

Appendix A

Incorporation by Reference of Commonly Owned Applications

The following patent applications, commonly owned and filed on the same day as the present application, are hereby incorporated herein in their entirety by reference thereto:

Title	Application No.	Attorney Docket No.
SYSTEMS AND METHODS FOR INITIALIZING REGIONS OF A FLASH DRIVE HAVING DIVERSE ERROR CORRECTION CODING (ECC) SCHEMES		DATUMTE.076A2
SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING AMONG DIFFERENT ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE		DATUMTE.076A3
SYSTEMS AND METHODS FOR REDUNDANTLY STORING ERROR CORRECTION CODES IN A FLASH DRIVE		DATUMTE.076A4
SYSTEMS AND METHODS FOR LOW LATENCY, HIGH RELIABILITY ERROR CORRECTION IN A FLASH DRIVE		DATUMTE.076A5
SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE		DATUMTE.076A6
SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE		DATUMTE.076A7

Title	Application No.	Attorney Docket No.
SYSTEMS AND METHODS FOR RECLAIMING FLASH BLOCKS OF A FLASH DRIVE		DATUMTE.076A8
SYSTEMS AND METHODS FOR REDUNDANTLY STORING DATA IN A FLASH DRIVE SUCH THAT A FAILED PAGE CAN BE TOLERATED		DATUMTE.076A9
SYSTEMS AND METHODS FOR RECOVERING DATA FROM FAILED PORTIONS OF A FLASH DRIVE		DATUMTE.076A10

[maxlife]
13316155_1
052112

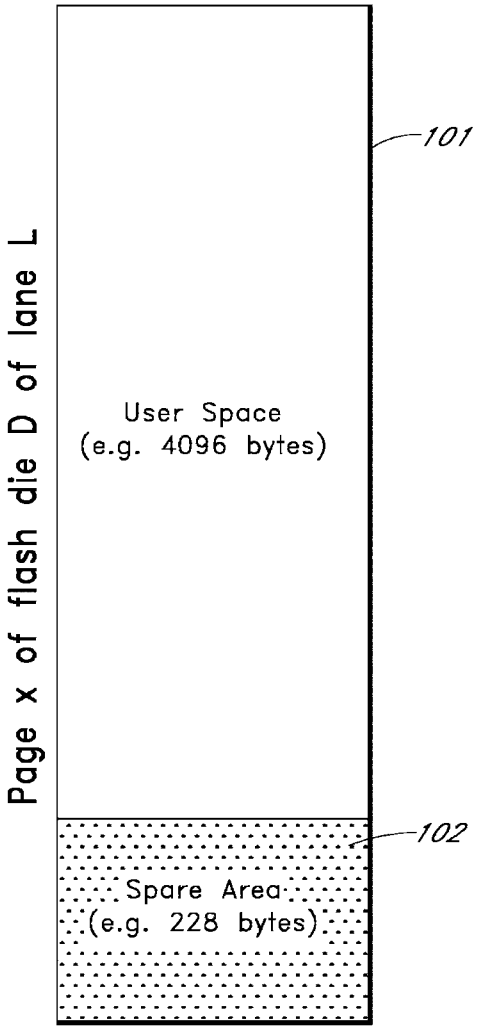


FIG. 1A
(PRIOR ART)

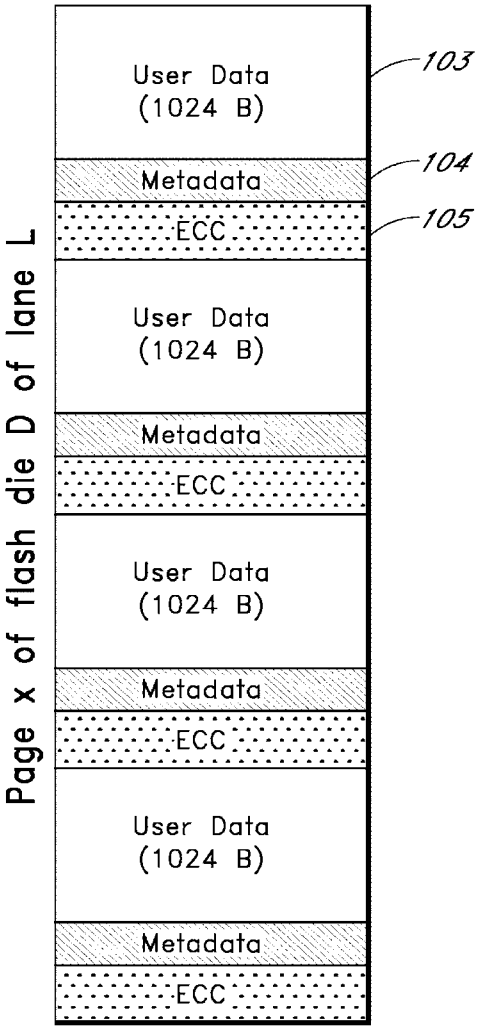


FIG. 1B
(PRIOR ART)

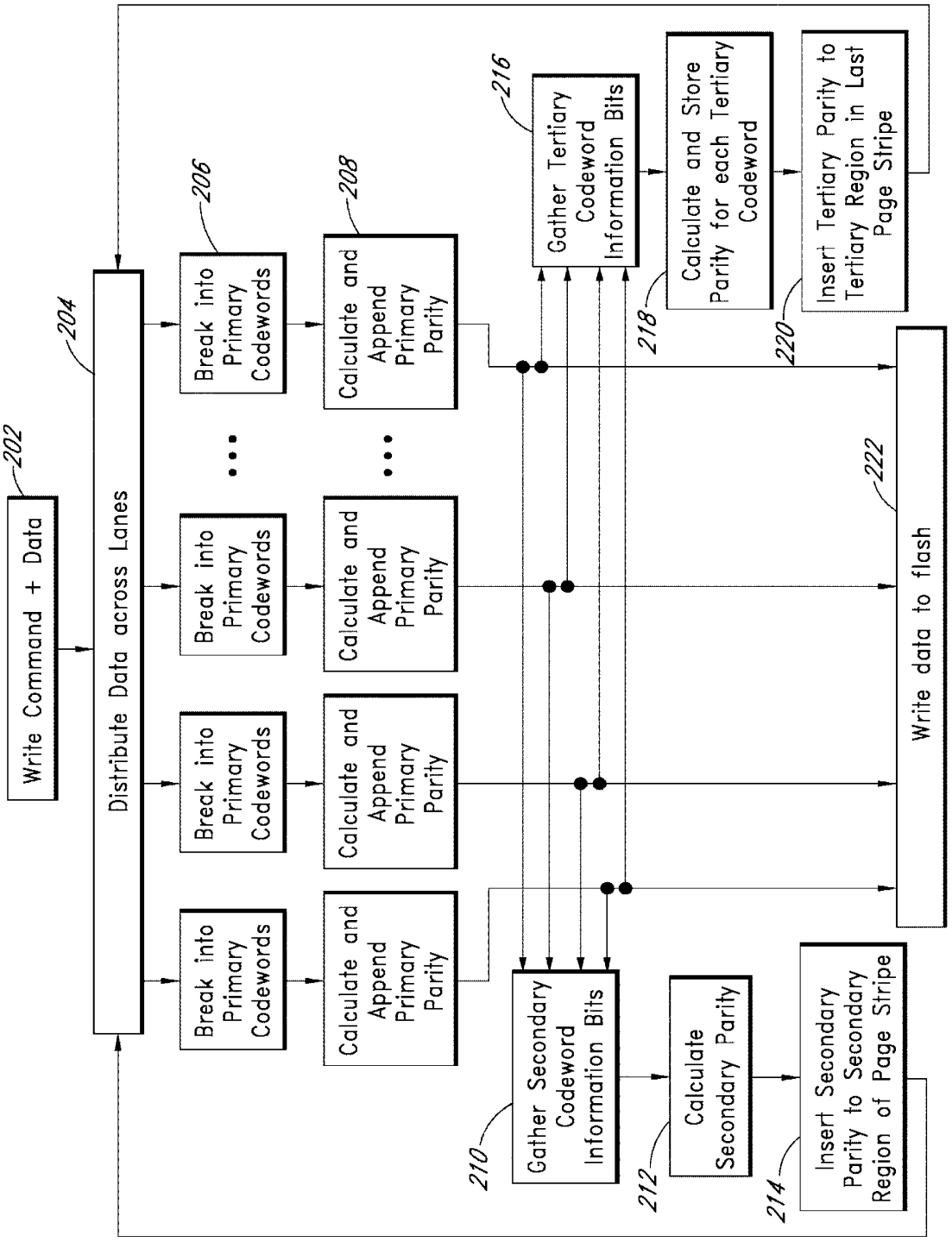

Write Command + Data
202
Distribute Data across Lanes
204
Break into Primary Codewords
206
Calculate and Append Primary Parity
208
Gather Secondary Codeword Information Bits
210
Calculate Secondary Parity
212
Insert Secondary Parity to Secondary Region of Page Stripe
214
Gather Tertiary Codeword Information Bits
216
Calculate and Store Parity for each Tertiary Codeword
218
Insert Tertiary Parity to Tertiary Region in Last Page Stripe
220
Write data to flash
222

FIG. 2

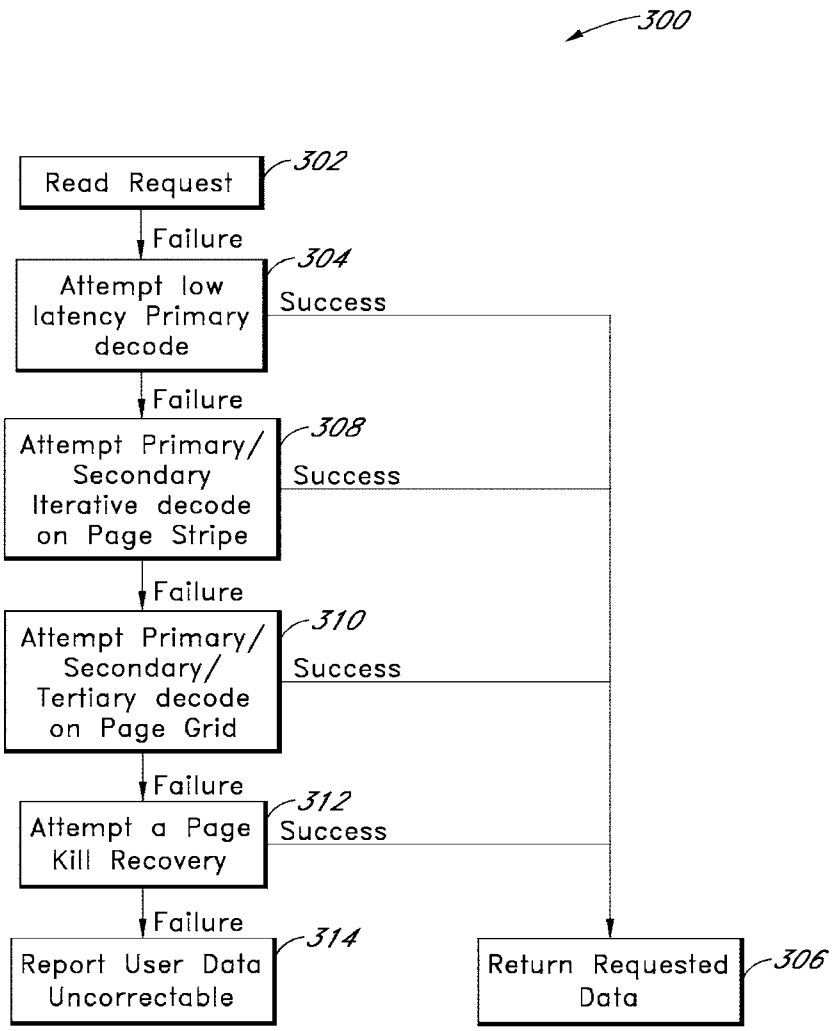

300
Read Request
302
Failure
Attempt low latency Primary decode
304
Success
Failure
Attempt Primary/ Secondary Iterative decode on Page Stripe
308
Success
Failure
Attempt Primary/ Secondary/ Tertiary decode on Page Grid
310
Success
Failure
Attempt a Page Kill Recovery
312
Success
Failure
Report User Data Uncorrectable
314
Return Requested Data
306

FIG. 3

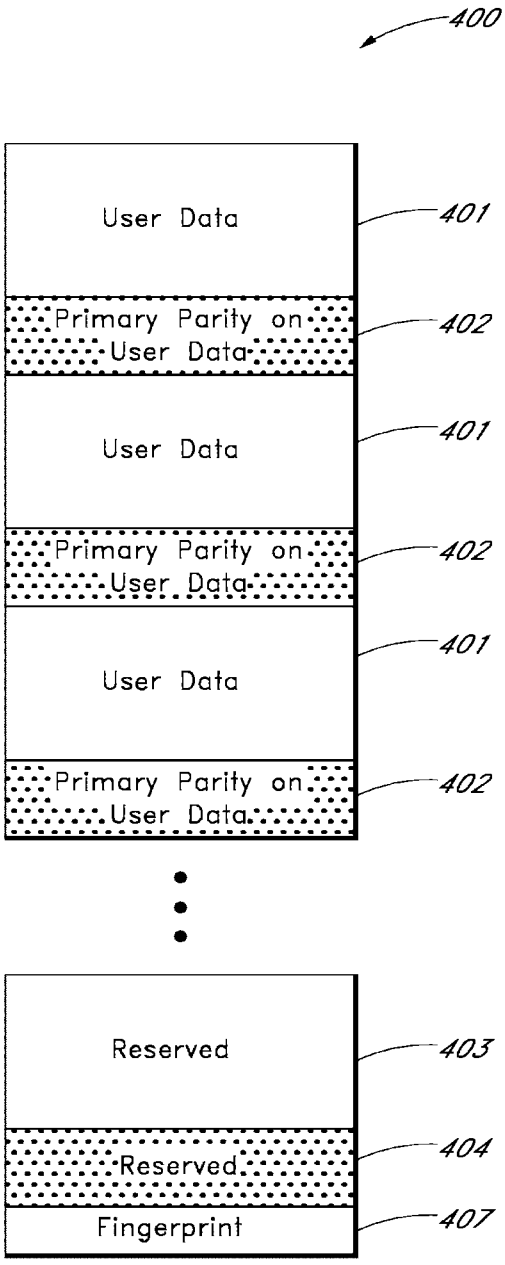

400
User Data
401
Primary Parity on User Data
402
User Data
401
Primary Parity on User Data
402
User Data
401
Primary Parity on User Data
402
Reserved
403
Reserved
404
Fingerprint
407

FIG. 4

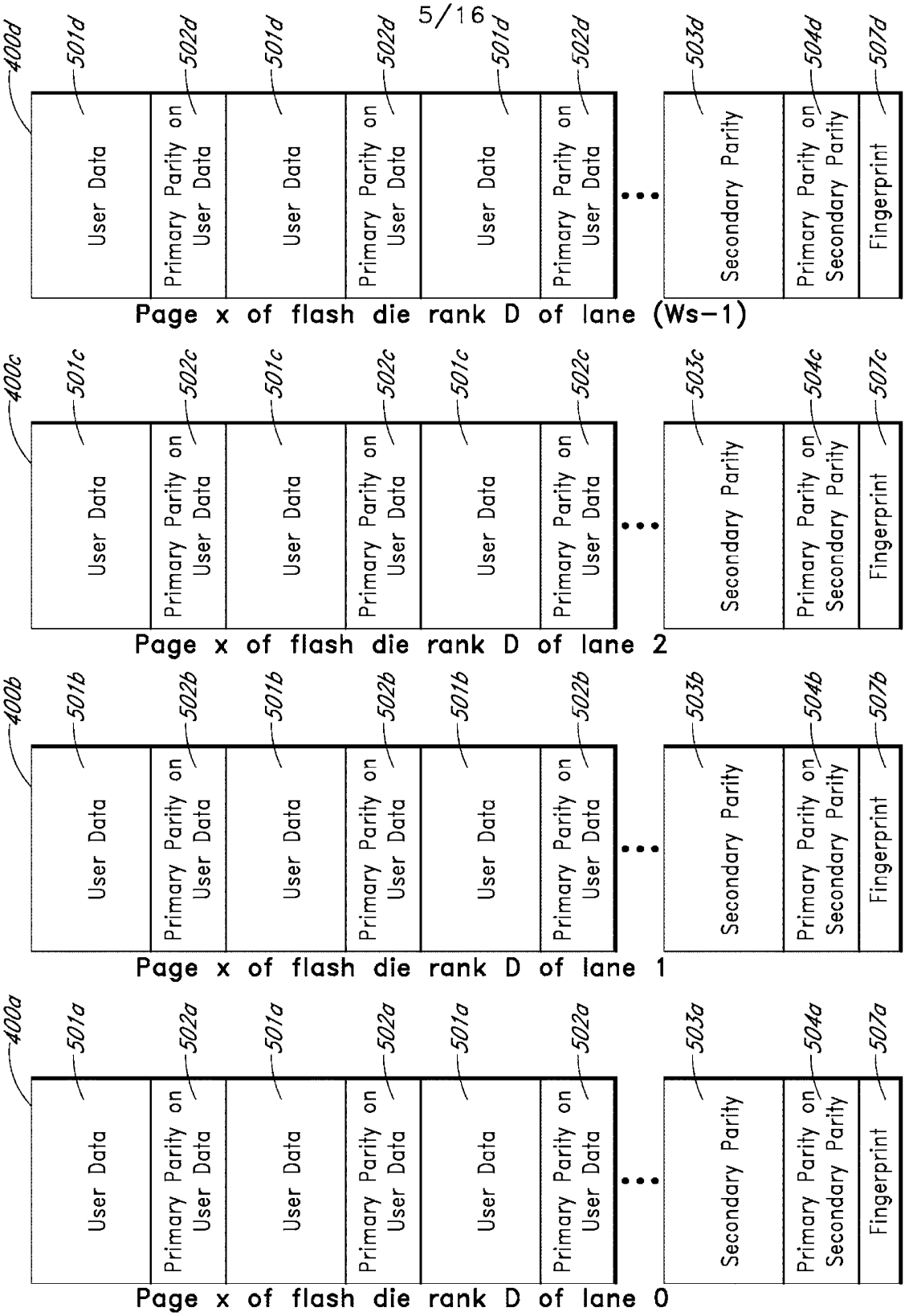

400d
501d
502d
501d
502d
501d
502d
503d
504d
507d
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
Secondary Parity
Primary Parity on Secondary Parity
Fingerprint
Page x of flash die rank D of lane (Ws−1)
550
400c
501c
502c
501c
502c
501c
502c
503c
504c
507c
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
Secondary Parity
Primary Parity on Secondary Parity
Fingerprint
Page x of flash die rank D of lane 2
400b
501b
502b
501b
502b
501b
502b
503b
504b
507b
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
Secondary Parity
Primary Parity on Secondary Parity
Fingerprint
Page x of flash die rank D of lane 1
400a
501a
502a
501a
502a
501a
502a
503a
504a
507a
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
User Data
Primary Parity on User Data
Secondary Parity
Primary Parity on Secondary Parity
Fingerprint
Page x of flash die rank D of lane 0

FIG. 5

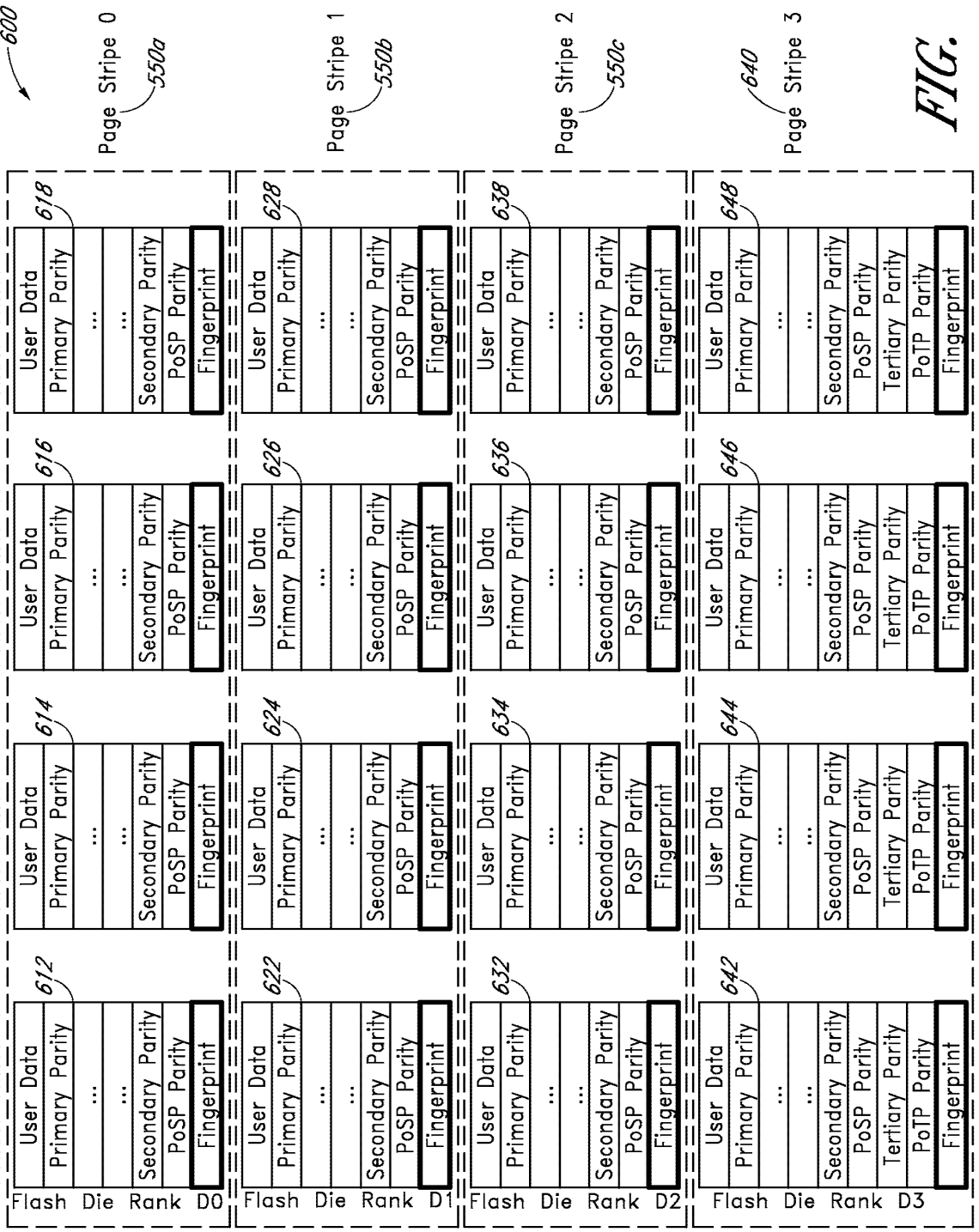

600
Flash Lane 0
Flash Lane 1
Flash Lane 2
Flash Lane 3
Page Stripe 0
550a
Page Stripe 1
550b
Page Stripe 2
550c
640
Page Stripe 3
User Data
Primary Parity
Secondary Parity
PoSP Parity
Tertiary Parity
PoTP Parity
Fingerprint
612
614
616
618
622
624
626
628
632
634
636
638
642
644
646
648
Flash Die Rank D0
Flash Die Rank D1
Flash Die Rank D2
Flash Die Rank D3

FIG. 6

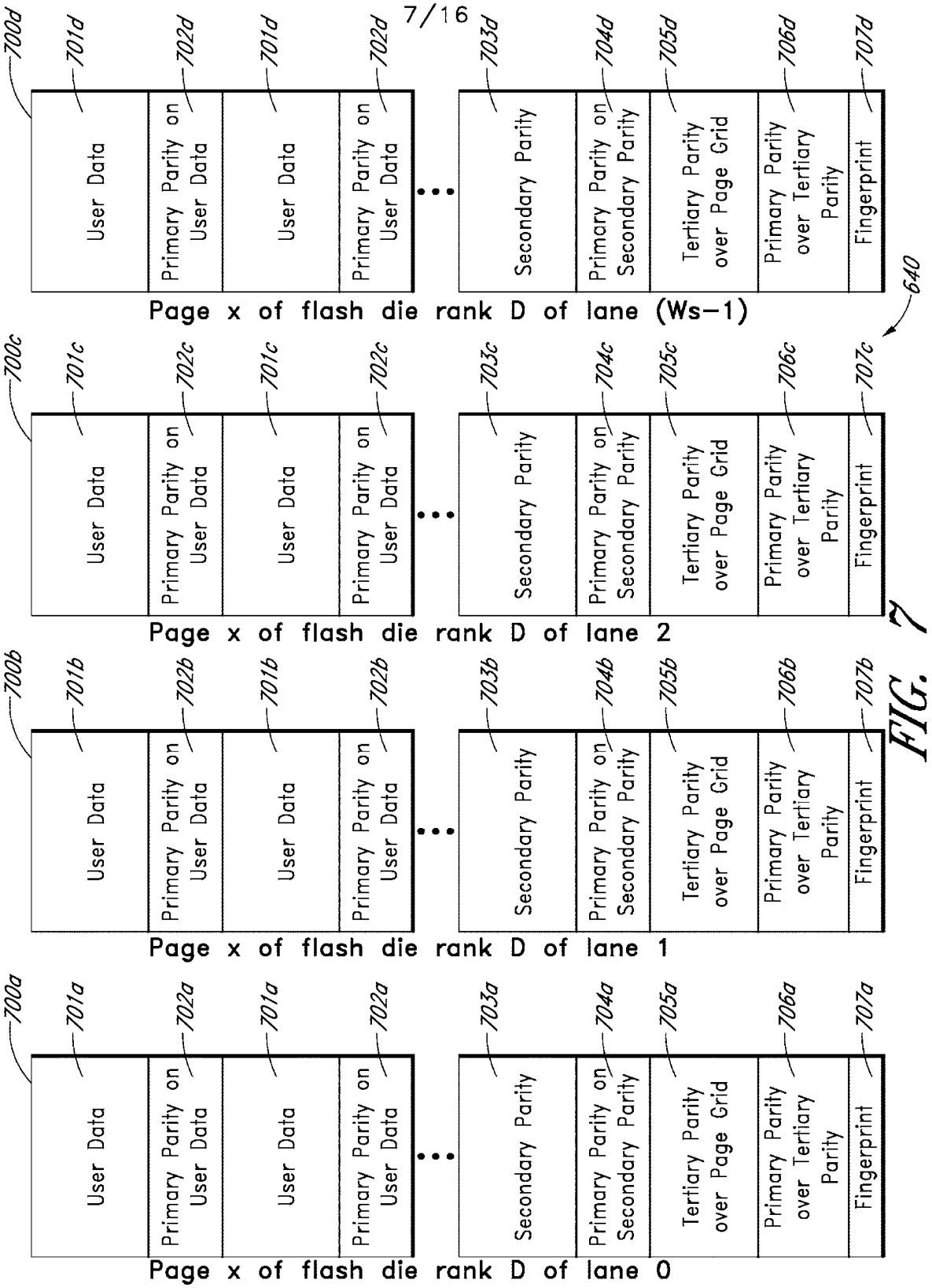

700d
701d
User Data
702d
Primary Parity on User Data
701d
User Data
702d
Primary Parity on User Data
703d
Secondary Parity
704d
Primary Parity on Secondary Parity
705d
Tertiary Parity over Page Grid
706d
Primary Parity over Tertiary Parity
707d
Fingerprint
Page x of flash die rank D of lane (Ws−1)
640
700c
701c
User Data
702c
Primary Parity on User Data
701c
User Data
702c
Primary Parity on User Data
703c
Secondary Parity
704c
Primary Parity on Secondary Parity
705c
Tertiary Parity over Page Grid
706c
Primary Parity over Tertiary Parity
707c
Fingerprint
Page x of flash die rank D of lane 2
7
700b
701b
User Data
702b
Primary Parity on User Data
701b
User Data
702b
Primary Parity on User Data
703b
Secondary Parity
704b
Primary Parity on Secondary Parity
705b
Tertiary Parity over Page Grid
706b
Primary Parity over Tertiary Parity
707b
Fingerprint
Page x of flash die rank D of lane 1
700a
701a
User Data
702a
Primary Parity on User Data
701a
User Data
702a
Primary Parity on User Data
703a
Secondary Parity
704a
Primary Parity on Secondary Parity
705a
Tertiary Parity over Page Grid
706a
Primary Parity over Tertiary Parity
707a
Fingerprint
Page x of flash die rank D of lane 0

FIG.

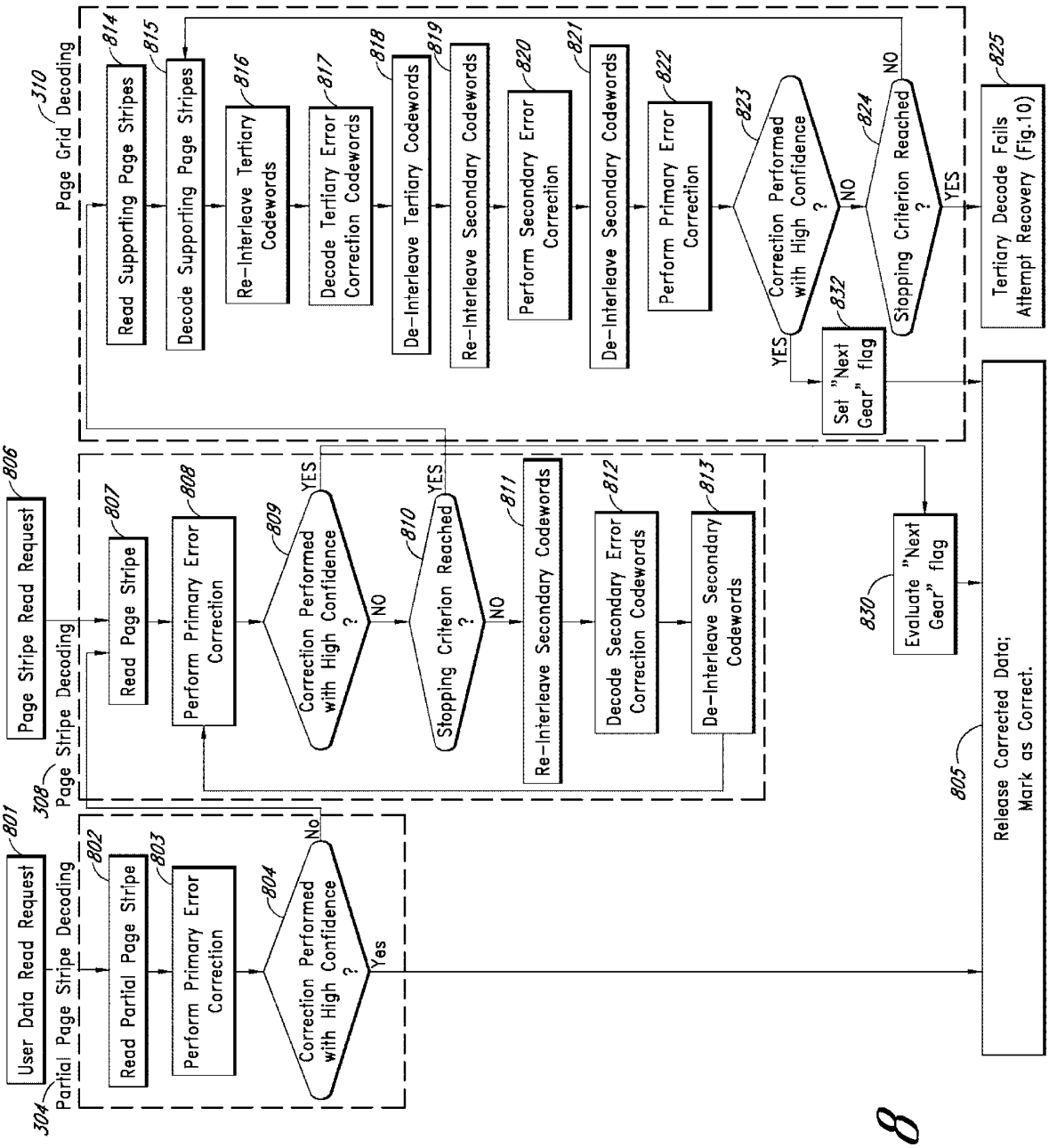

User Data Read Request
801
Partial Page Stripe Decoding
304
Read Partial Page Stripe
802
Perform Primary Error Correction
803
Correction Performed with High Confidence ?
804
No
Yes
Page Stripe Decoding
308
Page Stripe Read Request
806
Read Page Stripe
807
Perform Primary Error Correction
808
Correction Performed with High Confidence ?
809
YES
NO
Stopping Criterion Reached ?
810
YES
NO
Re-Interleave Secondary Codewords
811
Decode Secondary Error Correction Codewords
812
De-Interleave Secondary Codewords
813
Page Grid Decoding
310
Read Supporting Page Stripes
814
Decode Supporting Page Stripes
815
Re-Interleave Tertiary Codewords
816
Decode Tertiary Error Correction Codewords
817
De-Interleave Tertiary Codewords
818
Re-Interleave Secondary Codewords
819
Perform Secondary Error Correction
820
De-Interleave Secondary Codewords
821
Perform Primary Error Correction
822
Correction Performed with High Confidence ?
823
YES
NO
Stopping Criterion Reached ?
824
NO
YES
Tertiary Decode Fails Attempt Recovery (Fig.10)
825
Set "Next Gear" flag
832
Evaluate "Next Gear" flag
830
Release Corrected Data; Mark as Correct.
805

FIG. 8

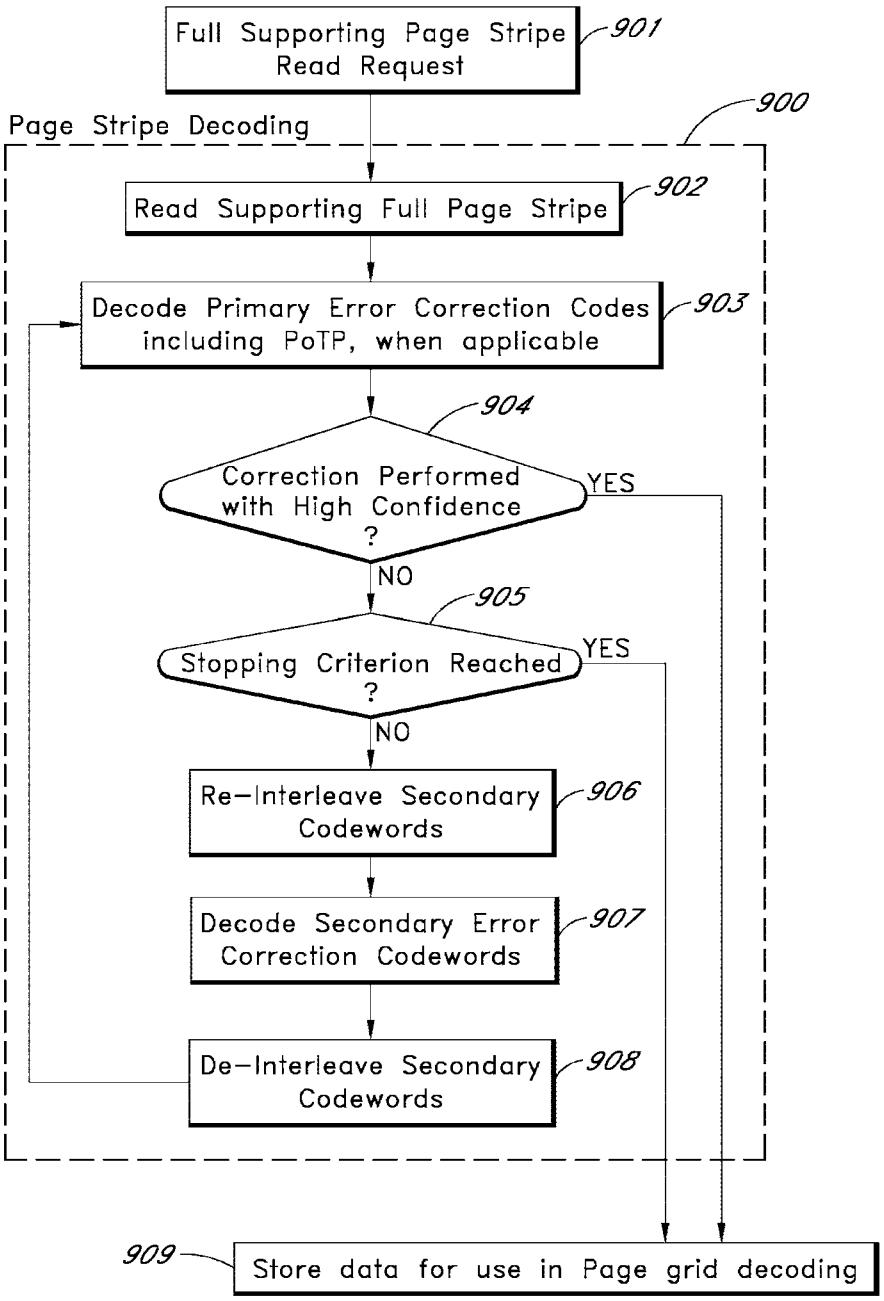

Full Supporting Page Stripe Read Request
901
Page Stripe Decoding
900
Read Supporting Full Page Stripe
902
Decode Primary Error Correction Codes including PoTP, when applicable
903
904
Correction Performed with High Confidence ?
YES
NO
905
Stopping Criterion Reached ?
YES
NO
Re-Interleave Secondary Codewords
906
Decode Secondary Error Correction Codewords
907
De-Interleave Secondary Codewords
908
909
Store data for use in Page grid decoding

FIG. 9

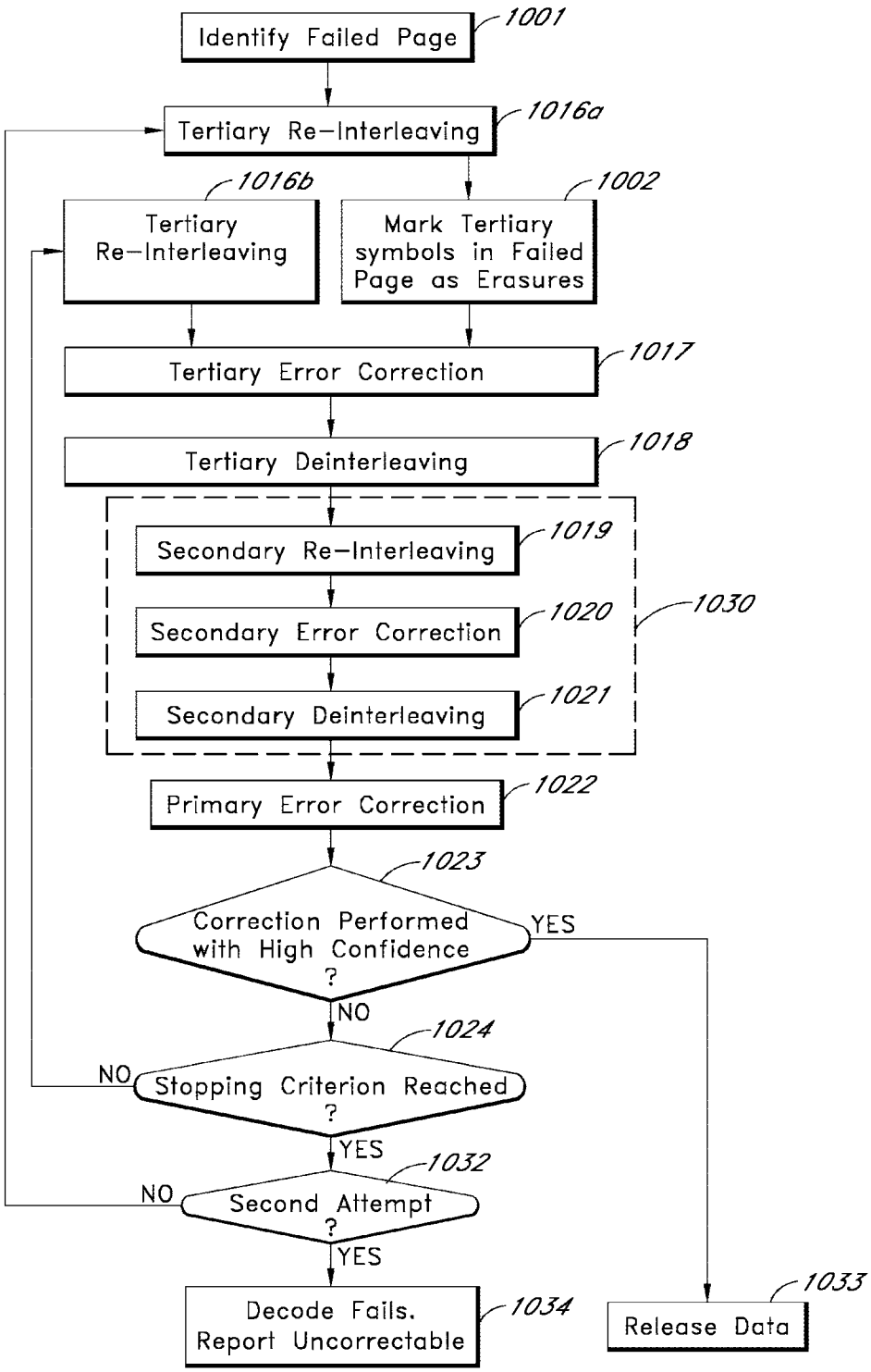

Identify Failed Page
1001
Tertiary Re-Interleaving
1016a
Tertiary Re-Interleaving
1016b
Mark Tertiary symbols in Failed Page as Erasures
1002
Tertiary Error Correction
1017
Tertiary Deinterleaving
1018
Secondary Re-Interleaving
1019
Secondary Error Correction
1020
Secondary Deinterleaving
1021
1030
Primary Error Correction
1022
Correction Performed with High Confidence ?
1023
YES
NO
Stopping Criterion Reached ?
1024
NO
YES
Second Attempt ?
1032
NO
YES
Decode Fails. Report Uncorrectable
1034
Release Data
1033

FIG. 10

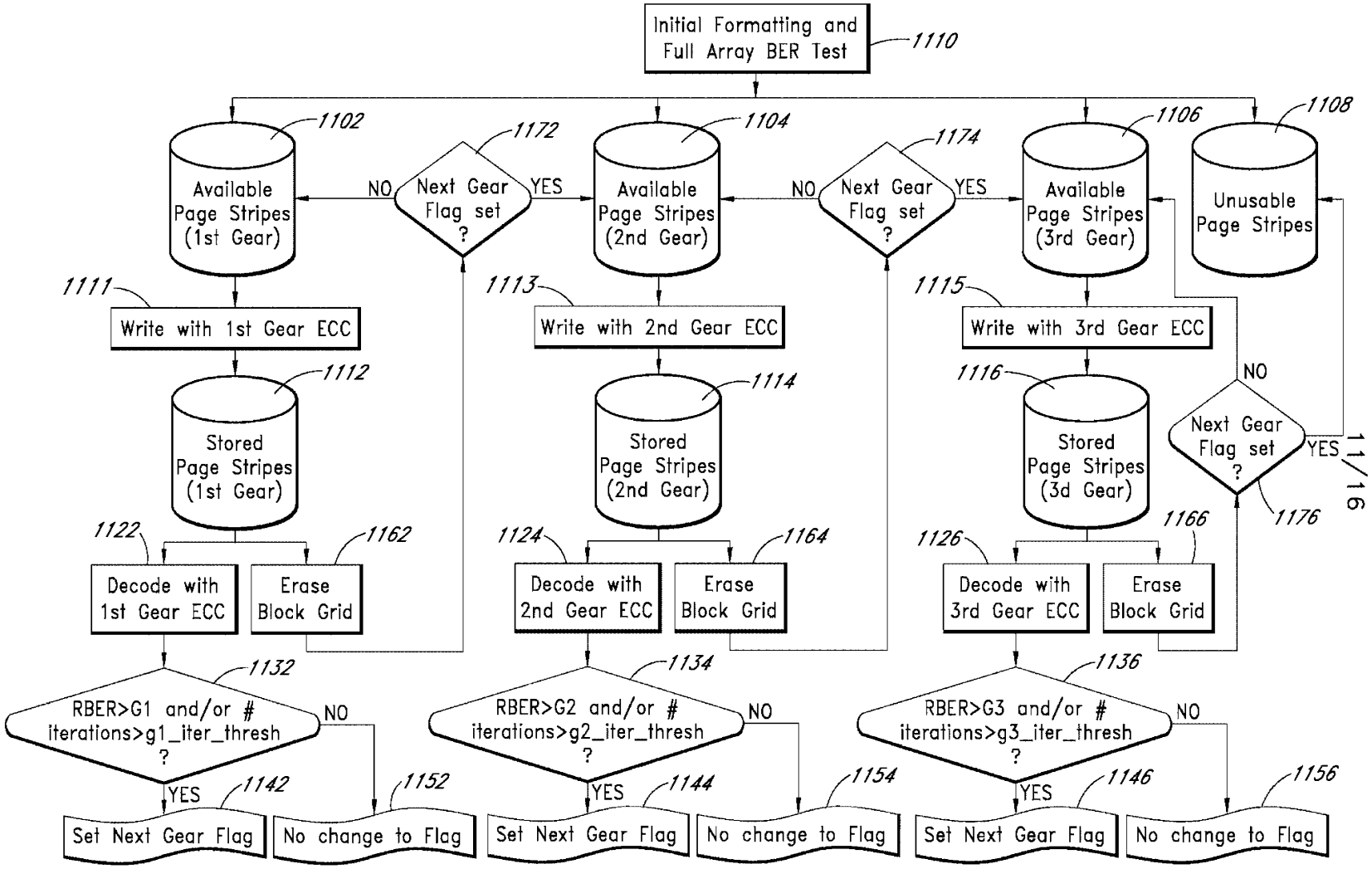


FIG. 11

Ex.1002 / Page 122 of 744
Sandisk Technologies, Inc. et al

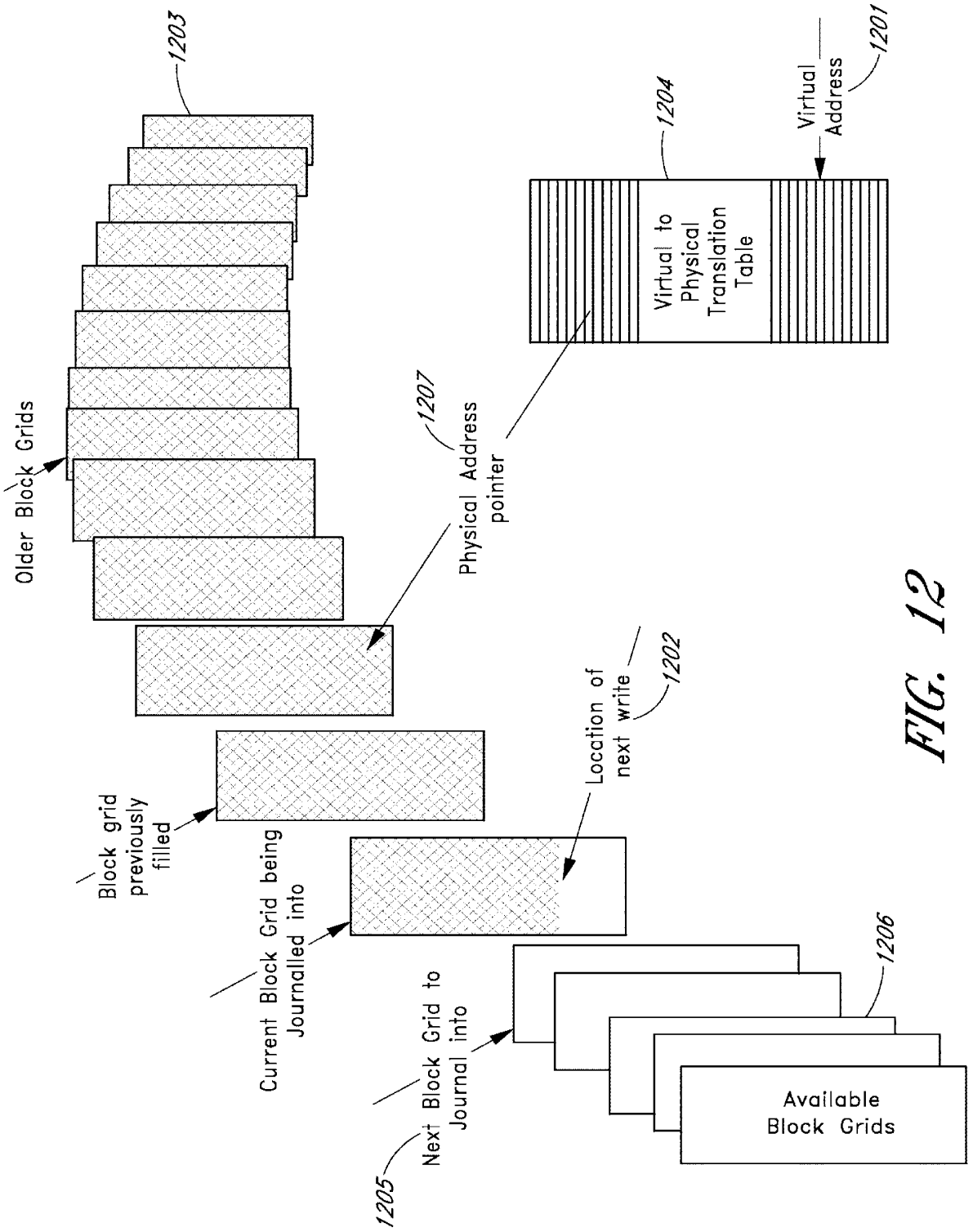

1203
Older Block Grids
1204
Virtual to Physical Translation Table
Virtual Address
1201
1207
Physical Address pointer
Location of next write
1202
Block grid previously filled
Current Block Grid being Journalled into
Next Block Grid to Journal into
1205
1206
Available Block Grids

FIG. 12

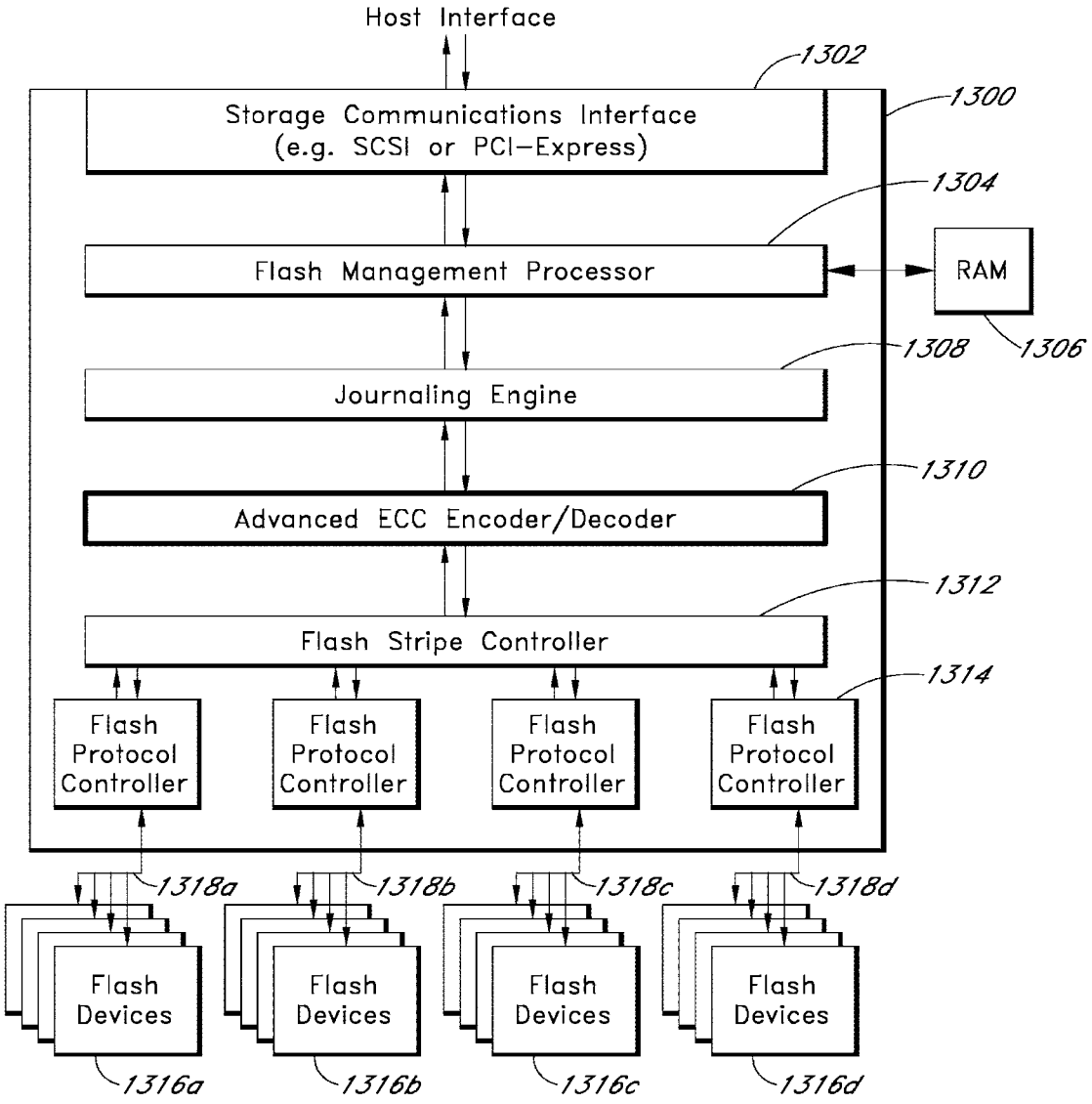

Host Interface
1302
1300
Storage Communications Interface
(e.g. SCSI or PCI–Express)
1304
Flash Management Processor
RAM
1306
1308
Journaling Engine
1310
Advanced ECC Encoder/Decoder
1312
Flash Stripe Controller
1314
Flash Protocol Controller
Flash Protocol Controller
Flash Protocol Controller
Flash Protocol Controller
1318a
1318b
1318c
1318d
Flash Devices
Flash Devices
Flash Devices
Flash Devices
1316a
1316b
1316c
1316d

FIG. 13

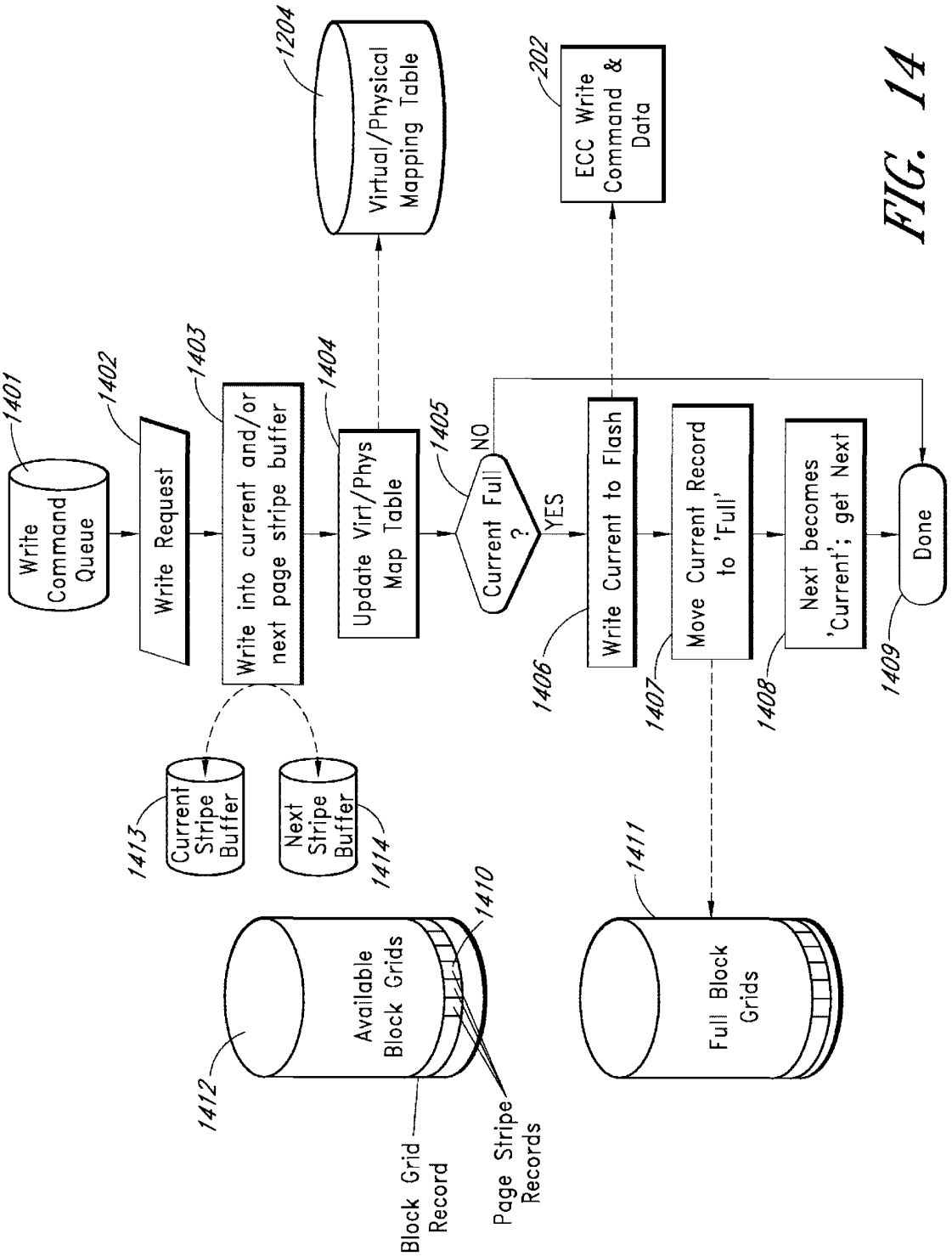

Write Command Queue
1401
Write Request
1402
Write into current and/or next page stripe buffer
1403
Update Virt/Phys Map Table
1404
Current Full ?
1405
NO
YES
Write Current to Flash
1406
Move Current Record to 'Full'
1407
Next becomes 'Current'; get Next
1408
Done
1409
Virtual/Physical Mapping Table
1204
ECC Write Command & Data
202
Current Stripe Buffer
1413
Next Stripe Buffer
1414
Available Block Grids
1412
1410
Full Block Grids
1411
Block Grid Record
Page Stripe Records

FIG. 14

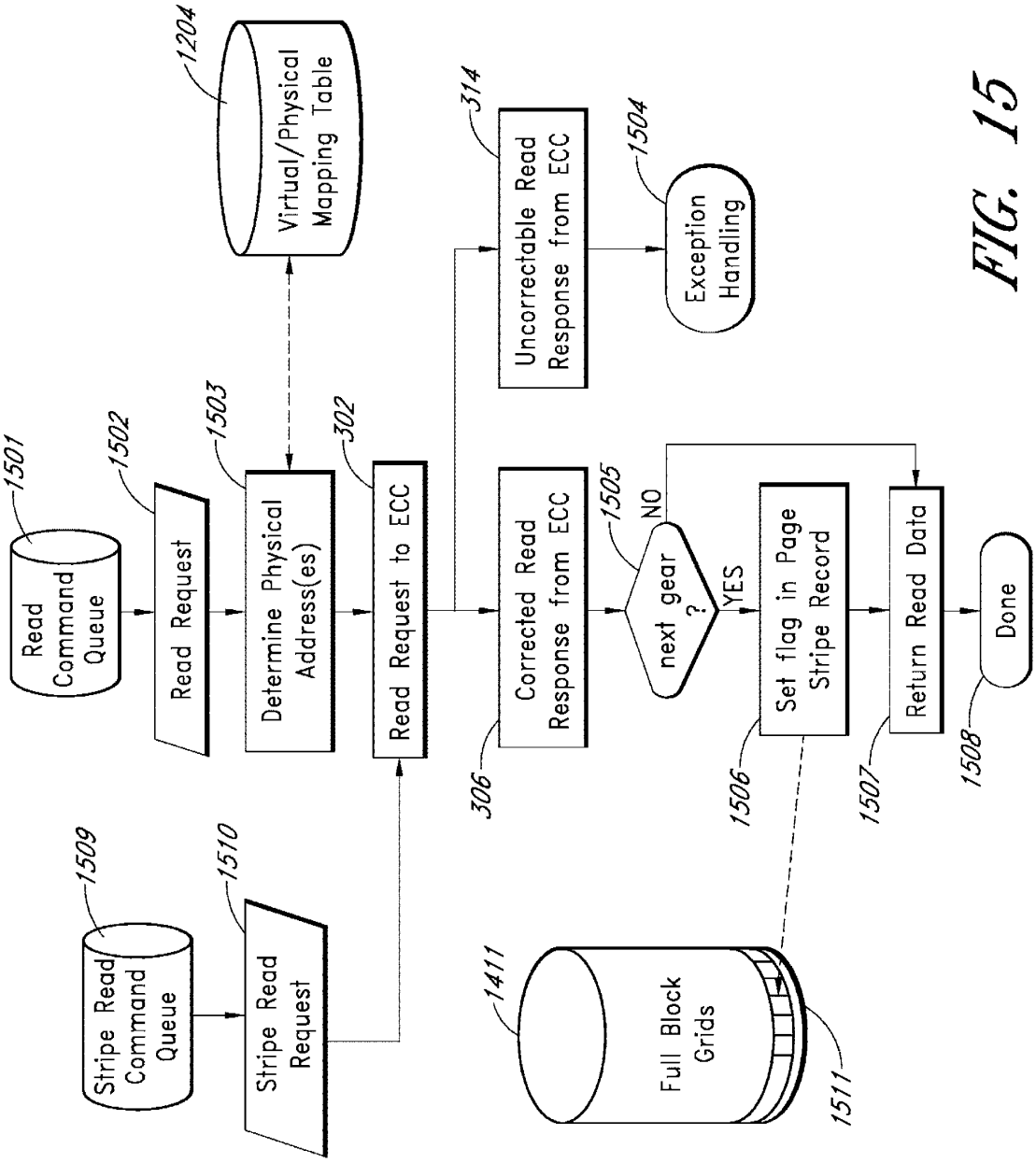
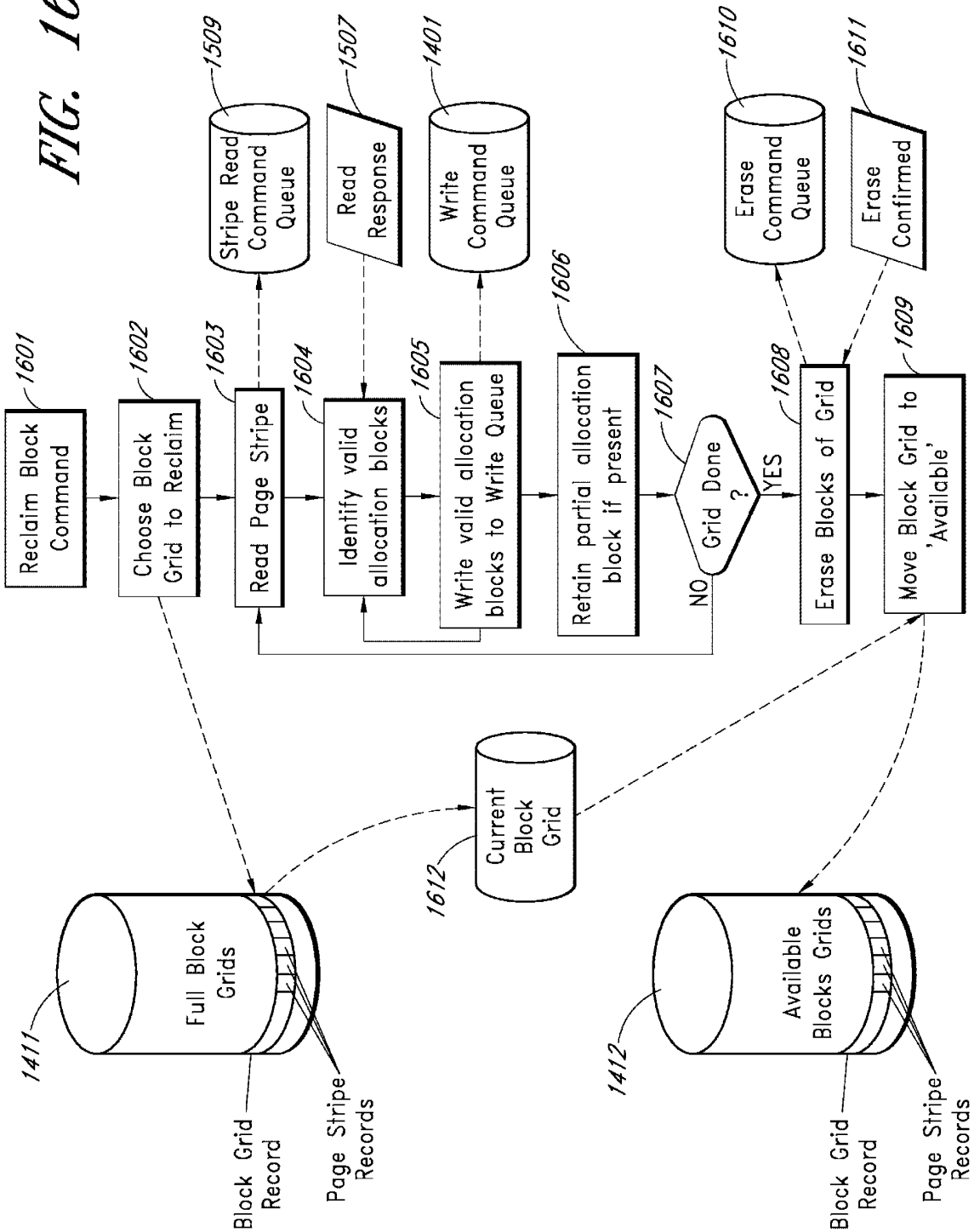

1204
Virtual/Physical Mapping Table
314
Uncorrectable Read Response from ECC
1504
Exception Handling
1501
Read Command Queue
1502
Read Request
1503
Determine Physical Address(es)
302
Read Request to ECC
306
Corrected Read Response from ECC
1505
next gear ?
NO
YES
1506
Set flag in Page Stripe Record
1507
Return Read Data
1508
Done
1509
Stripe Read Command Queue
1510
Stripe Read Request
1411
Full Block Grids
1511

FIG. 15

FIG. 16


1601
Reclaim Block Command
1602
Choose Block Grid to Reclaim
1603
Read Page Stripe
1604
Identify valid allocation blocks
1605
Write valid allocation blocks to Write Queue
1606
Retain partial allocation block if present
1607
Grid Done ?
NO
YES
1608
Erase Blocks of Grid
1609
Move Block Grid to 'Available'
1509
Stripe Read Command Queue
1507
Read Response
1401
Write Command Queue
1610
Erase Command Queue
1611
Erase Confirmed
1612
Current Block Grid
1411
Full Block Grids
Block Grid Record
Page Stripe Records
1412
Available Blocks Grids
Block Grid Record
Page Stripe Records

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-149592
(43)Date of publication of application : 30.05.2000

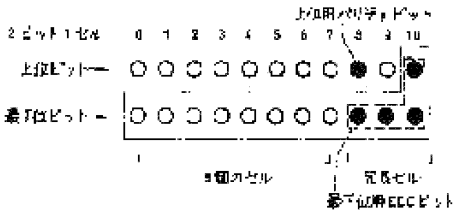
(51)Int.Cl. G11C 29/00
G11C 17/00
G11C 16/04
G11C 16/06
G11C 16/02

(21)Application number : 10-319696
(22)Date of filing : 11.11.1998

(71)Applicant : NEC IC MICROCOMPUT SYST LTD
(72)Inventor : NAKAMURA YOSHIHIRO

(54) MEMORY CELL FOR DEFECT RELIEF AND STORAGE DEVICE UTILIZING THE SAME

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a memory cell for defect relief capable of being composed of redundant cells of the less number even when an ECC(error detecting/ correcting circuit) is used.
SOLUTION: When an ECC is used in a multilevel memory, the least significant bit data (8 bits) in each cell includes, in redundant memory cells (4 bits), information indicating whether a defect exists or not as well as positional information of the defect, while more significant bit data (the bit data other than the least significant bit data) (8 bits) include, in a redundant memory cell (1 bit), only information indicating whether a defect exists or not. Since the more significant redundant memory cells only require 1 bit, the redundant memory cells of only 3 bits are required by storing, in the more significant redundant memory cells, information indicating whether a defect exists or not as well as positional information of the defect with respect to the least significant bit data.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2000－149592
（P2000－149592A）
(43)公開日　平成12年５月30日(2000.5.30)

(51)Int.Cl.[7]	識別記号	Ｆ Ｉ		テーマコード（参考）
Ｇ１１Ｃ　29/00	６３１	Ｃ１１Ｃ　29/00	６３１Ｚ	５Ｂ００３
17/00		17/00	Ｅ	５Ｂ０２５
16/04			３０５	５Ｌ１０６
16/06			６３９Ｃ	
16/02			６４１	

審査請求　有　請求項の数６　ＯＬ　（全 12 頁）

(21)出願番号　特願平10－319696

(22)出願日　平成10年11月11日(1998.11.11)

(71)出願人　000232036
日本電気アイシーマイコンシステム株式会社
神奈川県川崎市中原区小杉町１丁目403番53

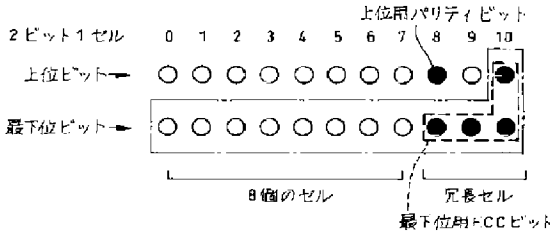
(72)発明者　中村　富浩
神奈川県川崎市中原区小杉町一丁目403番53　日本電気アイシーマイコンシステム株式会社内

(74)代理人　100088812
弁理士　▲柳▼川　信

Ｆターム(参考)　5B003 AB04 AC07 AD04 AE01
5B025 AA00 AD13 AE08
5L106 AA00 BB02 BB12 FF05

(54)【発明の名称】　不良救済用メモリセル及びそれを用いた記憶装置

(57)【要約】
【課題】　ＥＣＣを使用した場合でも、より少ない冗長セルで構成することができる不良救済用メモリセルを提供する。
【解決手段】　多値メモリにおいてはＥＣＣを使用すると、各セルの最下位のビットデータ（８ビット）には不良の有無／不良位置情報を冗長メモリセル（４ビット）を持たせ、上位のビットデータ（最下位のビットデータ以外のビットデータ）（８ビット）には不良の有無のみを冗長メモリセル（１ビット）に持たせている。その際、上位の冗長メモリセルは１ビットのみでよいため、上位の冗長メモリセルに最下位の不良の有無／不良位置情報を格納させることで、冗長メモリセルとしては３ビットで済む。



【特許請求の範囲】
【請求項１】　１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルであって、不良があるセルにおいて少なくとも予め設定された特定ビットに不良がある場合に救済範囲を限定するよう構成したことを特徴とする不良救済用メモリセル。
【請求項２】　前記特定ビットに対応する前記冗長セルに前記特定ビットの不良の有無及び不良位置情報を格納し、前記特定ビット以外のビットに対応する前記冗長セルに各々対応するビットの不良の有無だけの情報を格納するようにしたことを特徴とする請求項１記載の不良救済用メモリセル。
【請求項３】　前記特定ビットに対応する前記冗長セルは、前記特定ビット以外のビットに対応する前記冗長セルの未使用領域を含むことを特徴とする請求項１または請求項２記載の不良救済用メモリセル。
【請求項４】　前記特定ビットが最下位ビットであることを特徴とする請求項１から請求項３のいずれか記載の不良救済用メモリセル。
【請求項５】　１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルを用いる記憶装置であって、
前記不良救済用メモリセルを不良があるセルにおいて少なくとも予め設定された特定ビットに不良がある場合に救済範囲を限定するよう構成し、
前記不良救済用メモリセルから読出したデータを各メモリセル毎に前記特定ビットのデータ及び前記特定ビット以外のビットのデータにエンコードするエンコード手段と、
前記エンコード手段でエンコードされた前記特定ビット以外のビットのデータで作成したパリティビットと前記特定ビット以外のビットのパリティビットとを比較して前記特定ビット以外のビットのデータの不良の有無を判定するパリティ比較手段と、
前記エンコード手段でエンコードされた前記特定ビットのデータを基に前記特定ビットのデータの不良の有無と不良位置とを特定した特定ビット用訂正信号を生成する特定ビット誤り特定手段と、
前記パリティ比較手段の比較結果と前記特定ビット誤り特定手段の特定ビット用訂正信号とを基に前記特定ビット以外のビット用訂正信号を生成する特定ビット以外のビット誤り特定手段と、
前記エンコード手段でエンコードされた前記特定ビットのデータと前記特定ビット以外のビットのデータとのどちらかを選択して出力するデータ用切換手段と、
前記特定ビット誤り特定手段の特定ビット用訂正信号と前記特定ビット以外のビット誤り特定手段の前記特定ビット以外のビット用訂正信号とのどちらかを選択して出力する訂正用切換手段と、
前記訂正用切換手段からの訂正信号を基に前記データ用切換手段からのデータを補正する補正手段とを有することを特徴とする記憶装置。
【請求項６】　前記特定ビットに対応する前記冗長セルに前記特定ビットの不良の有無及び不良位置情報を格納し、前記特定ビット以外のビットに対応する前記冗長セルに各々対応するビットの不良の有無だけの情報を格納するようにしたことを特徴とする請求項５記載の記憶装置。
【請求項７】　前記特定ビットに対応する前記冗長セルは、前記特定ビット以外のビットに対応する前記冗長セルの未使用領域を含むことを特徴とする請求項５または請求項６記載の記憶装置。
【請求項８】　前記特定ビットが最下位ビットであることを特徴とする請求項５から請求項７のいずれか記載の記憶装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は不良救済用メモリセル及びそれを用いた記憶装置に関し、特に１個のメモリセルに複数のビットデータを記憶し、冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルに関する。
【０００２】
【従来の技術】従来、この種の不良救済用メモリセルにおいては、１個のメモリセルに複数のビットデータを記憶する多値メモリが用いられており、本来のデータを格納するセルとは別に設けた冗長セルに格納したデータを用いて、１ワード当り１ビットの不良ビットの訂正を行うＥＣＣ（Ｅｒｒｏｒ　Ｃｈｅｃｋｉｎｇ　ａｎｄ　Ｃｏｒｒｅｃｔｉｎｇ：エラー検出／訂正回路）を搭載している。
【０００３】この多値メモリにおいて、ＥＣＣを使用した場合、図１１に示すように、最下位ビット及び上位ビット（最下位ビット以外のビット）ともに不良の有無／不良位置情報を冗長メモリセルに持つ必要がある。
【０００４】例えば、８ビットのデータに対し、１ビットの不良を救済するのに必要なＥＣＣ用ビットは、図１１に示すように、不良の有無／不良位置を特定するために４ビットが必要となる。つまり、メモリセルのビット＃０～ビット＃７に、冗長メモリセルであるビット＃８～＃１１を加える必要がある。
【０００５】上記の多値メモリを用いた記憶装置を図１２に示す。この記憶装置では１ワードを８ビットとしており、多値メモリからなるメモリセル２１と、センスアンプ２２と、ラッチ・エンコード回路２３と、誤りビット特定回路２４と、補正回路２５とから構成されている。
【０００６】この記憶装置にワードの読出し指示が入力

されると、ワード線を３段階に変動した各段階での８セル＋ＥＣＣ冗長セルのオン／オフがセンスアンプ２２に読出され、その結果がラッチ・エンコード回路２３に入力される。

【０００７】ラッチ・エンコード回路２３においてはメモリセル２１から読出したデータを各メモリセル毎に最下位及び上位のビットデータにエンコードする。ラッチ・エンコード回路２３の出力の最下位及び上位のビットデータ（８ビット）＋最下位及び上位のＥＣＣビット（４ビット）は誤りビット特定回路２４に入力され、誤りビット特定回路２４は最下位及び上位のビットデータの不良の有無と不良位置とを特定した訂正信号を補正回路２５に出力する。この訂正信号は８本で、例えば、５ビット目に誤りがあれば、５ビット目の訂正信号に“Ｈ”、他のビットの訂正信号に“Ｌ”が夫々出力される。

【０００８】補正回路２５はラッチ・エンコード回路２３の出力の最下位及び上位のビットデータと誤りビット特定回路２４の訂正信号とを入力し、最下位及び上位のビットデータと訂正信号とを比較し、不良があれば補正して出力し、不良がなければそのまま出力する。

【０００９】

【発明が解決しようとする課題】上述した従来の多値メモリでは、１ワードを８ビットとするメモリ製品においてＥＣＣを使用した場合、不良救済用冗長メモリセルの数としてビットのほかに、４ビットの冗長セルが必要になるという問題がある。

【００１０】そこで、本発明の目的は上記の問題点を解消し、ＥＣＣを使用した場合でも、より少ない冗長セルで構成することができる不良救済用メモリセル及びそれを用いた記憶装置を提供することにある。

【００１１】

【課題を解決するための手段】本発明による不良救済用メモリセルは、１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルであって、不良があるセルにおいて少なくとも予め設定された特定ビットに不良がある場合に救済範囲を限定するよう構成している。

【００１２】本発明による記憶装置は、１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルを用いる記憶装置であって、前記不良救済用メモリセルを不良があるセルにおいて少なくとも予め設定された特定ビットに不良がある場合に救済範囲を限定するよう構成し、前記不良救済用メモリセルから読出したデータを各メモリセル毎に前記特定ビットのデータ及び前記特定ビット以外のビットのデータにエンコードするエンコード手段と、前記エンコード手段でエンコードされた前記特定ビット以外のビットのデータで作成したパリティビットと前記特定ビット以外のビットのパリティビットとを比較して前記特定ビット以外のビットのデータの不良の有無を判定するパリティ比較手段と、前記エンコード手段でエンコードされた前記特定ビットのデータを基に前記特定ビットのデータの不良の有無と不良位置とを特定した特定ビット用訂正信号を生成する特定ビット誤り特定手段と、前記パリティ比較手段の比較結果と前記特定ビット誤り特定手段の特定ビット用訂正信号とを基に前記特定ビット以外のビット用訂正信号を生成する特定ビット以外のビット誤り特定手段と、前記エンコード手段でエンコードされた前記特定ビットのデータと前記特定ビット以外のビットのデータとのどちらかを選択して出力するデータ用切換手段と、前記特定ビット誤り特定手段の特定ビット用訂正信号と前記特定ビット以外のビット誤り特定手段の前記特定ビット以外のビット用訂正信号とのどちらかを選択して出力する訂正用切換手段と、前記訂正用切換手段からの訂正信号を基に前記データ用切換手段からのデータを補正する補正手段とを備えている。

【００１３】すなわち、本発明の記憶装置は、多値メモリで、冗長セルに格納したデータを使って１ワードあたり１ビットの不良ビットの訂正を行うＥＣＣを搭載するメモリ製品を用いる装置において、不良があるセルにおいて少なくとも最下位ビットに不良がある場合に救済範囲を限定することで、最下位ビットに対して不良の有無及び不良位置情報を冗長セルに持たせ、最下位以外のビットに対して不良の有無だけの情報を冗長セルに持たせるようにしている。

【００１４】これによって、冗長セル数を減らすことが可能となる。例えば、８ビットのデータに対して１ビットの不良を救済するのに必要なＥＣＣ用ビットは不良の有無／不良位置を特定するために４ビットを必要とするが、不良の有無のみを特定する場合には１ビットで良い。

【００１５】救済範囲を上記のように限定するのは、セルのしきい値電圧Ｖｔを３段階以上に設定できることによって多値を実現する多値メモリ［例えば、マスクＲＯＭ（リードオンリメモリ）のイオン注入量、フラッシュ（Ｆｌａｓｈ）の書込み量によって実現］の場合、セルに設定できるしきい値電圧Ｖｔ各々と出力ビット各々との対応を、隣合うしきい値電圧Ｖｔが必ず最下位ビットが反転するように設定することで、しきい値電圧Ｖｔのバラツキによる不良があるセルの場合に少なくとも最下位ビットが反転するので、この不良が多い場合には本発明による冗長セル削減によるメリットが、救済範囲限定による救済率ダウンのデメリットを上回るからである。

【００１６】上記説明で最下位ビットとしているのは１つの例であり、最下位ビット以外の特定ビットに置き換えても同様の考え方を適用することで、同様の効果を発揮する回路が実現可能である。

【００１７】
【発明の実施の形態】次に、本発明の一実施例について図面を参照して説明する。図１は本発明の一実施例による不良救済用メモリセル（多値メモリ）の構成例を示す図である。図においては、２ビット１セルの多値メモリを使用した場合を示している。
【００１８】この場合、多値メモリにおいてはＥＣＣを使用すると、各セルの最下位のビットデータ（８ビット）には不良の有無／不良位置情報を冗長メモリセル（４ビット）を持たせ、上位のビットデータ（最下位のビットデータ以外のビットデータ）（８ビット）には不良の有無のみを冗長メモリセル（１ビット）に持たせている。
【００１９】その際、上位の冗長メモリセルは１ビットのみでよいため、上位の冗長メモリセルに最下位の不良の有無／不良位置情報を格納させることで、冗長メモリセルとしては３ビットで済む。つまり、メモリセルのビット＃０～ビット＃７に、冗長メモリセルであるビット＃８～＃１０を加えるだけで良いことになる。
【００２０】図２は本発明の一実施例による不良救済用メモリセル（多値メモリ）の他の構成例を示す図である。図においては、４ビット１セルの多値メモリを使用した場合を示している。
【００２１】この場合、多値メモリにおいてはＥＣＣを使用すると、各セルの最下位のビットデータ（８ビット）には不良の有無／不良位置情報を冗長メモリセル（４ビット）を持たせ、上位のビットデータ（最下位のビットデータ以外のビットデータ）（８ビット）には不良の有無のみを冗長メモリセル（１ビット）に持たせている。
【００２２】その際、上位の冗長メモリセルは１ビットのみでよいため、上位の冗長メモリセルに最下位の不良の有無／不良位置情報を格納させることで、冗長メモリセルとしては２ビットで済む。つまり、メモリセルのビット＃０～ビット＃７に、冗長メモリセルであるビット＃８，＃９を加えるだけで良いことになる。
【００２３】このように、未使用の上位ビットのＥＣＣ冗長セルを最下位ＥＣＣビットの格納セルに使用することによって、１ワードを８ビットとするメモリ製品において、２ビット１セルの多値メモリでは冗長セルを３ビット、４ビット１セル以上の多値メモリでは２ビットの冗長セルの使用で済ませることができる。
【００２４】図３は本発明の一実施例による多値メモリを使用した記憶装置の構成を示すブロック図である。図においては、２ビット１セルの多値メモリを使用した場合の構成を示している。
【００２５】この図において、本発明の一実施例による記憶装置は、多値メモリからなるメモリセル１と、センスアンプ２と、ラッチ・エンコード回路３と、パリティ比較回路４と、最下位誤りビット特定回路５と、上位誤りビット特定回路６と、データ用切換え回路７と、訂正用切換え回路８と、補正回路９とから構成されている。
【００２６】メモリセル１は図１に示す構成の不良救済用メモリセルから構成され、ワードの読出し指示が入力されると、ワード線を３段階に変動した各段階での８セル＋ＥＣＣ冗長セルのオン／オフをセンスアンプ２に読出す。
【００２７】ラッチ・エンコード回路３はメモリセル１から読出したデータを各メモリセル毎に最下位及び上位のビットデータにエンコードし、上位８ビットデータをパリティ比較回路４及びデータ用切換え回路７に出力し、下位８ビットデータを最下位誤りビット特定回路５及びデータ用切換え回路７に出力する。また、ラッチ・エンコード回路３は上位用パリティビットをパリティ比較回路４に出力し、最下位用ＥＣＣビットを最下位誤りビット特定回路５に出力する。
【００２８】パリティ比較回路４は上位８ビットデータで作成したパリティビットと上位用パリティビットとを比較することによって上位ビットデータの不良の有無を判定し、その判定結果である上位ビット誤り判定信号を上位誤りビット特定回路６に出力する。パリティ比較回路４は上位８ビットデータと上位用パリティビットとの比較を排他的論和（ＥＸＯＲ）をとることによって実施する。
【００２９】最下位誤りビット特定回路５はラッチ・エンコード回路３の出力の最下位８ビットデータ｜最下位ＥＣＣビット（４ビット）を入力し、最下位ビットデータの不良の有無と不良位置とを特定した最下位用訂正信号を上位誤りビット特定回路６及び訂正用切換え回路８に出力する。
【００３０】上位誤りビット特定回路６はパリティ比較回路４からの上位ビット誤り判定信号と最下位誤りビット特定回路５からの最下位用訂正信号とを入力し、パリティ比較回路４からの上位ビット誤り判定信号で上位ビットに誤り有無の判定を行い、誤り有りの場合に最下位誤りビット特定回路５からの最下位用訂正信号を上位用訂正信号として訂正用切換え回路８に出力し、誤り無しの場合に全て“Ｌ”の信号を訂正用切換え回路８に出力する。
【００３１】データ用切換え回路７はラッチ・エンコード回路３でエンコードされた最下位８ビットデータと上位８ビットデータとのどちらかを最下位／上位切換えアドレス信号に応じて補正回路９に出力する。
【００３２】訂正用切換え回路８は最下位誤りビット特定回路５からの最下位用訂正信号と上位誤りビット特定回路６からの上位用訂正信号とを最下位／上位切換えアドレス信号に応じて切換えて補正回路９に出力する。
【００３３】補正回路９は訂正用切換え回路８からの訂正信号を基にデータ用切換え回路７からのデータを補正する。すなわち、補正回路９は不良があれば最下位８ビ

ットデータを最下位用訂正信号で、上位８ビットデータを上位用訂正信号で夫々補正して出力し、不良がなければ最下位８ビットデータ及び上位８ビットデータをそのまま出力する。

【００３４】図４は図３の上位誤りビット特定回路６の構成を示す回路図である。図において、上位誤りビット特定回路６は最下位誤りビット特定回路５からの最下位用訂正信号とパリティ比較回路４からの上位ビット誤り判定信号とのナンドをとるナンド回路１１－１～１１－８と、ナンド回路１１－１～１１－８の出力を反転するインバータ回路１２－１～１２－８とから構成されている。

【００３５】図５は図３の記憶装置の動作を示すタイミングチャートであり、図６は本発明の一実施例による多値メモリのワード線の動作を示す図であり、図７は図６に示すワード線の動作時のセンスアンプ２及びラッチ・エンコード回路３の出力結果を示す図である。

【００３６】これら図１～図７を参照して本発明の一実施例による記憶装置の動作、つまりメモリセル１からの読出し動作について説明する。ここではメモリセル（多値メモリのセル）１からデータを読出す動作を、メモリセル１が２ビット１セルの場合について説明する。尚、メモリセル１のしきい値電圧をＶｔとして、４段階の電位（Ｖｔ０～Ｖｔ３とする）のうち、任意の電位を設定することで１セル当り４値を実現しているものとする。

【００３７】１セルにおいてワード線の電位は、図６に示すように、３段階（ＶＲ１，ＶＲ２，ＶＲ３）に変動し、各段階で読出しを行う。この時、セルのしきい値電圧Ｖｔとセンスアンプ２及びラッチ・エンコード回路３の出力結果は図７に示すようになる。ここで、Ｄｍ００は下位ビット、Ｄｍ０１は上位ビットを示している。

【００３８】すなわち、ワード電位がＶＲ１の時の出力はセルの電位Ｖｔ０であれば“１”、セルの電位Ｖｔ１であれば“０”、セルの電位Ｖｔ２であれば“０”、セルの電位Ｖｔ３であれば“０”となる。

【００３９】ワード電位がＶＲ２の時の出力はセルの電位Ｖｔ０であれば“１”、セルの電位Ｖｔ１であれば“１”、セルの電位Ｖｔ２であれば“０”、セルの電位Ｖｔ３であれば“０”となる。

【００４０】ワード電位がＶＲ３の時の出力はセルの電位Ｖｔ０であれば“１”、セルの電位Ｖｔ１であれば“１”、セルの電位Ｖｔ２であれば“１”、セルの電位Ｖｔ３であれば“０”となる。

【００４１】尚、上位ビットＤｍ０１はセルの電位Ｖｔ０であれば“０”（“１”）、セルの電位Ｖｔ１であれば“０”（“１”）、セルの電位Ｖｔ２であれば“１”（“０”）、セルの電位Ｖｔ３であれば“１”（“０”）となる。

【００４２】また、下位ビットＤｍ００はセルの電位Ｖｔ０であれば“０”（“１”）、セルの電位Ｖｔ１であれば“１”（“０”）、セルの電位Ｖｔ２であれば“０”（“１”）、セルの電位Ｖｔ３であれば“１”（“０”）となる。

【００４３】上記の例の場合、例えばセルの電位Ｖｔ１が不良の時に、セルの閾値電圧ＶｔはＶｔ０又はＶｔ２となり、最下位ビットが反転する。

【００４４】本発明の一実施例による記憶装置が１ワードを８ビットとするメモリ製品の場合について、この記憶装置からのデータの読出しについて以下説明する。

【００４５】上記の記憶装置にワードの読出し指示が入力されると、ワード線を３段階に変動した各段階での８セル＋ＥＣＣ冗長セルのオン／オフがセンスアンプ２に読出され、その結果がラッチ・エンコード回路３に入力される。

【００４６】ラッチ・エンコード回路３においてはメモリセル１から読出したデータを各メモリセル毎に最下位と上位のビットデータにエンコードする。ラッチ・エンコード回路３の出力の最下位８ビット＋最下位ＥＣＣビット（４ビット）は最下位用誤りビット特定回路５に入力される。最下位用誤りビット特定回路５は最下位ビットの不良の有無と不良位置とを特定した訂正信号を出力する。この訂正信号は８本で、例えば、５ビット目に誤りがあれば、５ビット目の訂正信号に“Ｈ”、他のビットの訂正信号に“Ｌ”が夫々出力される。

【００４７】また、ラッチ・エンコード回路３の出力の上位８ビットと上位用パリティビットとはパリティ比較回路４に入力される。パリティ比較回路４は上位８ビットのデータで作成したパリティビットと上位用パリティビットとを比較することによって上位ビットの不良の有無の判定を実施する。パリティ比較回路４は上位８ビットと上位用パリティビットとの比較を排他的論理和（ＥＸＯＲ）をとることで実施する。

【００４８】上位誤りビット特定回路６は最下位用誤りビット特定回路５からの最下位用訂正信号とパリティ比較回路４からの上位ビット誤り判定信号とを入力し、上位ビット誤り判定信号で上位ビットの誤りの有無を判定し、誤り有りの場合に最下位用誤りビット特定回路５からの最下位用訂正信号を上位用訂正信号として出力し、誤り無しの場合に全て“Ｌ”の信号を出力する（図４参照）。

【００４９】訂正用切換え回路８は最下位用誤りビット特定回路５からの最下位用訂正信号と上位誤りビット特定回路６からの上位用訂正信号とを入力し、これらの信号を最下位／上位切換えアドレス信号によって切換えて出力する。

【００５０】データ用切換え回路７はラッチ・エンコード回路３でエンコードされた最下位８ビットデータ及び上位８ビットデータを入力し、最下位／上位切換えアドレス信号によって最下位８ビットデータ及び上位８ビットデータのどちらかを出力する。

【００５１】補正回路９は訂正用切換え回路８からの訂正信号とデータ用切換え回路７からのデータとを入力し、不良があれば最下位８ビットデータを最下位用訂正信号で、上位８ビットデータを上位用訂正信号で夫々補正して出力し、不良がなければ最下位８ビットデータ及び上位８ビットデータをそのまま出力する。

【００５２】具体的には、各セルのビットデータ毎に対応する訂正信号が“Ｈ”であれば、誤りがあるということで、ビットデータを反転して出力し、対応する訂正信号が“Ｌ”であれば正しいということで、そのままビットデータを出力する。つまり各セル毎にビットデータと対応する訂正信号との排他的論理和の結果を出力する。

【００５３】このように、未使用の上位ビットのＥＣＣ冗長セルを最下位ＥＣＣビットの格納セルに使用することによって、１ワードを８ビットとするメモリ製品において、２ビット１セルの多値メモリでは冗長セルを３ビット、４ビット１セル以上の多値メモリでは２ビットの冗長セルの使用で済ませることができる（図１及び図２参照）。

【００５４】図８（ａ）～（ｃ）は本発明の一実施例による２ビット１セルにおける本発明の救済率と従来の救済率との関係を示す図であり、図９（ａ）～（ｃ）は本発明の一実施例による４ビット１セルにおける本発明の救済率と従来の救済率との関係を示す図である。

【００５５】ここで、ｂ：従来の救済率、ｃ：本発明の救済率、ｄ：従来の使用ビット数、ｅ：本発明の使用ビット数、Ｆ：本発明の救済率／従来の救済率、ｇ：全不良率、ｈ：従来のウェハ上のチップ数とした場合、

有効チップ数－全体チップ数－（不良数－救済個数）
＝全体チップ数－（不良数―不良数＊救済率）
＝全体チップ数（１－（不良率－不良率＊本発明の救済率））
＝（ｄ／ｅ）＊ｈ＊（１－ｇ＊（１－（ｃ／ｂ）＊ｂ））
＝（ｄ／ｅ）＊ｈ＊（１－ｇ＊（１－Ｆ＊ｂ））

という式が成り立つ。図８及び図９は夫々上記の式から求められたグラフを示している。

【００５６】図８（ａ）において、Ａ１は本発明のｇ－２０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ２は本発明のｇ＝２０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ３は本発明のｇ＝２０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している。

【００５７】また、Ｂ１は従来のｇ＝２０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数９２個）、Ｂ２は従来のｇ＝２０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数９５個）、Ｂ３は従来のｇ＝２０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している（有効チップ数９８個）。

【００５８】図８（ｂ）において、Ａ４は本発明のｇ＝５０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ５は本発明のｇ＝５０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ６は本発明のｇ＝５０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している。

【００５９】また、Ｂ４は従来のｇ＝５０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数８０個）、Ｂ５は従来のｇ＝５０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数８７．５個）、Ｂ６は従来のｇ＝５０％、ｂ－９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している（有効チップ数９５個）。

【００６０】図８（ｃ）において、Ａ７は本発明のｇ＝８０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ８は本発明のｇ＝８０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ９は本発明のｇ＝８０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している。

【００６１】また、Ｂ７は従来のｇ＝８０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数６８個）、Ｂ８は従来のｇ＝８０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数８０個）、Ｂ９は従来のｇ＝８０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している（有効チップ数９２個）。

【００６２】図９（ａ）において、Ａ１１は本発明のｇ＝２０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ１２は本発明のｇ＝２０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ１３は本発明のｇ＝２０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している。

【００６３】また、Ｂ１１は従来のｇ＝２０％、ｂ＝６

０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数９２個）、Ｂ１２は従来のｇ＝２０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数９５個）、Ｂ１３は従来のｇ－２０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している（有効チップ数９８個）。

【００６４】図９（ｂ）において、Ａ１４は本発明のｇ＝５０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ１５は本発明のｇ＝５０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ１６は本発明のｇ－５０％、ｂ－９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している。

【００６５】また、Ｂ１４は従来のｇ＝５０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数８０個）、Ｂ１５は従来のｇ＝５０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数８７．５個）、Ｂ１６は従来のｇ－５０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している（有効チップ数９５個）。

【００６６】図９（ｃ）において、Ａ１７は本発明のｇ＝８０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ１８は本発明のｇ＝８０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し、Ａ１９は本発明のｇ－８０％、ｂ－９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している。

【００６７】また、Ｂ１７は従来のｇ＝８０％、ｂ＝６０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数６８個）、Ｂ１８は従来のｇ＝８０％、ｂ＝７５％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示し（有効チップ数８０個）、Ｂ１９は従来のｇ＝８０％、ｂ＝９０％の時の（本発明の救済率／と従来の救済率）と有効チップ数との関係を示している（有効チップ数９２個）。

【００６８】よって、２ビット１セルでは、図８に示すように、ｇ：全不良率が５０％以下では、本発明の救済率／従来の救済率が０．８近くで効果がある。また、４ビット１セルでは、図９に示すように、ｇ：全不良率が８０％以下でも、本発明の救済率／従来の救済率が０．８近くで効果がある。

【００６９】図１０は本発明の一実施例による不良救済用メモリセル（多値メモリ）の別の構成例を示す図である。図においては、１ワードを８ビットとするメモリ製品（２ビット１セル）を横に、８ビットづつ配置したものである。但し、この構成例の機能は１ワードを８ビットとするメモリ製品であり、１ワードを１６ビットとするメモリ製品ではない。

【００７０】この場合、従来の技術では８個の冗長セルが必要であるが、上位ビット（最下位以外のビット）に不良がある場合、本発明の一実施例による別の不良救済用メモリセルでは不良の有無のみの情報を冗長セルに記憶させることによって、５個の冗長セルで対応することができる。

【００７１】このように、多値メモリで、冗長セルに格納したデータを使って１ワードあたり１ビットの不良ビットの訂正を行うＥＣＣを搭載するメモリ製品を用いる装置で、不良があるセルにおいて少なくとも最下位ビットに不良がある場合に救済範囲を限定することで、最下位ビットに対して不良の有無及び不良位置情報を冗長セルに持たせ、最下位以外のビットに対して不良の有無だけの情報を冗長セルに持たせることで、冗長セル数を減らすことができる。

【００７２】例えば、８ビットのデータに対して１ビットの不良を救済するのに必要なＥＣＣ用ビットは不良の有無／不良位置を特定するために４ビットを必要とするが、不良の有無のみを特定する場合には１ビットで良い。

【００７３】救済範囲を上記のように限定するのは、セルのしきい値電圧Ｖｔを３段階以上に設定できることによって多値を実現する多値メモリ［例えば、マスクＲＯＭ（リードオンリメモリ）のイオン注入量、フラッシュ（Ｆｌａｓｈ）の書込み量によって実現］の場合、セルに設定できるしきい値電圧Ｖｔ各々と出力ビット各々との対応を、隣合うしきい値電圧Ｖｔが必ず最下位ビットが反転するように設定することで、しきい値電圧Ｖｔのバラツキによる不良があるセルの場合に少なくとも最下位ビットが反転するので、この不良が多い場合には本発明による冗長セル削減によるメリットが、救済範囲限定による救済率ダウンのデメリットを上回るからである。

【００７４】尚、上記説明で最下位ビットとしているのは１つの例であり、最下位ビット以外の特定ビットに置き換えても同様の考え方を適用することで、同様の効果を発揮する回路を実現することができる。

【００７５】

【発明の効果】以上説明したように本発明によれば、１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルにおいて、不良があるセルにおいて少なくとも予め設定された特定ビットに不良がある場合に救済範囲を限定することによって、ＥＣＣを使用した場合でも、より少ない冗長セルで構成することができるという効果がある。

【図面の簡単な説明】

【図１】本発明の一実施例による不良救済用メモリセルの構成例を示す図である。

【図２】本発明の一実施例による不良救済用メモリセルの他の構成例を示す図である。

【図３】本発明の一実施例による多値メモリを使用した記憶装置の構成を示すブロック図である。

【図４】図３の上位誤りビット特定回路の構成を示す回路図である。

【図５】図３の記憶装置の動作を示すタイミングチャートである。

【図６】本発明の一実施例による多値メモリのワード線の動作を示す図である。

【図７】図６に示すワード線の動作時のセンスアンプ及びラッチ・エンコード回路の出力結果を示す図である。

【図８】（ａ）～（ｃ）は本発明の一実施例による２ビット１セルにおける本発明の救済率と従来の救済率との関係を示す図である。

【図９】（ａ）～（ｃ）は本発明の一実施例による４ビット１セルにおける本発明の救済率と従来の救済率との関係を示す図である。

【図１０】本発明の一実施例による不良救済用メモリセルの別の構成例を示す図である。

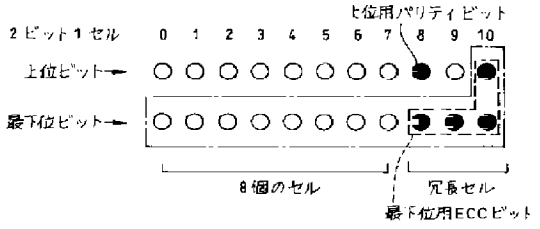
【図１１】従来例による不良救済用メモリセルの構成例を示す図である。

【図１２】従来例による多値メモリを使用した記憶装置の構成を示すブロック図である。

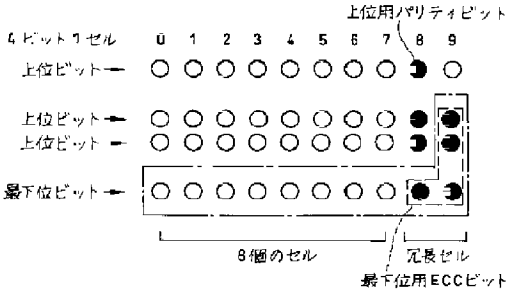
【符号の説明】

１　メモリセル
２　センスアンプ
３　ラッチ・エンコード回路
４　パリティ比較回路
５　最下位誤りビット特定回路
６　上位誤りビット特定回路
７　データ用切換え回路
８　訂正用切換え回路
９　補正回路
１１－１～１１－８　ナンド回路
１２－１～１２－８　インバータ回路

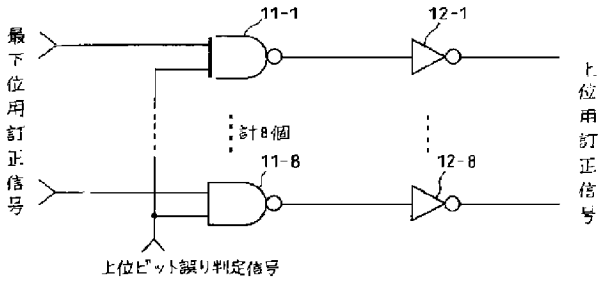
【図１】



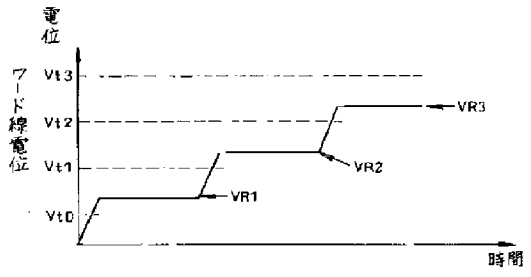
【図２】



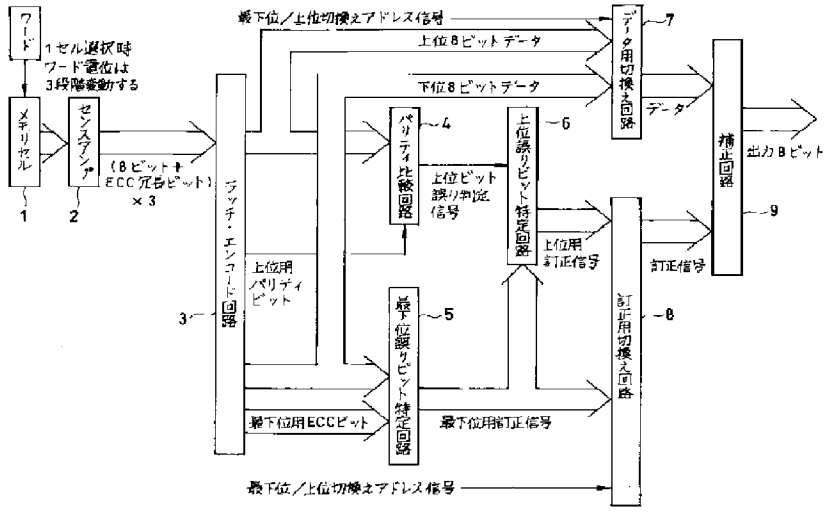
【図４】



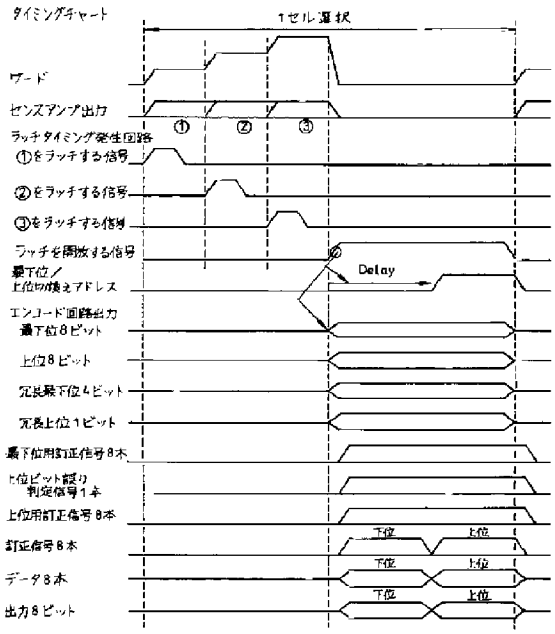
【図６】



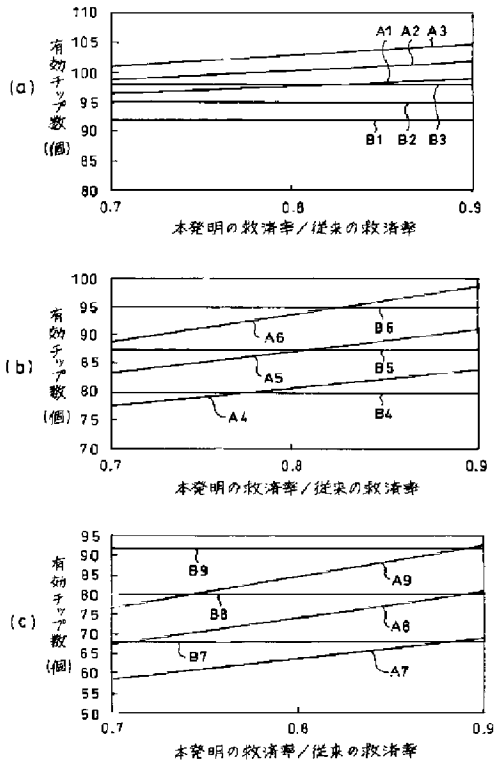
【図3】


ワード
1セル選択時
ワード電位は
3段階変動する
メモリセル
センスアンプ
(8ビット+
ECC冗長ビット)
×3
1
2
ラッチ・エンコード回路
3
最下位/上位切換えアドレス信号
上位8ビットデータ
下位8ビットデータ
データ用切換え回路
7
データ
パリティ比較回路
4
上位ビット
誤り判定
信号
上位誤りビット特定回路
6
上位用
訂正信号
訂正信号
補正回路
出力8ビット
9
訂正用切換え回路
8
上位用
パリティ
ビット
最下位誤りビット特定回路
5
最下位用ECCビット
最下位用訂正信号
最下位/上位切換えアドレス信号

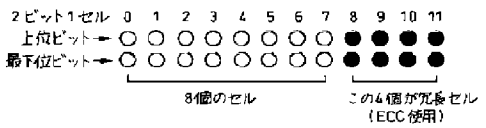
【図5】


タイミングチャート
1セル選択
ワード
センスアンプ出力
①
②
③
ラッチタイミング発生回路
①をラッチする信号
②をラッチする信号
③をラッチする信号
ラッチを開放する信号
最下位/
上位切換えアドレス
Delay
エンコード回路出力
最下位8ビット
上位8ビット
冗長最下位4ビット
冗長上位1ビット
最下位用訂正信号8本
上位ビット誤り
判定信号1本
上位用訂正信号8本
訂正信号8本
データ8本
出力8ビット
下位
上位

【図8】


(a)
有効チップ数(個)
A1 A2 A3
B1 B2 B3
本発明の救済率/従来の救済率
(b)
有効チップ数(個)
A6
B6
A5
B5
A4
B4
本発明の救済率/従来の救済率
(c)
有効チップ数(個)
B9
A9
B8
A8
B7
A7
本発明の救済率/従来の救済率

【図11】

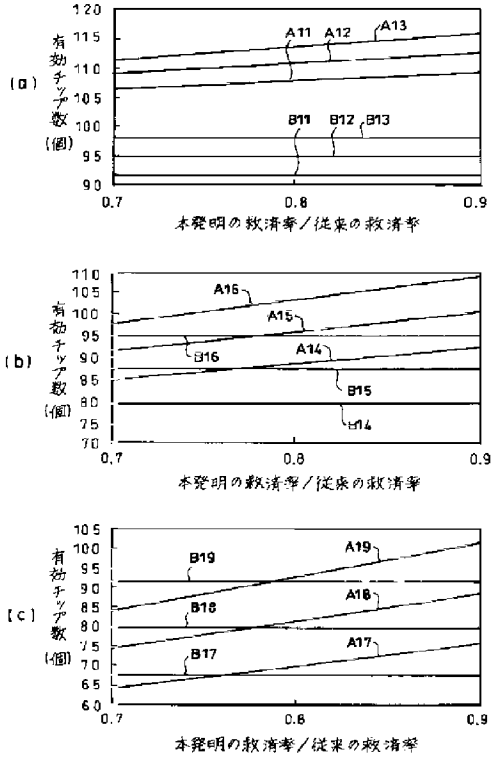

2ビット1セル 0 1 2 3 4 5 6 7 8 9 10 11
上位ビット
最下位ビット
8個のセル
この4個が冗長セル
(ECC使用)

【図７】

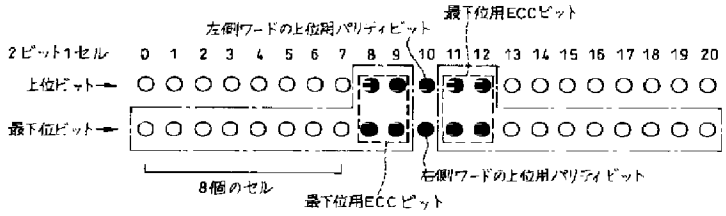
（Dm00が下位ビット、Dm01が上位ビット）

		ワード電位 VR1のときの出力	ワード電位 VR2のときの出力	ワード電位 VR3のときの出力	Dm01	Dm00
セルのVT	Vt0	1	1	1	0 (1)	0 (1)
	Vt1	0	1	1	0 (1)	1 (0)
	Vt2	0	0	1	1 (0)	0 (1)
	Vt3	0	0	0	1 (0)	1 (0)

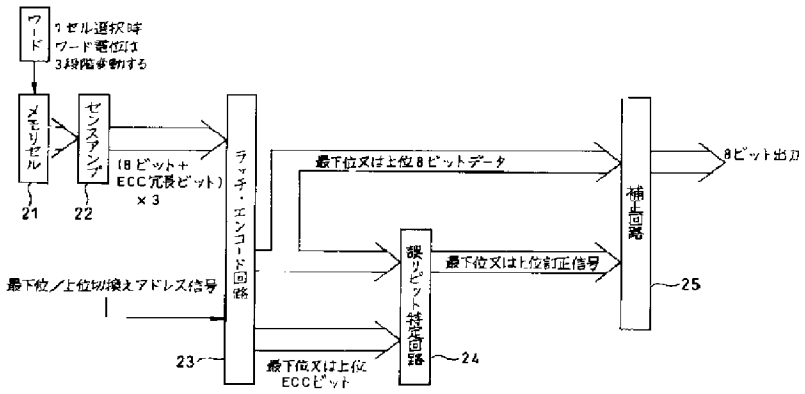
【図９】



【図１０】



【図１２】



【手続補正書】
【提出日】平成１１年１２月２０日（１９９９．１２．２０）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】特許請求の範囲
【補正方法】変更
【補正内容】
【特許請求の範囲】
【請求項１】　１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルであって、不良があるセルにおいて最下位ビットに不良がある場合にのみ救済可能としたことを特徴とする不良救済用メモリセル。
【請求項２】　前記最下位ビットのエラー検出／訂正用の最下位用ＥＣＣビットに前記最下位ビットの不良の有無及び不良位置情報を格納し、前記最下位用ＥＣＣビット以外の冗長セルに各々対応するビットの不良の有無だけの情報を格納するようにしたことを特徴とする請求項１記載の不良救済用メモリセル。
【請求項３】　前記最下位用ＥＣＣビットは、前記最下位用ＥＣＣビット以外の冗長セルの未使用領域を含むことを特徴とする請求項１または請求項２記載の不良救済用メモリセル。
【請求項４】　１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルを用いる記憶装置であって、
前記不良救済用メモリセルの不良があるセルにおいて最下位ビットに不良がある場合にのみ救済可能とし、
前記不良救済用メモリセルから読出したデータを各メモリセル毎に前記最下位ビットのデータ及び前記最下位ビット以外のビットのデータにエンコードするエンコード手段と、
前記エンコード手段でエンコードされた前記最下位ビット以外のビットのデータで作成したパリティビットと前記最下位ビット以外のビットのパリティビットとを比較して前記最下位ビット以外のビットのデータの不良の有無を判定するパリティ比較手段と、
前記エンコード手段でエンコードされた前記最下位ビットのデータを基に前記最下位ビットのデータの不良の有無と不良位置とを特定した最下位ビット用訂正信号を生成する最下位ビット誤り特定手段と、
前記パリティ比較手段の比較結果と前記最下位ビット誤り特定手段の最下位ビット用訂正信号とを基に前記最下位ビット以外のビット用訂正信号を生成する最下位ビット以外のビット誤り特定手段と、
前記エンコード手段でエンコードされた前記最下位ビットのデータと前記最下位ビット以外のビットのデータとのどちらかを選択して出力するデータ用切換手段と、
前記最下位ビット誤り特定手段の最下位ビット用訂正信号と前記最下位ビット以外のビット誤り特定手段の前記最下位ビット以外のビット用訂正信号とのどちらかを選択して出力する訂正用切換手段と、
前記訂正用切換手段からの訂正信号を基に前記データ用切換手段からのデータを補正する補正手段とを有することを特徴とする記憶装置。
【請求項５】　前記最下位ビットのエラー検出／訂正用の最下位用ＥＣＣビットに前記最下位ビットの不良の有無及び不良位置情報を格納し、前記最下位用ＥＣＣビット以外の冗長セルに各々対応するビットの不良の有無だけの情報を格納するようにしたことを特徴とする請求項

４記載の記憶装置。
【請求項６】　前記最下位用ＥＣＣビットは、前記最下位用ＥＣＣビット以外の冗長セルの未使用領域を含むことを特徴とする請求項４または請求項５記載の記憶装置。
【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】００１１
【補正方法】変更
【補正内容】
【００１１】
【課題を解決するための手段】本発明による不良救済用メモリセル、１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルであって、不良があるセルにおいて最下位ビットに不良がある場合にのみ救済可能としている。
【手続補正３】
【補正対象書類名】明細書
【補正対象項目名】００１２
【補正方法】変更
【補正内容】
【００１２】本発明による記憶装置は、１個のメモリセルに複数のビットデータを記憶しかつ冗長セルに格納したデータを用いて不良ビットの訂正を行う不良救済用メモリセルを用いる記憶装置であって、
前記不良救済用メモリセルの不良があるセルにおいて最下位ビットに不良がある場合にのみ救済可能とし、
前記不良救済用メモリセルから読出したデータを各メモリセル毎に前記最下位ビットのデータ及び前記最下位ビット以外のビットのデータにエンコードするエンコード手段と、
前記エンコード手段でエンコードされた前記最下位ビット以外のビットのデータで作成したパリティビットと前記最下位ビット以外のビットのパリティビットとを比較して前記最下位ビット以外のビットのデータの不良の有無を判定するパリティ比較手段と、
前記エンコード手段でエンコードされた前記最下位ビットのデータを基に前記最下位ビットのデータの不良の有無と不良位置とを特定した最下位ビット用訂正信号を生成する最下位ビット誤り特定手段と、
前記パリティ比較手段の比較結果と前記最下位ビット誤り特定手段の最下位ビット用訂正信号とを基に前記最下位ビット以外のビット用訂正信号を生成する最下位ビット以外のビット誤り特定手段と、
前記エンコード手段でエンコードされた前記最下位ビットのデータと前記最下位ビット以外のビットのデータとのどちらかを選択して出力するデータ用切換手段と、
前記最下位ビット誤り特定手段の最下位ビット用訂正信号と前記最下位ビット以外のビット誤り特定手段の前記最下位ビット以外のビット用訂正信号とのどちらかを選択して出力する訂正用切換手段と、
前記訂正用切換手段からの訂正信号を基に前記データ用切換手段からのデータを補正する補正手段とを備えている。
【手続補正４】
【補正対象書類名】明細書
【補正対象項目名】００２１
【補正方法】変更
【補正内容】
【００２１】この場合、多値メモリにおいてはＥＣＣを使用すると、各セルの最下位のビットデータ（８ビット）の冗長メモリセル（４ビット）には不良の有無／不良位置情報を格納し、上位のビットデータ（最下位のビットデータ以外のビットデータ）（８ビット）の冗長メモリセル（１ビット）には不良の有無のみを格納する。

(19) KOREAN INTELLECTUAL PROPERTY OFFICE

KOREAN PATENT ABSTRACTS

(11) Publication number: **100766042 B1**

(44) Date of publication of specification: **04.10.2007**

(21) Application number: **1020060123339**

(22) Date of filing: **06.12.2006**

(51) Int. Cl: *G11C 7/10 (2006.01); G11C 7/20 (2006.01); G11C 7/22 (2006.01);*

(71) Applicant: **SAMSUNG ELECTRONICS CO., LTD.**

(72) Inventor: **PARK, SUNG CHUNG**
KONG, JUN JIN
LEE, YOUNG HWAN
KANG, DONG KU

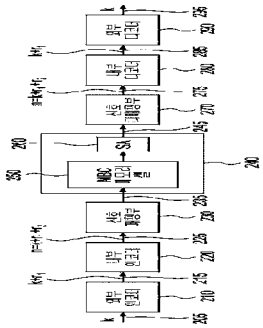
(54) **MULTI LEVEL CELL MEMORY DEVICE USING CONCATENATED CODING, CAPABLE OF CONTROLLING SIZE OF OVERHEAD DUE TO ERROR CORRECTION METHOD**

(57) Abstract:

PURPOSE: A multi level cell memory device using concatenated coding is provided to increase the number of bits stored in one memory cell stably, by adopting a new error correction method in a multi-level cell memory.

CONSTITUTION: A multi level cell memory device storing data includes an MLC(Multi Level Cell) memory cell(240) and an outer encoder(210). The outer encoder generates outer encoded bit stream by encoding the data through a first encoding method. An inner encoder(220) generates inner encoded bit stream by encoding the outer encoded bit stream through a second encoding method. A signal mapping module(230) records the data in the MLC memory cell by applying a program pulse according to the inner encoded bit stream to the MLC memory cell.

© KIPO 2007



This Facsimile First Page has been artificially created from the Korean Patent Abstracts CD Rom

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2006 年7 月6 日 (06.07.2006)

PCT

(10) 国際公開番号
WO 2006/070668 A1

(51) 国際特許分類:
G11B 20/18 (2006.01) *G11B 20/10* (2006.01)
G06F 13/00 (2006.01)

(21) 国際出願番号: PCT/JP2005/023488

(22) 国際出願日: 2005 年12 月21 日 (21.12.2005)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願 2004-378509
2004 年12 月28 日 (28.12.2004) JP

(71) 出願人 (米国を除く全ての指定国について): インターナショナル・ビジネス・マシーンズ・コーポレーション (INTERNATIONAL BUSINESS MACHINES CORPORATION) [US/US]; 10504 ニューヨーク州 アーモンク ニュー オーチャード ロード New York (US).

(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 松尾 久人 (MATSUO, Hisato) [JP/JP]; 〒2428502 神奈川県大和市下鶴間１６２３番地１４日本アイ・ビー・エム株式会社大和事業所内 Kanagawa (JP). 福田 純一 (FUKUDA, Junichi) [JP/JP]; 〒2428502 神奈川県大和市下鶴間１６２３番地１４日本アイ・ビー・エム株式会社大和事業所内 Kanagawa (JP).

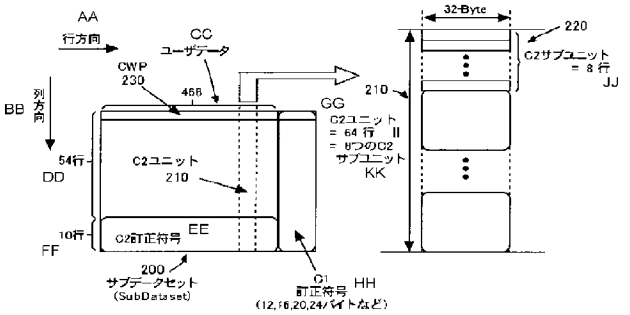
(74) 代理人: 坂口 博, 外(SAKAGUCHI, Hiroshi et al.); 〒2428502 神奈川県大和市下鶴間１６２３番地１４日本アイ・ビー・エム株式会社大和事業所内 Kanagawa (JP).

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO,

[続葉有]

(54) **Title:** INFORMATION RECORDING DEVICE, DATA-FLOW CONTROLLER FOR THE DEVICE, AND CONTROL METHOD FOR THE DATA-FLOW

(54) 発明の名称: 情報記録装置、そのデータフロー・コントローラ、及びそのデータフローの制御方法



AA - ROW DIRECTION
BB - COLUMN DIRECTION
CC - USER DATA
DD - 54 ROWS
210 - C2 UNIT
EE - C2 CORRECTION CODE
FF- 10 ROWS
200 - SUBDATA SET

GG - C2 UNIT
II - = 64 - ROWS
KK - = EIGHT- C2 SUBUNITS
HH - C1 CORRECTION CODE
(12, 16, 20, 24 BYTES)
220 -C2 SUBUNIT
JJ - = 8 ROWS

(57) **Abstract:** An information recording device for transferring error-corrected subunits only partially, when the corrected data is returned from error correcting means to an external buffer, and a data-flow control method for the device. The information recording device records the data, in which a correction code is added to user data of plural bytes transferred from a host device, in a recording medium, corrects an error of the data read from the recording medium, with the correction code, and transfers the user data to the host device. The information recording device comprises an external buffer for temporarily storing the data read from the recording medium, error correcting means for correcting

[続葉有]

WO 2006/070668 A1

RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書
— 補正書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

the error of the data transferred from the external buffer thereby to create the correction data, and for dividing the corrected data into a plurality of subunits thereby to attach a transfer flag to the subunits containing the corrected data, and data-flow means for transferring only the subunit having the transfer flag attached thereto, to the external buffer when the corrected data is sent from the error correcting means to the external buffer, and for rewriting only the portion of the subunits of the data stored in the external buffer.

(57) 要約: エラー訂正手段から外部バッファに訂正データを戻す際に、エラー訂正された一部のサブユニットのみを転送する情報記録装置、及びそのデータフロー制御方法を提供する。 解決手段として、上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、記録媒体から読取られたデータを訂正符号を用いてエラー訂正を行いユーザデータを上位装置に転送する情報記録装置であって、記録媒体から読取されたデータを一時保管する外部バッファと、外部バッファから転送されたデータのエラーを訂正して訂正データを生成し、訂正データを複数のサブユニットに分割して、訂正されたデータを含むサブユニットに転送フラグを付する、エラー訂正手段と、訂正データをエラー訂正手段から外部バッファに送る際に転送フラグを付されたサブユニットのみを外部バッファに転送し、外部バッファに保管されたデータのサブユニットの箇所のみ書換えるデータフロー手段とを備える情報記録装置である

明 細 書

情報記録装置、そのデータフロー・コントローラ、及びそのデータフローの制御方法

技術分野

[0001] 本発明は、記録媒体に記録されているデータを効率的に転送する情報記録装置のデータフロー制御に関する。

より詳細には、本発明は、ECCエンジンから外部バッファへ訂正データを効率的に転送するLTO(Lincar Tapc Opcn)方式のテープドライブ、そのデータフロー・コントローラ、及び、そのデータフローの制御方法に関する。

背景技術

[0002] 近年、社会のデジタル化の進展により、情報記録装置の需要は拡大し、情報記録装置は大容量化・高記録密度化の進展は急速である。現実の使用環境では、大容量の情報記録装置は、大量のデータを上位装置(ホスト)から要求に応じて高速に転送することを要求される。

[0003] 特に、磁気テープドライブは、シーケンシャルなデジタル・データの大規模なバックアップ装置として重要な位置を占めている。IBM社は、LTO方式の各世代のテープドライブを開発している。LTO方式のテープドライブの性能(例えば容量及び転送速度など)のターゲットは、規格により定められている(非特許文献1)。

[0004] 典型的な情報記録装置において、データを記録する場合、ホストから送られたユーザデータは、外部バッファに一時保管される。保管されたユーザデータは、ECCエンジンに転送され、ECCパリティ(例えばC2訂正符号)を生成し、ユーザーデータに例えばC2パリティを付加して外部バッファに戻される。その後、外部バッファからパリティを付加されたユーザデータが記録媒体に記録される。記録媒体のデータをホストに転送する場合、記録媒体から読み出されたデータは外部バッファに一時保管される。その保管データは、ECCパリティを付加されたままECCエンジンに転送される。ECCエンジンは、エラーデータを訂正して外部バッファに書き戻す。外部バッファの戻された訂正データは、ホストに送られる。

[0005] 図1は、LTOのドライブにおける、テープ10及びホスト70間のデータの入出力機構（入出力チャネル）の構成要素を示す。矢印A～Dは、テープ10から読み出されたデータの転送フローのステップを示す。テープ10から読まれたデータは、Buffer Manager40により外部バッファ30にいったん書き込まれる（ステップA）。外部バッファ30は、例えば、広帯域を必要とするのでDDR SDRAMであり、そのバス幅は4バイト又は8バイトである。 外部バッファ30に書き込まれたデータは、ECCエンジン（具体的にはC2ECC50）のSRAMに転送される（ステップB）。C2ECC50では、全てのデータのエラー訂正をし、訂正データをDDR SDRAM30に書き戻す（ステップC）。訂正データは、随時、DDR SDRAM30から要求に基づいてホスト70に転送される（ステップD）。

[0006] このように外部バッファ30は、チャネル10側・ホスト70側・C2ECCエンジン50からReadあるいはWriteアクセスされる。この外部バッファ30へのアクセスを制御するのがBuffer Manager40である。Buffer Manager40は、データフロー・コントローラ又はASICの機能として提供される。チャネル10から読出されたデータ（データセット）は、インターリーブ・バッファ制御20により、図2に示すサブデータセット200の各行のコードワードペア（CWP）230ごとに転送される。CWP230は、Buffer Manager40によりDDR SDRAM30に保管される（ステップA）。ホスト70には、必要に応じてデータ圧縮がされて、データが転送される（ステップD）。データフロー・コントローラの機能として、Buffer Manager40は、各ブロック20、50、及び60が効率よくDDR SDRAM30をアクセスできるよう、時分割でバスを切り換えている。図1からも判るように、C2ECC50にとっては、DDR SDRAM30から読んだデータをDDR SDRAM30に書き戻している構造上、処理時間の要求も厳しい。

[0007] このデータ転送時間の制約は、現在開発中の第3世代のLTO方式ドライブが第2世代に比べ2倍のデータ処理能力（パフォーマンス）が要求されることから、一層大きくなる。また、その記録密度は、第2世代に比べて1．8倍に上っていることから、データのエラーレートも高くなることが予見される。したがって、記録データを転送する高速化を実効する上で、読取りデータ転送をいかに効率的に行うことが重要である。

[0008] 従来の記憶装置では、ECCエンジン50及び外部バッファ30などのデータフローに

関係するコンポーネントの改良と性能の向上により、読取りデータの高速転送を実行している場合が多い。例えば、第2世代に対して第3世代のLTOでは、(1)エラー訂正機能(C2ECCエンジン)自体のパフォーマンスを高める(例えば従来の3倍以上)こと、(2)外部バッファ(DDR SDRAM)30の動作クロックを上げること、(3)例えばDDR SDRAMのバス幅広げること、などをコンポーネント個々の性能アップにより、規格で定められたデータの転送スピードを達成し得る。

[0009] LTOの各世代のドライブ間のデータの読み書きの互換性、及び、製品価格の経済性の観点から、単に高性能のコンポーネントを用いるのみでは、データ転送の効率化には限界がある。エラー訂正手段(C2ECCエンジン)から訂正データを外部バッファに送る際に、テープ記録媒体からの読取りデータのフロー自体の制御に課題があれば、その課題を改善が必要である。

[0010] 一般に外部バッファ30にDDR SDRAMを採用することにより、バーストアクセスが可能となる。バースト転送により外部バッファ30への全転送を行うことができるため、高速のデータ転送が可能となる。しかしこの全転送では、訂正データの有無に関わらず全てのデータを転送するので、データ転送効率の観点から無駄が生じる。

非特許文献1:Ultrium Generation3 16−Channel Format Specification Document U−316

特許文献1:特開2001−177419号公報

発明の開示

発明が解決しようとする課題

[0011] ところで、特許文献1は、積符号(内符号・外符号)を有するエラー訂正装置を含むデジタルVTRにおいて、外部バッファへの高速アクセスを課題とする発明である(段落10)。その課題を解決するために、入力データは内符号・外符号訂正訂段に全転送し、訂正されたデータをバンク切替によって、バースト転送している点で本発明と共通する(段落17～25及び図1及び図2)。しかし、特許文献1においても、訂正データの有無に関わらず外部バッファへバースト転送により全転送しているため、積符号訂正装置と外部バッファとの間においてデータ転送効率を依然と改善されていない。

[0012] この問題を解決するために、情報記録装置のデータフロー制御において、ECCエ

ンジンから外部バッファに転送されるデータの単位を複数のサブユニットに分割し、必要なサブユニットのみで転送できれば、無駄なデータ転送を省略できるため効率的なデータ転送が可能となる。更に、この効率的なデータ転送を、データフローに関係するコンポーネントの高性能化と併せて採用できれば、より高速のデータ転送が実現できる。

[0013] 本発明は、訂正データを外部バッファに効率的に転送する、情報記録装置、データフロー・コントローラ、及びデータフローの制御方法を提供すること目的とする。

具体的には、本発明は、エラー訂正手段から外部バッファに訂正データを戻す際に、エラー訂正された一部のサブユニットのみを転送する、LTO方式のテープドライブ、データフロー・コントローラ、及びそのデータフローの制御方法を提供することを目的とする。

課題を解決するための手段

[0014] 上記目的を達成するために、本発明は、上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、記録媒体から読取られたデータを訂正符号を用いてエラー訂正を行いユーザデータを上位装置に転送する情報記録装置であって、録媒体から読取されたデータを一時保管する外部バッファと、外部バッファから転送されたデータのエラーを訂正して訂正データを生成し、訂正データを複数のサブユニットに分割して、訂正されたデータを含むサブユニットに転送フラグを付する、エラー訂正手段と、訂正データをエラー訂正手段から外部バッファに送る際に、転送フラグを付されたサブユニットのみを外部バッファに転送し、外部バッファに保管されたデータのサブユニットの箇所のみ書換えるデータフロー手段とを備えることを特徴とする。

[0015] 好ましくは、本発明の情報記録装置の外部バッファ内の2次元のバイト・マトリックスのデータは、64行×480列、64行×484列、64行×488列、又は、64行×492列のサブデータセットであり、各行の480、484、488、又は492バイトに、468バイトのユーザデータに第1の訂正符号(p＝12、16、20、又は、24バイト)を含み、各列の64バイトのデータは、54バイトのユーザデータに第2の訂正符号(q＝10バイト)を含むことを特徴とする。

[0016] 好ましくは、本発明の情報記録装置のデータフロー手段は、外部バッファからエラー訂正手段にデータ転送する際に、外部バッファに保管されたサブデータセットを、行方向の480、484、488、又は492バイトを外部バッファのバス幅の整数倍（xバイト）を分割単位とした2次元バイト・マトリックスのユニット（64×x）を単位として転送することを特徴とする。

[0017] 好ましくは、本発明の情報記録装置のエラー訂正手段は、外部バッファから転送されたユニット（64×x）を、y行単位に複数の2次元バイト・マトリックスのサブユニット（y×x）単位に分割して、エラー訂正において訂正の生じたサブユニットに転送フラグを付し、データフロー手段は、転送フラグの付されたサブユニットのみを転送して、外部バッファのサブユニットに対応する箇所を書換えることを特徴とする。

[0018] 好ましくは、本発明のサブユニットは、列方向のM＝64をy＝8行を単位として分割した8×4、8×8、8×16、8×32、又は8×64の2次元バイト・マトリックスである、ことを特徴とする。

[0019] また、本発明の情報記録装置は、データをテープ記録媒体の走行方向（長さ方向）に記録し読取る、複数の書込み及び読込みチャネルを有するヘッドを含むLTO方式のテープドライブも含む。LTOのドライブは、テープ記録媒体は、幅方向にヘッドに対応するデータバンドに分割され、データバンドは、ヘッドの有するチャネルの各々に対応する複数のデータサブバンドに分割され、データサブバンドは、チャネルの1つによりアクセスされる複数のトラックを含み、外部バッファに保管されたサブデータセットの各行のコードワードペア（CWP）は、ローテーション方式により規定され所定の間隔で別のデータサブバンドのトラックに順次書かれる、エラー訂正の際に転送フラグを付されたサブユニットのみ外部バッファに転送することを特徴とする。

[0020] 好ましくは、本発明のLTOのドライブは、ヘッドは16のチャネルを有し、テープ記録媒体は4つのデータバンドに分割され、データバンドは16のデータサブバンドに分割され、データサブバンドは11のトラックを含む構成を特徴とする。また、本発明のLTOドライブは、この構成に限定されない。例えば、ヘッドは8のチャネルを有し、データバンドは8のデータサブバンドに分割され、データサブバンドは12及び16のトラックを含むLTOのドライブも可能である。

[0021] さらに好ましくは、本発明のLTOのドライブは、データサブバンドを6つ又は1つのシフトさせて別のデータサブバンドのトラックに、サブデータセットの各行のCWPを順次書くローテンション方式によることを特徴とする。

[0022] また、本発明は、上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、上位装置の要求に応じて読み出されたデータを訂正符号を用いてエラー訂正を行い、訂正データからユーザデータを上位装置に転送する情報記録装置においてデータのフローを制御するデータフロー・コントローラも含む。データフロー・コントローラは、外部バッファに一時保管された記録媒体から読取されたデータを受取りエラーを訂正して訂正データを生成し、訂正データを複数のサブユニットに分割して、訂正されたデータを含むサブユニットに転送フラグを付する、エラー訂正手段と、訂正データを外部バッファに送る際に、転送フラグを有するサブユニットのみを外部バッファに転送し、外部バッファ中のデータの該当サブユニット箇所のみ書換えるデータフロー手段とを備える。

[0023] また、本発明は、上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、上位装置の要求に応じて読み出されたデータを訂正符号を用いてエラー訂正を行って訂正データを生成し、訂正データを上位装置に転送するデータフローの制御方法も含む。データフローの制御方法は、外部バッファに一時保管された記録媒体から読取されたデータを受取りエラーを訂正し、訂正データを複数のサブユニットに区分けして、訂正されたバイトデータを含むサブユニットに転送フラグを付するステップと、訂正データをバッファ手段に送る際に、転送フラグを有するサブユニットのみを外部バッファに転送し、外部バッファ中のサブユニット箇所のみ書換えるステップとを備えることを特徴とする。

発明の効果

[0024] 以上のように構成される本発明は、訂正データをC2ECCエンジン50から外部バッファ30に送る際のデータフローを効率化できれば、全体として記録媒体のデータ転送の高速化に貢献できる有利な効果を有する。特に、既存の電子コンポーネントの性能向上を意識することなく、記録媒体のデータを効率的にエラー訂正及び転送をできることは、既存の情報記録装置に容易に適用できる点で有利である。また、本発

明では、エラー訂正手段の訂正データに対してエラー訂正されたサブユニットの少数のみの転送に限定できるため、結果として情報記録装置のシステムの消費電力の削減が可能となる。

発明を実施するための最良の形態

[0025] 本発明に係る最良の実施形態(以下、実施形態)を第3世代のLTO方式の磁気テープドライブについて詳細に説明する。しかしながら、この実施形態に限定されることなく、ECC訂正符号を用い、媒体ディフェクト及び書込み・読取りチャネル不良に起因するエラーをECC訂正符号を用いて訂正する各世代のLTOドライブ及び他の情報記録装置に、本発明のデータフロー制御を適用できる。

[0026] LTO方式おいては、データ転送の単位はデータセットと呼ばれる。データセットは、複数(例えば16又は64)のサブデータセットを連れたものである。転送データは、Buffer Manager40の制御により、サブデータセットを単位として外部バッファファ30例えばDDR SDRAMに一時保管される。

[0027] 図2は、外部バッファ30でのサブデータセット200(左図)と、外部バッファ30とエラー訂正手段50間の転送データの単位(C2ユニット210とC2サブユニット220)(右図)とを示す。図2を用いて、第3世代のLTOテープドライブにおけるC2エラー訂正データのデータフロー制御方法について説明する。ユーザデータ468バイトにC1訂正符号(12、16、20、24バイトなど)を付加した1行(480、484、488、492バイトなど)は、コードワードペア230(CWP)と呼ばれる。このCWP230を基本単位として、テープに記録される。1つのサブデータセットは、C2訂正符号の10行を含めて64行のCWPで構成される。テープからの読出しの単位は、データセットである。1つのデータセットは、サブデータセットを基本単位として64個を順次連ねたものである。このデータセットを外部バッファ30に転送する場合、サブデータセット200を単位として順次一時保管される。エラー訂正手段50は、C2ECCエンジンを含む。C1パリティは、CWPのユーザデータ468バイトを転送しながら付加される(On the fly)ので外部バッファには書き戻されない。C2ECCエンジンは、縦方向(列方向)の54バイトに対して10バイトのC2訂正符号を生成する。本発明では、主としてC2ECCエンジン50について説明するがそれに限定されるものではなく、以下の議論は多次元のECCエンジ

ンにも適用できる。外部バッファ30からC2ECC50へのデータの転送単位は、サブデータセットの列方向に分割したC2ユニット210と呼ぶ。第3世代LTOのデータフロー・コントローラは、C2ユニット210としてCWPの行方向を32バイトごとに区切り、その32列を単位として外部バッファ30からC2ECCエンジン50に転送する。

[0028] C2サブユニット220は、C2ユニット210を列方向に8行単位ごとに8に分割したものである。このC2サブユニット220の大きさ(32バイト×8行)は、バッファへの1回の転送サイクルの最小単位である。外部バッファ30からC2ECCエンジン50に送られるデータ単位はC2ユニットであるのに対して、外部バッファ30に戻される単位はC2ユニット210を分割したC2サブユニット220である点が本発明の特徴的である。データフロー・コントローラは、C2サブユニット256バイト(32バイトx8行)に訂正データが含まれている場合は、そのC2サブユニットを外部データバッファ30に書き戻す。一方、C2ユニット内、訂正データが含まれていないC2サブユニットは外部バッファに戻されない。C2ユニット内に訂正データが偏在する場合、訂正データの存在するC2サブユニットのみを転送すればよいため、データ転送効率を上げることができる。また、不要なC2ユニットの転送を削減できる結果として、LTOドライブの消費電力削減が可能となる。

[0029] 図3は、第3世代LTO方式のテープ300上のトラックの物理フォーマットを示す。テープ上のごみや傷などのディフェクト、及び、読み書きチャネルの不良が、データエラーの原因となる。テープ上の典型なディフェクトが、サブデータセット200、C2ユニット210、又はC2サブユニット220に与える影響を考察し、各行のCWPのローテーションという書込み方式について説明する。まず、C2サブユニット220とLTOのテープ上のトラックとの配置関係ついて説明する。磁気テープ上には潜在的にディフェクトが存在する。第2世代より高い記録密度を要求される第3世代のLTOでは、テープ上のトラック(データ)はディフェクトの影響を受けやすくなる。そこで、LTOでは、テープ上でのディフェクトの形状、及び、位置を考えて、サブデータセット200の各行のCWP230のローテンション方式による書込みを規定している(非特許文献1)。

[0030] 図3の左図は、テープ300の幅方向がサーボバンド(Servo Band)により4つのデータバンド(DataBand)に分割されているのを示す。テープヘッド310は、各データ

バンドを読み書きする。中央の図は、各データバンドが複数（例えば16）のデータサブバンド（DataSubBand）を含む構成を示す。各データサブバンド（DataSubBand＃0～15）は、1つのテープヘッド310に含まれる対応するチャネル（Channel＃0～15）によりアクセスされる。右図は、テープヘッドの1つのチャネルによりアクセスされる1つのデータサブバンドのトラック構造を示す。第3世代では、1つのデータサブバンドは11本のトラックからなる。これら11のトラックは、1つのチャネルにより向きを変えながらアクセスされる。1つのチャネルは、矢印の示すようの先ず外側の物理トラックからアクセスする。そのチャネルはそのトラック即ちテープの終端に達すると、点線に示す反対側のトラックに向きを変えてアクセスする。このようにチャネルがテープの終端に至るごとに、順次外側のトラックから内側のトラックに向きを変えつつアクセスする。16のデータサブバンドの同じ位置のトラックは1本ずつ、テープヘッドの16のチャネル（channel＃0～15）により平行して同時にアクセスされる。

[0031] 図4は、図3の中央の図に示すデータバンドの物理フォーマットの詳細と、サブデータセットの1行CWPの関係を示す。

LTOでは、1つのサブデータセット200へのディフェクトによる影響を小さくするために、サブデータセットの各行のデータ（CWP）を別のチャンネルにより別のデータサブバンドに書く。ホストからの転送データは、テープヘッドの有する16チャネル各々により、CWPを単位としてデータサブバンド中のトラックに書き込まれる。次のCWPの書き込みは、ディフェクトに起因するエラーを分散させるために、書かれるデータサブバンド＃をローテーションさせる。図4は、1つのサブデータセットの各行のCWPを順次、データサブバンドを6つシフトさせローテーション方式の書込み態様を示す。例えば1つのサブデータセットの1のCWPは、チャネル＃0より対応するデータサブバンド＃0内のトラックに記録される。次の行のCWPは、チャネル＃6によりデータサブバンド＃6内のトラックに記録される。

[0032] サブデータセット200の各行のCWPは、順次縦方向の16のデータサブバンド＃0～15のトラックに右方向に向かって記録されていく。図4に示す黒で示したパケットは、1行のCWP230（図2）を意味する。黒のパケットは、同じサブデータセットの各行のCWPであり、左から順にサブデータセットの行のCWPに対応する。例えば、一番左

に示すデータサブバンド0のトラック上のパケットは、図2のサブデータセット200の1行目のCWPとなる。次にデータサブバンド＃6のパケットが同じサブデータセットの2行目のCWPに、データサブバンド＃12のパケットが同じサブデータセットの3行目のCWPに対応する。黒のパケット(CWP)は1つのトラック幅を有するが、図4において、説明の便宜のため各々のパケット(CWP)の幅を拡大して示す。図4を参照すると、最初の8個のパケットが同じC2サブユニット220(図2を参照)に入ることがわかる。図4は、同一のC2サブユニット220に入るデータ(CWP)を点線で区切って示す。

[0033] 図5に、テープ上の1つのデータバンド上の典型的なディフェクト分布の例を示す。テープ上のディフェクトによる読み取りエラーは、ごみなどの付着や傷、チャネル不良などいくつかに分類される。本発明では、C2ECCエンジンの訂正能力に対するディフェクトの影響の観点から、ディフェクトの形状を次の1～5に分類する。ただしディフェクトは5mm幅程度の帯状や、10mm程度のものを想定する。これ以上の大きさのディフェクトでは、サブデータセットのC2ECCエンジンによる訂正ができない場合が多い。また、軽微な読み取りエラーはC1訂正符号を用いてC1ECCエンジンにより十分訂正可能である。

[0034] 図5に与える各種ディフェクト1～5が、サブデータセット200内のC2ユニット210に与える影響(エラー)について説明する。

1.テープ走行方向のディフェクト:

3個のC2サブユニットにまたがる訂正が発生する。

2.複数のデータバンドに渡るテープ横方向のディフェクト:

隣のデータバンドは別データであり、1個のC2サブユニットの訂正が発生する。

3.円状のディフェクト:

2個のC2サブユニット訂正が発生する。

4.テープ対角方向の細いディフェクト:

隣のデータバンドは別データであり、1～2個のC2サブユニットの訂正が発生する。

5.トラック1本が読み取り不可、あるチャネル1本が読めない場合:

1個おきのC2サブユニットの訂正が発生する。

この中で最後の5はデッドトラックと呼ばれ、チャネル不良などに起因するエラーで

あるが、ディフェクトのひとつと考える。図5の例では、5はデータサブバンド14の全てが読めないケースである。このケースでは、データサブバンド＃14に含まれるパケット（CWP）は1個おきのC2サブユニットに含まれ、1つのC2ユニットの8のC2サブユニット中4つの訂正が必要となる。

[0035] 特に、サブデータセットの各行のCWPを上述のローテンション方式によりテープに記録する場合、トラック方向に連続するディフェクト（図5の1や5のタイプ）の影響は受けにくい仕組みになっている。図2に示すC2ユニット210はエラーデータを含むとしても、そのエラーデータは特定のC2サブユニット220に集中し、他のC2サブユニット220には含まれないケースが頻繁に起こる。同様に、図5のテープの幅方向に伸びるディフェクト（2～4のタイプ）では、8のC2サブユニットのうち2つほどにエラーが含まれる。両タイプのエラーに対して、C2ECCエンジン50から外部バッファ30にC2ユニット210を戻す場合、エラーの有無によりC2サブユニットごとに転送フラグを立て、必要なサブユニットのみC2エンジンから転送すると、外部バッファの転送効率を上げることができる。

[0036] このように典型的なディフェクトの場合、2つ程度のC2サブユニットのみのデータ訂正が必要の場合がほとんどである。C2ECCエンジン50から外部バッファ30への転送は、C2ユニットの8つのC2サブユニットうち2つに限定できる。

[0037] 更に、この転送されるC2サブユニットの削減は、バッファも含めたシステムの消費電力削減の効果もある。第3世代のLTOでは第2世代に対して高速転送を要求されるため、外部バッファ30のバス幅を第2世代の4バイトから8バイトに広げている。そのため第三世代のLTOでは、8バイトを1ワードと定義する。LTOのデータフロー・コントローラは、一回のバスサイクルで計168ワードを外部バッファ30に転送する。168ワードの内でC2ECCエンジンの帯域は、C2サブユニット1個分即ち32ワードを占める。つまりC2訂正のバス占有率は、19％（＝32÷168）である。LTOにおけるC2訂正データの分布性を考えればC2ECCエンジンから外部バッファへの転送が2つのC2サブユニットの転送しか発生しない場合が多いと考えられる。従って、データフロー・コントローラが訂正されたC2サブユニットのみを転送する機能を採用すると、LTOドライブは5％（19％×2／8)までバス帯域効率を低減できる。

[0038] 上述したように、第3世代のLTOにおいてテープ上の特定の部分に局在したディフェクトを回避するデータ分散の方式(例えばローティション)を適用されたとしても、サブデータセット(又はC2サブユニット)においてもなおデータエラーは特定の箇所に(C2サブユニット)に集中して現れる特性がある。この特性を考慮すれば、本発明のLTOドライブへの適用範囲は今後とも広がることが考えられる。なお、第1及び第2世代のLTOでは、それぞれデータバンドの数は4、チャネル及びデータサブバンドの数は8、トラック数は12、16、ローテンション方式でのデータサブバンドのシフトは1である。

[0039] 以上の実施形態では、LTOのテープドライブのおけるC2ECCエンジンから外部バッファへの訂正データの効率的な転送について説明したが、本発明は、これらテープドライブの技術分野に限定されものではない。近年の記録媒体の高密度化、ECCエンジンの訂正能力の拡大、ホストへの高速データ転送化などの技術的要求は全ての記憶装置に存在し、将来的にはECCエンジンが処理するデータ単位は大きくなる。一方、記録媒体の品質の向上により、エラーデータはそのECCエンジンの処理するデータ単位の特定の箇所に局在することが予想される。このような構成の記憶装置に、本発明のデータフローの制御方法を適用できれば、ECCエンジンから外部バッファへの無駄なデータ転送を回避できるため、データ転送の高速化、及びシステムの消費電力の低減が実現できる。

図面の簡単な説明

[0040] [図1]は、情報記録装置のデータの入出力機構を示す。

[図2]は、本発明のエラー訂正手段50(ECCエンジン)に送られるC2ユニットとそれを分割したC2サブユニットとの例を示す(特に右図)。

[図3]は、典型的な第3世代LTO方式のテープ300上のトラックの物理フォーマットを示す。

[図4]は、図3の中央の図に示すデータバンドの物理フォーマットと、サブデータセットの各行のCWPの配置関係を示す。

[図5]は、テープ上の典型的なディフェクト分布の例を示す。

請求の範囲

[1] 上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、前記記録媒体から読取られた前記データを前記訂正符号を用いてエラー訂正を行い前記ユーザデータを前記上位装置に転送する情報記録装置であって、

前記記録媒体から読取されたデータを一時保管する外部バッファと、

前記外部バッファから転送された前記データのエラーを訂正して訂正データを生成し、前記訂正データを複数のサブユニットに分割して、訂正されたデータを含むサブユニットに転送フラグを付する、エラー訂正手段と、

前記訂正データを前記訂正手段から前記外部バッファに送る際に、前記転送フラグを付されたサブユニットのみを前記外部バッファに転送し、前記外部バッファに保管されたデータの前記サブユニットの箇所のみ書換えるデータフロー手段と、

を備える情報記録装置。

[2] 前記データは、複数の行（1行は横方向のバイトデータ）×複数の列の数（1列は縦方向のバイトデータ）の2次元のバイト・マトリックスを基本単位として構成され、

前記行のバイトデータは、ユーザデータに前記エラー訂正手段により生成された第1の訂正符号を含み、

前記列のバイトデータは、ユーザデータに前記エラー訂正手段により生成された第2の訂正符号を含む、

ことを特徴とする請求項1記載の情報記録装置。

[3] 前記データは、M行×N列の2次元のバイト・マトリックスを単位として前記外部バッファに保管され、

前記行のNバイトは、前記ユーザデータ（nバイト）に前記エラー訂正手段により生成された第1の訂正符号（pバイト）を含み、

前記列のMバイトは、前記ユーザデータ（mバイト）に前記エラー訂正手段により生成された第2の訂正符号（qバイト）を含む、

ことを特徴とする請求項2記載の情報記録装置。

[4] 前記2次元のバイト・マトリックスは、64行×480列、64行×484列、64行×488列

、又は、64行×492列（M＝64；N＝480、484、488、又は492）のサブデータセットであり、

前記行のN＝480、484、488、又は492バイトは、前記ユーザデータのn＝468バイトに第1の訂正符号（p＝12、16、20、又は、24バイト）を含み、

前記列の64バイトは、前記ユーザデータのm＝54バイトに第2の訂正符号（q＝10バイト）を含む、

ことを特徴とする請求項3記載の情報記録装置。

[5] 前記データフロー手段は、前記外部バッファから前記エラー訂正手段にデータ転送する際に、前記外部バッファに保管されたサブデータセットを、前記行方向のNを前記外部バッファのバス幅の整数倍（xバイト）を分割単位とした2次元バイト・マトリックスの第2のユニット（64×x）を単位として転送する請求項4記載の情報記録装置。

[6] 前記外部バッファはDDR SDRAMであり、前記分割単位はx＝4、8、16、32、又は64バイトであり、前記ユニット（64×x）は、64×4、64×8、64×12、64×32、又は64×64の2次元バイト・マトリックスである、

ことを特徴とする請求項5記載の情報記録装置。

[7] 前記エラー訂正手段は、前記外部バッファから転送されたユニット（64×x）を、y行単位に複数の2次元バイト・マトリックスのサブユニット（y×x）単位に分割して、前記エラー訂正において訂正の生じたサブユニットに転送フラグを付し、

前記データフロー手段は、前記転送フラグの付されたサブユニットのみを転送して、前記外部バッファの前記サブユニットに対応する箇所を書換える、

ことを特徴とする請求項7記載の情報記録装置。

[8] 前記サブユニット（y×x）は、列方向のM＝64をy＝8行を単位として分割した8×4、8×8、8×16、8×32、又は8×64の2次元バイト・マトリックスである、ことを特徴とする請求項7記載の情報記録装置。

[9] 前記情報記録装置は、前記データをテープ記録媒体の走行方向（長さ方向）に記録し読取る、複数の書込み及び読込みチャネルを有するヘッドを含むLTOドライブであり、

前記テープ記録媒体は、

幅方向に前記ヘッドに対応するデータバンドに分割され、

前記データバンドは、前記ヘッドの有する前記チャネルの各々に対応する複数のデータサブバンドに分割され、

前記データサブバンドは、前記チャネルの1つによりアクセスされる複数のトラックを含み、

前記外部バッファに保管されたサブデータセットの各行のコードワードペア（CWP）は、ローテーション方式により規定され所定の間隔で別のデータサブバンドのトラックに順次書かれる、

前記エラー訂正の際、前記転送フラグを付されたサブユニットのみ外部バッファに転送することを特徴とする請求項7又は8記載の情報記録装置。

[10] 前記ヘッドは16のチャネルを有し、

前記テープ記録媒体は、4つのデータバンドに分割され、

前記データバンドは、16のデータサブバンドに分割され、

前記データサブバンドは11のトラックを含む、

ことを特徴とする請求項9に記載の情報記録装置。

[11] 前記ローテーション方式は、前記データサブバンドを6つのシフトさせて別のデータサブバンドのトラックに、前記サブデータセットの各行のCWPを順次書く、

ことを特徴とする請求項10記載の情報記録装置。

[12] 上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、前記上位装置の要求に応じて読み出された前記データを前記訂正符号を用いてエラー訂正を行って訂正データを生成し、前記訂正データからユーザデータを前記上位装置に転送する情報記録装置において前記データのフローを制御するデータフロー・コントローラであって、

外部バッファに一時保管された前記記録媒体から読取されたデータを受取りエラーを訂正し、前記訂正データを複数のサブユニットに分割して、訂正されたデータを含むサブユニットに転送フラグを付する、エラー訂正手段と、

前記訂正データを外部バッファに送る際に、前記転送フラグを有するサブユニットのみを前記外部バッファに転送し、前記外部バッファ中のデータの該当サブユニット

箇所のみ書換えるデータフロー手段と、

を備えるデータフロー・コントローラ。

[13] 前記データは、前記2次元のバイト・マトリックスは、64行×480列、64行×484列、64行×488列、又は、64行×492列のサブデータセットであり、

前記行の480、484、488、又は492バイトは、前記ユーザデータのn＝468バイトに12、16、20、又は、24バイトの第1の訂正符号を含み、

前記列の64バイトデータは、54バイトの前記ユーザデータに10バイトの第2の訂正符号を含む、

ことを特徴とする請求項12記載のデータフロー・コントローラ。

[14] 前記データフロー手段は、前記外部バッファから前記エラー訂正手段にデータ転送する際に、前記外部バッファに保管されたサブデータセットを、前記行方向のNバイト（N＝480、484、488、又は、492）を前記外部バッファのバス幅の整数倍（xバイト）を分割単位とした2次元バイト・マトリックスの第2のユニット（64×x）を単位として転送する請求項13記載のデータフロー・コントローラ。

[15] 前記エラー訂正手段は、前記外部バッファから転送されたユニット（64×x）を、y行単位に複数の2次元バイト・マトリックスのサブユニット（y×x）単位に分割して、前記エラー訂正において訂正の生じたサブユニットに転送フラグを付し、

前記データフロー手段は、前記転送フラグの付されたサブユニットのみを転送して、前記外部バッファの前記サブユニットに対応する箇所を書換える、

ことを特徴とする請求項14記載のデータフロー・コントローラ。

[16] 上位装置から転送された複数バイトのユーザデータに訂正符号を付加したデータを記録媒体に記録し、前記上位装置の要求に応じて読み出された前記データを前記訂正符号を用いてエラー訂正を行いって訂正データを生成し、前記訂正データを前記上位装置に転送するデータフローの制御方法であって、

外部バッファに一時保管された前記記録媒体から読取された前記データを受取りエラーを訂正し、前記訂正データを複数のサブユニットに区分けして、訂正されたバイトデータを含むサブユニットに転送フラグを付するステップと、

前記訂正データをバッファ手段に送る際に、前記転送フラグを有するサブユニット

のみを前記外部バッファに転送し、前記外部バッファ中のサブユニット箇所のみ書換えるステップと、

を備えるデータフローの制御方法。

補正書の請求の範囲
[2006年4月24日 (24.04.06) 国際事務局受理：出願当初の請求の範囲 7 は補正された；
他の請求の範囲は変更なし。]

、又は、64行×492列(M＝64;N＝480、484、488、又は492)のサブデータセットであり、

前記行のN＝480、484、488、又は492バイトは、前記ユーザデータのn＝468バイトに第1の訂正符号(p＝12、16、20、又は、24バイト)を含み、

前記列の64バイトは、前記ユーザデータのm＝54バイトに第2の訂正符号(q＝10バイト)を含む、

ことを特徴とする請求項3記載の情報記録装置。

[5] 前記データフロー手段は、前記外部バッファから前記エラー訂正手段にデータ転送する際に、前記外部バッファに保管されたサブデータセットを、前記行方向のNを前記外部バッファのバス幅の整数倍(xバイト)を分割単位とした2次元バイト・マトリックスの第2のユニット(64×x)を単位として転送する請求項4記載の情報記録装置。

[6] 前記外部バッファはDDR SDRAMであり、前記分割単位はx＝4、8、16、32、又は64バイトであり、前記ユニット(64×x)は、64×4、64×8、64×12、64×32、又は64×64の2次元バイト・マトリックスである、

ことを特徴とする請求項5記載の情報記録装置。

[7] 前記エラー訂正手段は、前記外部バッファから転送されたユニット(64×x)を、y行単位に複数の2次元バイト・マトリックスのサブユニット(y×x)単位に分割して、前記エラー訂正において訂正の生じたサブユニットに転送フラグを付し、

前記データフロー手段は、前記転送フラグの付されたサブユニットのみを転送して、前記外部バッファの前記サブユニットに対応する箇所を書換える、

ことを特徴とする請求項6記載の情報記録装置。

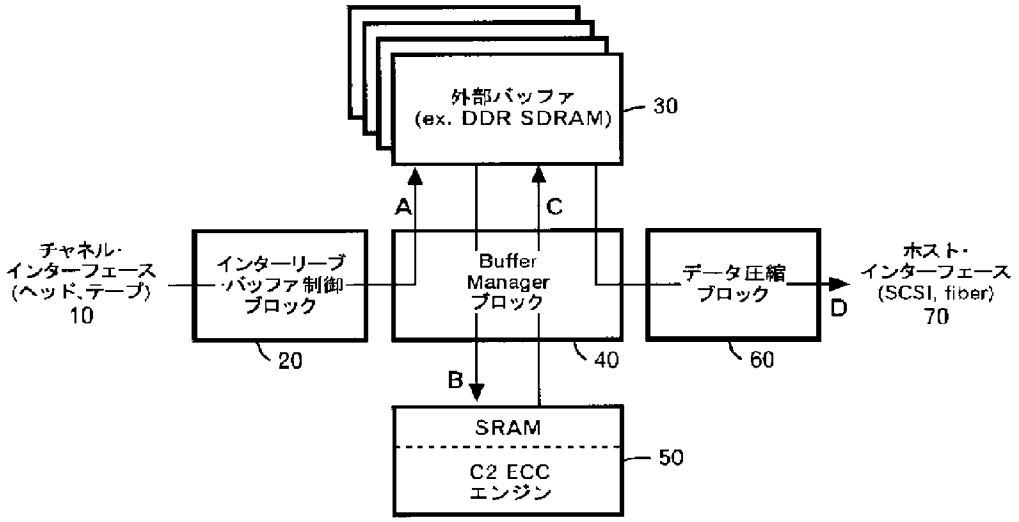
[8] 前記サブユニット(y×x)は、列方向のM＝64をy＝8行を単位として分割した8×4、8×8、8×16、8×32、又は8×64の2次元バイト・マトリックスである、ことを特徴とする請求項7記載の情報記録装置。

[9] 前記情報記録装置は、前記データをテープ記録媒体の走行方向(長さ方向)に記録し読取る、複数の書込み及び読込みチャネルを有するヘッドを含むLTOドライブであり、

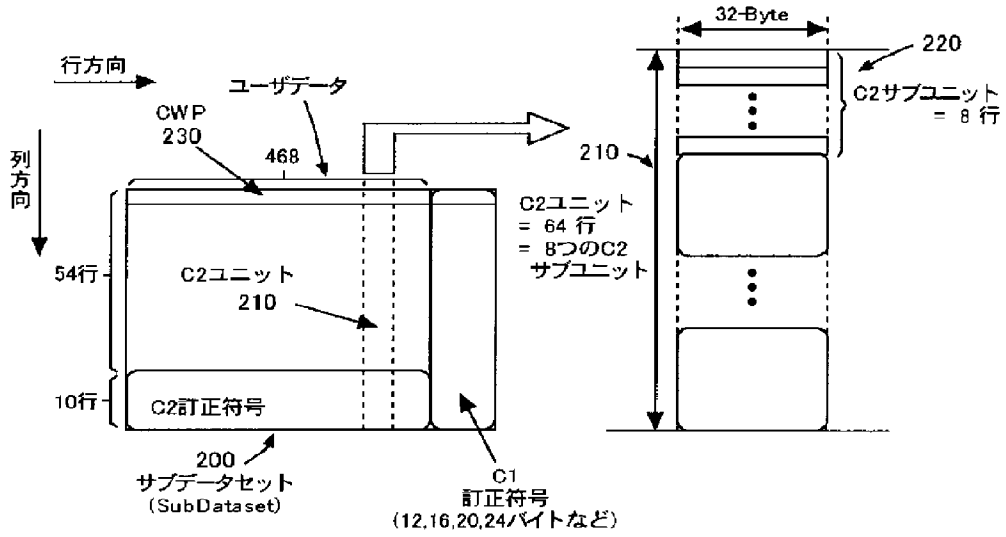
前記テープ記録媒体は、

Ex.1002 / Page 161 of 744
Sandisk Technologies, Inc. et al

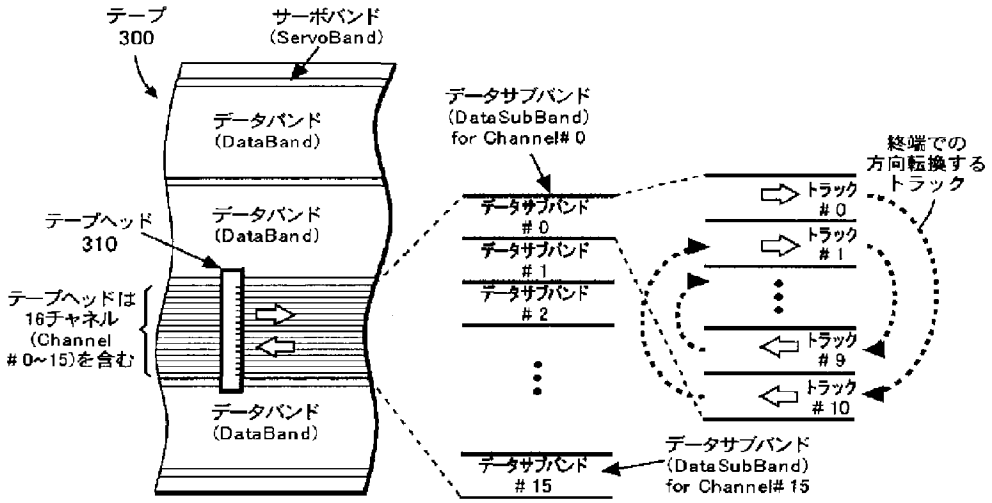
[図1]


外部バッファ
(ex. DDR SDRAM)
30
A
C
チャネル・
インターフェース
(ヘッド、テープ)
10
インターリーブ
・バッファ制御
ブロック
Buffer
Manager
ブロック
データ圧縮
ブロック
D
ホスト・
インターフェース
(SCSI, fiber)
70
20
40
60
B
SRAM
C2 ECC
エンジン
50

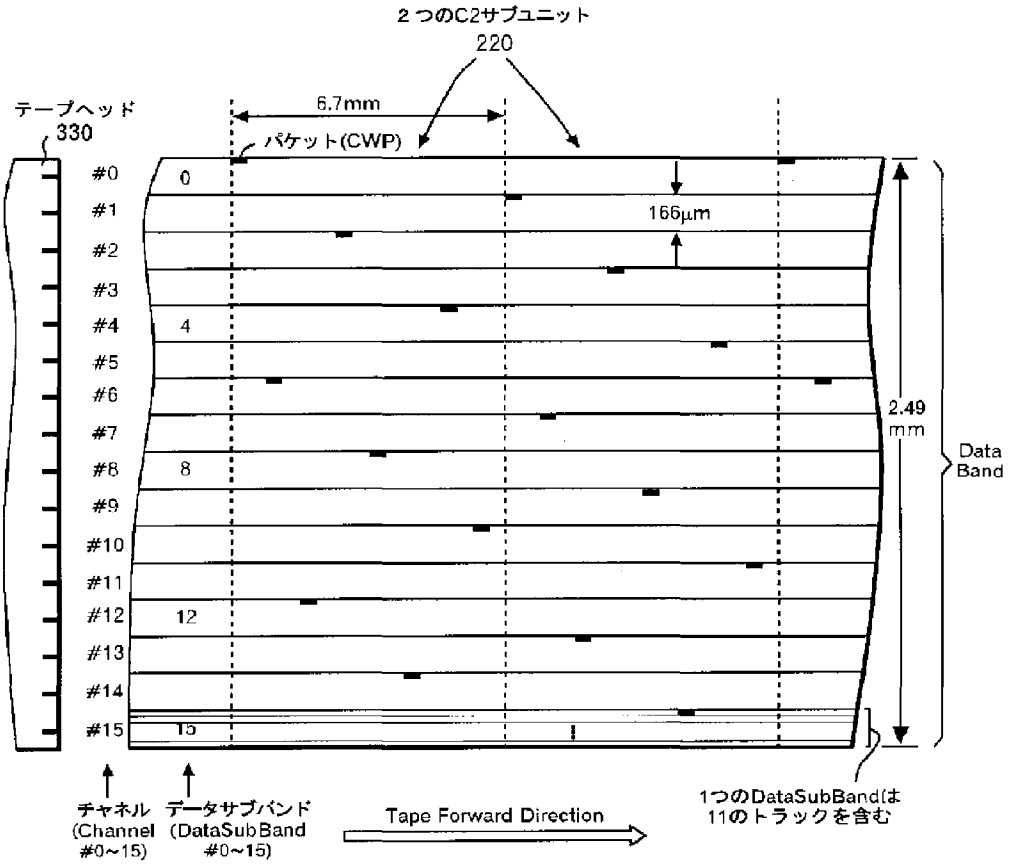
[図2]


行方向
ユーザデータ
CWP
230
468
列方向
54行
C2ユニット
210
10行
C2訂正符号
200
サブデータセット
(SubDataset)
C1
訂正符号
(12,16,20,24バイトなど)
32-Byte
220
C2サブユニット
= 8 行
210
C2ユニット
= 64 行
= 8つのC2
サブユニット

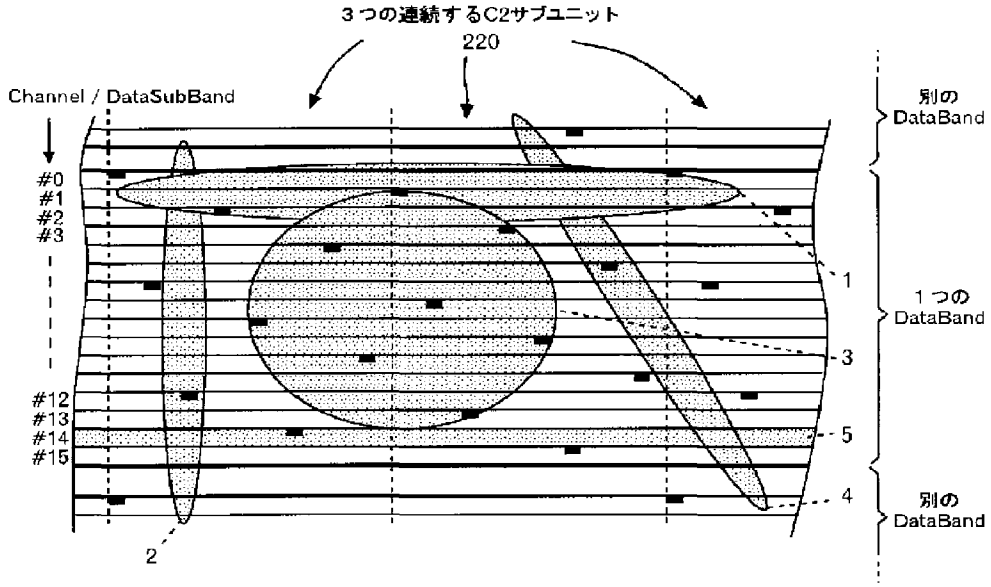
[図3]


テープ
300
サーボバンド
(ServoBand)
データバンド
(DataBand)
データバンド
(DataBand)
データバンド
(DataBand)
テープヘッド
310
テープヘッドは
16チャネル
(Channel
#0~15)を含む
データサブバンド
(DataSubBand)
for Channel# 0
データサブバンド
0
データサブバンド
1
データサブバンド
2
データサブバンド
15
データサブバンド
(DataSubBand)
for Channel# 15
終端での
方向転換する
トラック
トラック
0
トラック
1
トラック
9
トラック
10

[図4]


2つのC2サブユニット
220
テープヘッド
330
6.7mm
パケット(CWP)
#0
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
#11
#12
#13
#14
#15
0
4
8
12
15
166μm
2.49
mm
Data
Band
チャネル
(Channel
#0~15)
データサブバンド
(DataSubBand
#0~15)
Tape Forward Direction
1つのDataSubBandは
11のトラックを含む

[図5]


3つの連続するC2サブユニット
220
Channel / DataSubBand
#0
#1
#2
#3
#12
#13
#14
#15
1
2
3
4
5
別の DataBand
1つの DataBand
別の DataBand

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/023488

A. CLASSIFICATION OF SUBJECT MATTER
G11B20/18(2006.01), ***G06F13/00***(2006.01), ***G11B20/10***(2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G11B20/18(2006.01), ***G06F13/00***(2006.01), ***G11B20/10***(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2006
Kokai Jitsuyo Shinan Koho 1971-2006 Toroku Jitsuyo Shinan Koho 1994-2006

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 4-332971 A (Sanyo Electric Co., Ltd.), 19 November, 1992 (19.11.92), Full text; all drawings (Family: none)	1-16
A	JP 9-251730 A (Toshiba Corp.), 22 September, 1997 (22.09.97), Full text; all drawings (Family: none)	1-16
A	JP 2000-57711 A (Victor Company Of Japan, Ltd.), 25 February, 2000 (25.02.00), Full text; all drawings (Family: none)	1-16

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 January, 2006 (18.01.06)	31 January, 2006 (31.01.06)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/023488

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 8-45199 A (Sony Corp.), 16 February, 1996 (16.02.96), Full text; all drawings (Family: none)	1-16
A	JP 11-88840 A (Canon Inc.), 30 March, 1999 (30.03.99), Par. Nos. [0012] to [0013]; Fig. 9 (Family: none)	1-16

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００５／０２３４８８

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　*G11B20/18(2006.01), G06F13/00(2006.01), G11B20/10(2006.01)*

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　*G11B20/18(2006.01), G06F13/00(2006.01), G11B20/10(2006.01)*

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００６年
日本国実用新案登録公報	１９９６－２００６年
日本国登録実用新案公報	１９９４－２００６年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Ａ	ＪＰ　４－３３２９７１　Ａ（三洋電機株式会社） １９９２．１１．１９，全文，全図（ファミリーなし）	１－１６
Ａ	ＪＰ　９－２５１７３０　Ａ（株式会社東芝） １９９７．０９．２２，全文，全図（ファミリーなし）	１－１６
Ａ	ＪＰ　２０００－５７７１１　Ａ（日本ビクター株式会社） ２０００．０２．２５，全文，全図（ファミリーなし）	１－１６

☑ Ｃ欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
１８．０１．２００６	３１．０１．２００６

国際調査機関の名称及びあて先	特許庁審査官（権限のある職員）	５Ｑ	３１４６
日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	松平　英		
	電話番号　０３－３５８１－１１０１　内線　３５９１		

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（２００５年４月）

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Ａ	ＪＰ　８－４５１９９　Ａ（ソニー株式会社） １９９６．０２．１６，全文，全図（ファミリーなし）	１－１６
Ａ	ＪＰ　１１－８８８４０　Ａ（キヤノン株式会社） １９９９．０３．３０，段落【００１２】－段落【００１３】，第９図（ファミリーなし）	１－１６

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 July 2004 (22.07.2004)

PCT

(10) International Publication Number
WO 2004/062113 A1

(51) International Patent Classification[7]: H03M 13/35, 13/27

(21) International Application Number: PCT/US2003/038223

(22) International Filing Date: 3 December 2003 (03.12.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/434,232	16 December 2002 (16.12.2002)	US
60/444,068	30 January 2003 (30.01.2003)	US
60/470,921	15 May 2003 (15.05.2003)	US
60/494,404	11 August 2003 (11.08.2003)	US

(71) Applicant: **INTERDIGITAL TECHNOLOGY CORPORATION** [US/US]; 300 Delaware Avenue, Suite 527, Wilmington, DE 19801 (US).

(72) Inventors: **PIETRASKI, Philip, J.**; 199 N. Oak Street, North Massapequa, NY 11758 (US). **STERNBERG, Gregory, S.**; 50 Brompton Road, Apt. 2B, Great Neck, NY 11021 (US).

(74) Agent: **WOLINSKY, Scott**; Volpe and Koenig, P.C., United Plaza, Suite 1600, 30 South 17th Street, Philadelpha, PA 19103 (US).

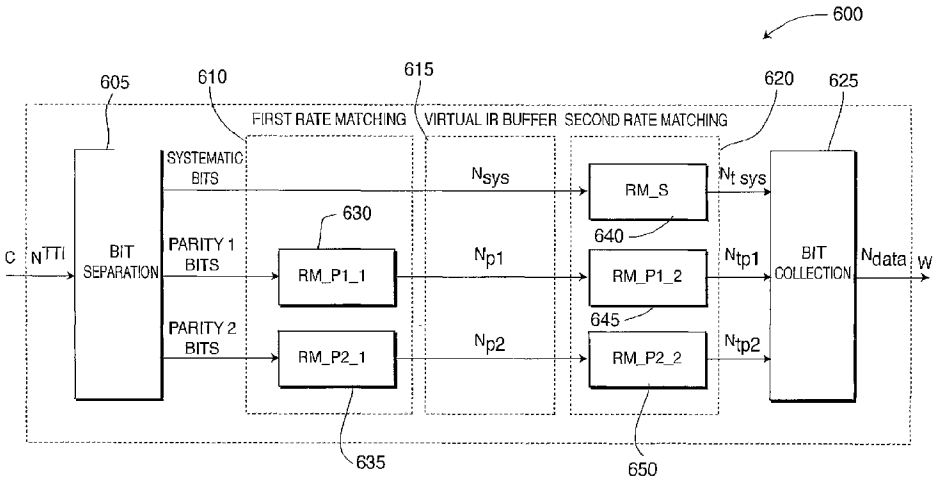
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DETECTION, AVOIDANCE AND/OR CORRECTION OF PROBLEMATIC PUNCTURING PATTERNS IN PARITY BIT STREAMS USED WHEN IMPLEMENTING TURBO CODES



(57) Abstract: Detecting, avoiding and/or correcting problematic puncturing patterns in parity bit streams used when implementing punctured Turbo codes is achieved without having to avoid desirable code rates. This enables identification/avoidance of regions of relatively poor Turbo code performance. Forward error correction comprising Turbo coding and puncturing achieves a smooth functional relationship between any measure of performance and the effective coding rate resulting from combining the lower rate code generated by the Turbo encoder (600) with puncturing of the parity bits. In one embodiment, methods to correct/avoid degradations due to Turbo coding are implemented by puncturing interactions when two or more stages of rate matching (610, 620) are employed.

WO 2004/062113 A1

[0001] # DETECTION, AVOIDANCE AND/OR CORRECTION OF PROBLEMATIC PUNCTURING PATTERNS IN PARITY BIT STREAMS USED WHEN IMPLEMENTING TURBO CODES

[0002] ## FIELD OF THE INVENTION

[0003] This invention generally relates to the use of Turbo codes in general, and, more particularly to a method for detecting and correcting performance degradation in punctured Turbo codes using a Turbo encoder.

[0004] ## BACKGROUND

[0005] Wireless communication systems are well known in the art. Generally, such systems comprise communication stations that transmit and receive wireless communication signals between each other. Typically, base stations are provided which are capable of conducting wireless concurrent communications with a plurality of subscriber stations. In Code Division Multiple Access (CDMA) systems specified by the Third Generation Partnership Project (3GPP), base stations are called Node Bs, subscriber stations are called User Equipments (UE) and the wireless interface between the Node Bs and UEs is known as the Uu interface. Figure 1 illustrates a typical 3GPP CDMA system.

[0006] The Uu radio interface of a 3GPP communications system uses Transport Channels (TrCH) for transfer of user data and signaling between UEs and Node Bs. In 3GPP Time Division Duplex (TDD) communications, TrCH data is conveyed by one or more physical channels defined by mutually exclusive physical resources. TrCH data is transferred in sequential groups of Transport Blocks (TB) defined as Transport Block Sets (TBS). Each TBS is transmitted in a given Transmission Time Interval (TTI) which may span a plurality of consecutive system time frames. A typical system time frame is 10 milliseconds and TTIs are currently specified as spanning 1, 2, 4 or 8 of such time frames.

[0007] Figure 2 illustrates the processing of TrCHs in TTD mode into a Coded Composite TrCH (CCTrCH) and then into one or more physical channel data streams in accordance with 3GPP TS 25.222 v3.8.0. Starting with the TB of

Ex.1002 / Page 170 of 744
Sandisk Technologies, Inc. et al

data, Cyclic Redundancy Check (CRC) bits are attached and TB concatenation and code block segmentation is performed. Convolution coding or turbo coding is then performed, but in some instances no coding is specified. The steps after coding, include radio frame equalization, a first interleaving, radio frame segmentation and rate matching. The radio frame segmentation divides the data over the number of frames in the specified TTI. The rate matching function operates by means of bit repetition or puncturing and defines the number of bits for each processed TrCH which are thereafter multiplexed to form a CCTrCH data stream.

[0008] The processing of the CCTrCH data stream includes bit scrambling, physical channel segmentation, a second interleaving and mapping onto one or more physical channels. The number of physical channels corresponds to the physical channel segmentation. For uplink transmissions, UE to Node B, the maximum number of physical channels for transmission of a CCTrCH is currently specified as two. For downlink transmissions, Node B to UEs, the maximum number of physical channels for transmission of a CCTrCH is currently specified as sixteen. Each physical channel data stream is then spread with a channelization code and modulated for over air transmission on an assigned frequency.

[0009] In the reception/decoding of the TrCH data, the processing is essentially reversed by the receiving station. Accordingly, UE and Node B physical reception of TrCHs require knowledge of TrCH processing parameters to reconstruct the TBS data. For each TrCH, a Transport Format Set (TFS) is specified containing a predetermined number of Transport Formats (TF). Each TF specifies a variety of dynamic parameters, including TB and TBS sizes, and a variety of semi static parameters, including TTI, coding type, coding rate, rate matching parameter and CRC length. The predefined collection of TFSs for the TrCHs of a CCTrCH for a particular frame is denoted as a Transport Format Combination (TFC).

[0010] Receiving station processing is facilitated by the transmission of a Transport Format Combination Indicator (TFCI) for a CCTrCH. 3GPP optionally

Ex.1002 / Page 171 of 744
Sandisk Technologies, Inc. et al

provides for "blind transport format detection" by the receiving station, in which case the receiving station considers the potential valid TFCIs. Where there is only one valid TFCI, that TFCI is used in either case.

[0011] In 3GPP, time slot transmissions are made in predefined bursts where the transmitted physical channel data is divided into a beginning time slot portion and an ending time slot portion. A selected midamble is included between the two physical channel data portions. The TFCI is currently specified as transmitted in two parts on either side of the midamble and also between the two physical channel data portions. Two examples from 3GPP TR 25.944 V3.5.0 are illustrated in Figures 3 and 4 respectively where the block labeled MA represents the midamble and the block labeled T represents the parts of the TFCI. In Figure 4, the CCTrCH is mapped to two physical channels, but only one physical channel includes the TFCI.

[0012] Figures 5A, 5B and 6 are diagrams of channel coding and multiplexing examples in accordance with 3GPP Frequency Division Duplex (FDD) specifications.

[0013] The various encoding steps implemented in communication systems play a key role in both performance and capacity of a wireless telecommunications systems. In particular, the turbo coding of the data being processed for wireless transmission plays a significant role for both TDD and FDD communications in a 3GPP system.

[0014] Turbo coding principles have found widespread applications in information theory and, in part, have entered into mainstream telecommunication theory and practice. These principles are used in error control, detection, interference suppression, equalization, and other communications-related areas. Turbo codes are a form of parallel, recursive systematic convolution codes, which can be used for channel coding and decoding in order to detect and correct errors that may occur in the transmission of digital data through different channels. Turbo codes are particularly useful for data transfers in which, under certain conditions, the data transfer rate can approach the theoretical limits of Shannon's Law. These favorable conditions generally

Ex.1002 / Page 172 of 744
Sandisk Technologies, Inc. et al

include large block size transfers, which are particularly useful in mobile communications.

[0015] Figure 7 shows how particular codes created by puncturing exhibit poorer than expected performance. As much as a 4dB loss can be measured in the critical Signal-to-Noise Ratio (SNR) value.

[0016] Typically, high-rate Turbo codes are created by puncturing (i.e., eliminating) parity bits of a lower-rate code until the desired code rate is attained. Particular codes created by puncturing exhibit poorer than expected performance due to certain punctured bit patterns. Because the recursive encoding blocks in a Turbo encoder have an infinite impulse response, information about each systematic bit is distributed across many of the parity bits.

[0017] In any error control coding scheme that employs puncturing, the location of the punctured bits will have an impact on performance. This is because puncturing strings of successive bits can be viewed as burst errors by the decoder. All error control mechanisms can only correct burst errors that are less than some threshold. Therefore, it is important when designing the puncturing scheme that the burst error correcting abilities of the code be considered. In the absence of specific knowledge of the error control coding scheme, a good rule is to minimize the maximum string of punctured bits. For high coding rates this corresponds to distributing the non-punctured bits uniformly across the transmitted block. Puncturing schemes that uniformly distribute the non-punctured bits can deleteriously interact with certain error control codes, causing a degradation of decoder performance.

[0018] Because the recursive encoding blocks in the Turbo encoder have infinite impulse response, information about each systematic bit is distributed across many of the parity bits. The impact of certain periodic non-puncture patterns in the parity bits can be qualitatively illustrated by considering how much information remains in the surviving parity bits.

[0019] Regions where performance of the punctured Turbo code is degraded can be determined by measuring how close the puncturing pattern is to being

periodic with period related to the period of the semi-periodic impulse response of the recursive encoding blocks of the Turbo encoder.

[0020] For a 3GPP Turbo encoder, the constituent recursive encoders have an impulse response that is periodic with a period of 7 symbols for positive time (semi-periodic). This can readily be seen by interpretation of the encoder as an M-sequence generator, i.e., if the shift register starts in the zero state and a one is applied only at $t = 0$, then the encoder is exactly a Linear Feedback Shift Register (LFSR) with an associated 3rd order primitive polynomial and initial state {1 0 0}, thus a period $2^3-1 = 7$ M-sequence is generated for positive time.

[0021] The encoder is a Linear Time Invariant (LTI) system on a binary field. Therefore, the output of the encoder is a sum of shifted versions of the impulse response of the encoder, each shifted impulse response corresponding to the locations of the 1's in the TB. If we consider an input that is all 0's for $t > T$, then the output of the constituent recursive encoding block will also be semi-periodic for $t > T$.

[0022] Certain non-puncturing periods may result in a loss of distributed information about a small group of bits across the set of parity bits. For example, let the TB consist of 7 bits followed by many zeros. Now consider the problem of trying to determine the seven bits by observation of the output of one encoder after heavy puncturing. Each observation is made at the output is a sum of a subset of the 7 bits, the subset being determined by the non-puncture period and the impulse response of the recursive encoding block. For most non-puncture patterns (those that have a period relatively prime with the period of the semi-periodic impulse response), sums over 7 different subsets can eventually be observed. Therefore, a system of 7 equations with 7 unknowns can be formed. Assuming they are linearly independent, the values of the 7 bits may be determined. However, consider the case where the non-puncture period is simple (one surviving parity bit per period) and the same as the period of the recursive encoding block, 7. Then every observation for $t > 6$ is the sum of the same subset of the 7 bits and so no new information is obtained, i.e., a rank-1 system of equations is formed and a unique solution does not exist. This is due to the

periodic nature of the signal. When the non-puncture period is relatively prime with the period of the recursive encoding block, a series of observations of the parity bits will eventually form a rank-7 system of equations. For our 7-bit group, this is sufficient to uniquely determine the bits, i.e., information sufficient to determine the bits is distributed across the surviving parity bits. When the periods are not relatively prime, the rank of the system becomes L/K where L is the period of the recursive encoding block and K is the Greatest Common Divider (GCD) of L and N where N is the period of the non-puncture pattern.

[0023] In the above illustration, a long sequence of zeros after the small group of bits was supposed for clarity sake. However, allowing additional groups of bits to follow the first group does not add to the distributed information about earlier groups. This can be seen from the causality of the system.

[0024] It is also possible that some distributed information is lost even for non-simple periods, e.g., consider the case where the non-puncture pattern alternates between a simple period of three and four. This results in a period 7 non-puncture pattern but with two surviving parity bits per period. The same argument above leads to a rank-2 system of seven equations which is still insufficient to uniquely determine the group of bits, but does reduce the dimension of the space they may span. Clearly, some information about the small group of bits is lost for some non-punctured patterns.

[0025] In order to obtain adequate performance, it is necessary to detect, avoid and/or correct the problematic puncturing patterns in the parity bit streams without having to avoid code rates. Typically, problematic interactions occur between the Turbo coder output and puncturing in the rate matching stages.

[0026] It would be desirable to have a method for creating high-rate Turbo codes that avoids problematic interactions by modifying the puncturing patterns.

[0027] SUMMARY OF THE INVENTION

[0028] According to the present invention, degradations in quality of punctured error correction coded transmissions are identified and the code

transmissions are modified accordingly. A puncturing pattern which approximates a particular code rate is identified, and a value for anticipated degradation is adjusted in accordance with a matching of the puncturing pattern and the particular code rate.

[0029] For both FDD and TDD, two stages of rate matching and two puncturing operations are included, and this is done because of a certain technique called, incremental redundancy, which allows for a wireless transmit/receive unit (WTRU) to receive and soft combine multiple transmissions of the same TB from the base station. If the transmission fails the first time, the signal is retransmitted with more data in an attempt to obtain a successful communication the second time. In order to do that, two different stages of puncturing are used because the WTRU has certain capabilities, certain buffer sizes that it can support, and in order to be able to fit the retransmissions that are combined into the buffer, there are two stages of puncturing. The first stage of rate matching punctures enough bits such that the surviving bits fit into a buffer, and the second stage of puncturing (or repetition) is to achieve the desired overall code rate. When the two stages of rate matching are combined, a plot of problematic regions of coding rate is a multidimensional plot. Thus, there are two rates now involved, which are a rate in a first stage of puncturing and a rate in a second stage of puncturing.

[0030] According to a first embodiment of the present invention, P1/P2 puncturing biasing for a single stage of rate matching is implemented using the punctured error correction coded transmissions.

[0031] According to a second embodiment, two or more stages may be employed at each stage of rate matching for detecting and correcting performance in punctured Turbo codes on a High Speed Downlink Shared Channel (HS-DSCH) in R5 High Speed Downlink Packet Access (HSDPA).

[0032] BRIEF DESCRIPTION OF THE DRAWINGS

[0033] A more detailed understanding of the invention may be had from the following description of a preferred embodiment, to be understood in conjunction

with the accompanying drawing wherein:

[0034] Figure 1 is a schematic diagram of a typical CDMA system in accordance with current 3GPP specifications;

[0035] Figure 2 is a processing diagram of TrCH data for a CCTrCH in accordance with 3GPP TDD specifications;

[0036] Figures 3 and 4 are diagrams of channel coding and multiplexing examples in accordance with 3GPP TDD specifications;

[0037] Figures 5A, 5B and 6 are diagrams of channel coding and multiplexing examples in accordance with 3GPP FDD specifications;

[0038] Figure 7 shows typical results of particular codes being created by puncturing, where as much as 4 dB loss can be measured in critical SNR value;

[0039] Figure 8 graphically depicts the SNR required to successfully decode a data block (critical SNR) with high probability vs. Code Rate when P1/P2 biasing is applied;

[0040] Figure 9 is a plot of the problematic regions with respect to an ordered pair (r1,r2) which are the rates of puncturing the parity streams in the first and second stages of rate matching, respectively;

[0041] Figure 10 is a schematic block diagram showing a circuit for 3GPP rate matching for HSDPA, using a Turbo coded HS-DSCH; and

[0042] Figure 11 graphically depicts initial Block Error Rate (BLER) for HS-DSCH transport blocks with certain effective coding rates.

[0043] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0044] In accordance with the present invention, several methods and embodiments are disclosed for detecting, avoiding and/or correcting the problematic puncturing patterns in the parity bit streams without having to avoid code rates.

[0045] The present invention is potentially applicable to layers 1 and 2 in a FDD, CDMA 2000 and TDD Universal Mobile Telecommunications System (UMTS) modes of transmission as well as other modes of transmission.

Ex.1002 / Page 177 of 744
Sandisk Technologies, Inc. et al

Additionally, methods are described hereinafter for modifying the puncturing patterns such that performance of degraded Turbo code is restored.

[0046] In accordance with the present invention, the methods disclosed herein may be implemented in a wireless transmit/receive unit (WTRU) and/or a base station. Hereinafter, a WTRU includes but is not limited to a UE, mobile station, fixed or mobile subscriber unit, pager, or any other type of device capable of operating in a wireless environment. When referred to hereafter, a base station includes but is not limited to a base station, Node-B, site controller, access point or other interfacing deice in a wireless environment.

[0047] Figure 8 graphically depicts the SNR required to successfully decode a data block (critical SNR) with high probability versus Code Rate when biasing of parity bits P1/P2 is applied. As shown in Figure 7, it has been found that particular codes created by puncturing the rate 1/3 3GPP Turbo code exhibit poorer than expected performance, as can be seen from the peaks in the figure.

[0048] Figure 9 is a plot of the problematic regions with respect to an ordered pair (r1,r2) which are the rates of puncturing the parity streams in the first and second stages of rate matching, respectively. As can be seen, there are particular regions where the rates of puncturing of the parity streams are such that some combinations of rates are particularly problematic.

[0049] In one embodiment, puncture biasing for parity bits P1 and P2 is implemented for one stage rate matching. Figure 10 is a schematic block diagram showing circuitry 600 for 3GPP rate matching for HSDPA, using a Turbo coded HS-DSCH. The circuitry 600 implements a 3GPP rate-matching scheme for HSDPA. The circuitry 600 includes a bit separation circuit 605, a first rate matching stage 610, a virtual incremental redundancy (IR) buffer 615, a second rate matching stage 620 and a bit collection circuit 625. The first rate matching stage 605 includes a Parity 1 (P1) bit rate matching circuit 630 and Parity 2 (P2) bit rate matching circuit 635. The second rate matching stage 620 includes a systematic bits rate matching circuit 640, a Parity 1 (P1) bit matching circuit 645 for the second rate matching and a Parity 2 (P2) bit matching circuit 650 for the second rate matching. In operation, the systematic bits, Parity 1 (P1)

Ex.1002 / Page 178 of 744
Sandisk Technologies, Inc. et al

and Parity 2 (P2) bits are processed through the first rate matching stage 610, virtual IR buffer 615, second rate matching stage 620 and bit collection circuit 625. Note that the Parity 1 (P1) and Parity 2 (P2) bits are treated separately. The systematic bits, Parity 1 (P1) and Parity 2 (P2) bits are combined at the bit collection circuit 625 to provide a single data output N_{data}. Also note that if the number of coded bits is less than or equal to the size of the virtual IR buffer 615, the first stage of rate matching 630 is transparent. The transparent first stage case and Rel-4 rate matching are considered.

[0050] In its actual implementation, it is anticipated that the different circuit functions will be executed by a common circuit or circuits, such as microprocessor circuitry, and in accordance with program instructions. The specific circuit functions used to implement the invention would therefore be a matter of choice of those configuring the circuitry within the rate matching circuitry 600.

[0051] The rate matching algorithm performs puncturing if the overall code rate is greater than 1/3 and repetition if the code rate is less than 1/3. Currently code rates greater than 1/3 are accomplished by applying the same puncturing rate to both P1 and P2 bits (to within one bit) but with different puncturing pattern phases. In order to avoid non-puncturing periodicities that have been shown to degrade the Turbo code performance, the puncturing rates for P1 and P2 be independently biased. For example, if the number of P1 bits is decreased by and the number of P2 bits is increased by, the overall code rate is unchanged but the problematic non-puncturing periods can be avoided. Given this approach to avoiding the problematic code rates, an analytical expression for the requisite bias, has been derived.

[0052] A simple and efficient method of modifying the puncture patterns is biasing the number of punctured bits in P1 and P2 by adding additional puncturing to one and removing it from the other. The amount of the bias should be no more than is needed to avoid certain periodicities. In biasing the puncturing rates of P1 and P2 bits, two constraints must be met.

[0053] When non-puncturing patterns with period $7\hat{N}/2$ ($\hat{N} = 1, 2, 3...$) are used, degradation in performance results. These periods will be employed whenever the average non-puncturing period of P1 or P2 bits is within ±1 or ±½ of $7\hat{N}/2$ for even and odd $\hat{N}$ respectively.

[0054] The average non-puncturing rate of the P1 and P2 rate matching blocks is $I/(P/2)$ where I is the number of bits at the input to each branch of rate matching and P is the total number of parity bits (P1 and P2) at the output of rate matching. Therefore, the desired code rate will lead to a problematic puncturing pattern if,

$$\left|\frac{I}{(P/2)} - \frac{7\hat{N}}{2}\right| < 1 - \frac{\hat{N}}{2} + \left\lfloor\frac{\hat{N}}{2}\right\rfloor,$$

Equation 1

for $\hat{N} > 0$ where,

$$\hat{N} = \left\lfloor\frac{4I}{7P} + \frac{1}{2}\right\rfloor.$$

Equation 2

[0055] In biasing the puncturing rates of the parity bit streams 1 and 2, P1 and P2, two constraints must be met.

[0056] First, the number of surviving bits in parity bit stream 1 must be increased sufficiently by an amount Δ to avoid the problematic puncturing patterns. Second, the number of surviving bits in parity bit stream 2 must be decreased by an equal amount Δ that is also sufficient to avoid problematic puncturing patterns.

[0057] These two constraints can be combined in to the following expression for Δ, where:

[0058] $$\Delta = \left\lceil \max\left\{ \left\lfloor \frac{I}{\left\lfloor \frac{7\hat{N}-1}{2} \right\rfloor} - \frac{P}{2} \right\rfloor, \left\lfloor \frac{P}{2} - \frac{I}{\left\lceil \frac{7\hat{N}+1}{2} \right\rceil} \right\rfloor \right\} \right\rceil$$

Equation 3

[0059] Alternatively, non-punctured bits may be increased in P2 and decreased in P1. The procedure to determine and apply the necessary P1/P2 bias is as follows:

Step 1) Compute $\hat{N}$ using Equation 2.

Step 2) Determine if the desired code rate results in a problematic puncturing pattern using Equation 1. If it does, continue to Step 3, otherwise set $\Delta = 0$ (if $\left\lfloor \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right\rfloor \geq 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$).

Step 3) Compute the bias Δ using Equation 3.

Step 4) Compute rate matching parameters by modifying Table

in TS25.212/222.

[0060] In order to avoid puncturing patterns that degrade Turbo code performance, the amount of puncturing applied to parity bit streams 1 and 2 respectively could be slightly different, while keeping the overall number of punctured bits constant.

[0061] If the number of punctured bits in parity bit stream 1 is decreased by Δ and accordingly, the number punctured bits in parity bit stream 2 bits is increased by Δ, the overall code rate is unchanged, but the problematic puncturing patterns can be avoided. Note that bit collection and bit de-collection stages need to be adjusted correspondingly for this case. Since individual regions of degraded performance tend to be small, it may be advantageous to simply avoid the use of code rates with degraded performance. Since code rates with

degraded performance can now be identified using the above equation, it is a simple matter to avoid them.

[0062] For Hybrid Automatic Repeat reQuest (HARQ) as deployed in 3GPP, it may be necessary to employ two stages of rate matching. This occurs whenever incremental redundancy (IR) is employed and the number of bits input to the rate-matching algorithm is larger than the virtual IR buffer. Note that when Chase combining HARQ is used, there is no need for a special IR buffer; a received bit buffer would be sufficient. (If constellation rearrangement were not used, a buffer of received symbols would be sufficient.) If it is determined that IR provides only a small improvement over Chase combining, then removal of IR from the standards may be advisable.

[0063] In order to obtain higher data rates, the 3GPP has introduced the HSDPA option wherein link adaptation is performed. One component of the link adaptation is Adaptive Modulation and Coding (AMC). With AMC, a channel quality estimate is used to select the modulation type and code rate that attains the largest throughput. When the channel quality is high, QAM modulation and a high code rate are selected. High code rates are obtained by puncturing a rate 1/3 Turbo code. During the design of the channel quality indication (CQI) table, it was observed that certain Transport Block Set Sizes (TBSS) exhibited poorer than expected performance. The CQI table is used to map a channel quality estimate into a recommended modulation and TBSS. This problem of poorer than expected performance is currently addressed by simply "hand tweaking" the CQI table to avoid the TBSSs that cause problems. In AMC, this leads to less than optimal throughput since optimal code rates will inevitably be avoided.

[0064] Support for link adaptation by means of AMC is an integral feature of R5 HSDPA. With AMC, modulation type and code rate on HS-DSCH can be varied according to channel conditions. Variable coding rates on HS-DSCH are achieved by means of rate matching (repetition or puncturing) output of the R99 Turbo-coder in a pre-determined manner, i.e. as a function of TBSS and available number of Physical Channel bits. Both R99/R4 and R5 use the same code rate 1/3 Turbo-coder. While the principle of rate-matching is the same for both R4

Ex.1002 / Page 182 of 744
Sandisk Technologies, Inc. et al

and R5, the likely and more frequent use of higher effective code rates (>1/2), the use of multiple stages of rate matching, and the standard allowing for the possibility of Incremental Redundancy (IR) differentiate R99/R4 and R5.

[0065] HSDPA performance requirements in WG4 have indicated unusually high and rather unpredictable SNR performance degradations for Turbo-coded and punctured transport blocks on HS-DSCH up to several dB for a range of effective coding rates above 1/2.

[0066] The Turbo-coding performance on HS-DSCH is heavily impacted by unfavorable puncturing patterns of the parity bit streams output by the baseline rate 1/3 Turbo-coder. Although the original WG4 simulation results were provided in a TDD-context, exactly the same degradations also occur in FDD, the reason being that Turbo-coding and rate-matching (TS25.212 and TS25.222) on HS-DSCH transport blocks (where the problematic interactions originate) are done exactly the same way for both modes.

[0067] The poorer than expected performance at some data rates is also a problem with a fixed modulation and coding scheme. If a user is assigned a code rate that exhibits the problem, the poor performance of the code will be compensated for by power control thus reducing cell capacity.

[0068] In another embodiment, adjustments to the parity stream puncturing allows the Medium Access Control – High Speed (MAC-hs) to simply select any desired transport block size as an outcome of the scheduler decision. It would be required to slightly change the generation of the puncturing patterns in the rate-matching stages in TS25.212/222. For example, one implementing modification to the current FDD standard as set for in TS 25.212 is appended hereto along with a brief discussion of why the modification is desirable.

[0069] For both the case of 1st rate-matching stage being transparent and the case of both 1st and 2nd rate-matching stage being employed, a number of acceptable solutions exist to avoid the problematic interactions by means of adjustments to the parity bit puncturing.

[0070] One of these solutions, e.g. a simple biasing method, is outlined below in the interest of providing a conceptual example for the case of the 1st rate-matching stage being transparent.

[0071] Currently, coding rates >1/3 are accomplished by applying the same amount of puncturing to both parity bit streams 1 and 2 at the output W of the Turbo-coder (Figure 10), although the puncturing pattern "phases" can be adjusted.

[0072] When a particular code rate is desired and it is determined that its use will result in degraded performance, a slightly lower code rate may be selected to avoid the degradation. Good choice for the substituted code rate would be the largest allowable code rate less than the desired code rate that has been determined to have an acceptable degradation.

[0073] If it is desirable for a system to be allowed to make unrestricted use of available code rates, then performance may be restored by modification of the puncture pattern. The key to restoring performance to the punctured Turbo code is to avoid the case where the surviving parity bits are overly redundant with respect to the systematic bits, i.e., modify the puncturing pattern to improve performance. One way to accomplish this is to sample the parity bits such that all the points corresponding to one complete period of the semi-periodic impulse response are sampled with as few samples as possible while maintaining the same code rate and not overly distorting the uniformity of the puncturing over the data block or creating overly large imbalances in the number of surviving parity bits in P1 and P2.

[0074] It should also be noted that this method of mitigation could also be applied to systems with more than 2 parity streams. For example, the 3GPP Turbo code (CDMA 2000) includes four parity streams. If a problematic puncturing pattern occurs it can be alleviated by adjusting the puncturing rates of each of the individual parity streams while maintaining a constant overall effective coding rate.

[0075] In the case of single stage of rate matching, as used in R99/R4 and in many R5 configurations, the problematic interactions between the Turbo-coded

output and the punctured transmitted sequence occurs in regions where the puncturing patterns are nearly periodic with surviving bit period equal to a multiple of 7, and when an average 1 out of every 3.5 parity bits survive.

[0076] These degradations correspond to rank-1 and rank-2 scenario respectively. This case corresponds to the rank-1 scenario. Rank-2 scenarios have less impact on the SNR degradation than rank-1 scenarios. Since higher rank scenarios (>2) do not significantly contribute to performance degradation, ranks other than rank-1 and rank-2 scenarios are not considered here.

[0077] The code rates at the center of these regions are determined by:

[0078] $$CR = \frac{1}{1+\frac{2}{3.5N}} = \frac{3.5N}{3.5N+2} \qquad N = 1,2,...$$

Equation 4

[0079] The critical code rate, *CR,* corresponds to a rank-1 scenario when N is even and rank-2 when N is odd. For rank-1, we have critical code rates CR = 7/9, 7/8, 21/23,... and for rank-2, we have CR = 7/11, 21/25,....

[0080] The ratio of the number of systematic bits and the number of parity bits in either parity bit stream 1 or 2 can be considered as "parity puncturing rate". When surviving bit patterns with parity puncturing rate close or equal to $7N/2$ (N =1,2,3...) are used, degradation in performance results. Some amount of Turbo-coding and puncturing interaction will also occur whenever the average periodicity of surviving bits of the parity bit streams 1 or 2 is within ±1 or ±½ of $7N/2$ for even and odd N respectively.

[0081] When two (or more) stages of rate matching are employed, problematic puncturing patterns can be generated by any stage, or by the interaction between the stages of rate matching.

[0082] A closed form expression(s) has been developed for the Release 5 3GPP Turbo coder / Rate Matching that predicts the performance degradation. The expression takes as input either the triplet {Bits Before 1st stage of rate Matching, Max Bits after 1st Rate Matching, Bits after 2nd Rate matching} or the pair {Rate of 1st stage Rate Matching with respect to parity bits, Rate of 2nd stage

Ex.1002 / Page 185 of 744
Sandisk Technologies, Inc. et al

Rate Matching with respect to parity bits}. The output, as depicted in Equation 5 below, is a rational number that may be compared to a threshold to accept/reject the configuration as providing/not providing acceptable performance. The parameters *r1* and *r2* represent the rates associated with the 1st and 2nd stages of rate matching with respect to the parity bit streams. Although complicated, the Equation 5 is a piecewise linear and continuous function in r1, r2 and is therefore trivial to evaluate.

$$\lambda(r1,r2)=\sum_{n=1}^{L-1}\alpha_n\left[\left(1-\left\lceil\frac{1}{L}\lceil r1\lceil nr2\rceil\rceil-\left\lfloor\frac{1}{L}\lceil r1\lceil nr2\rceil\rceil\right\rfloor\right\rceil\right)(nr2-\lfloor r2\rfloor)(r1\lceil nr2\rceil-\lfloor r1\lceil nr2\rceil\rfloor)+\right.$$
$$\left(1-\left\lceil\frac{1}{L}\lfloor r1\lceil nr2\rceil\rfloor-\left\lfloor\frac{1}{L}\lfloor r1\lceil nr2\rceil\rfloor\right\rfloor\right\rceil\right)(nr2-\lfloor nr2\rfloor)(-r1\lceil nr2\rceil-\lfloor r1\lceil nr2\rceil\rfloor+1)+$$
$$\left(1-\left\lceil\frac{1}{L}\lceil r1\lfloor nr2\rfloor\rceil-\left\lfloor\frac{1}{L}\lceil r1\lfloor nr2\rfloor\rceil\right\rfloor\right\rceil\right)(-nr2+\lfloor nr2\rfloor+1)(r1\lfloor nr2\rfloor-\lfloor r1\lfloor nr2\rfloor\rfloor)+$$
$$\left.\left(1-\left\lceil\frac{1}{L}\lfloor r1\lfloor nr2\rfloor\rfloor-\left\lfloor\frac{1}{L}\lfloor r1\lfloor nr2\rfloor\rfloor\right\rfloor\right\rceil\right)(-nr2+\lfloor nr2\rfloor+1)(-r1\lfloor nr2\rfloor+\lfloor r1\lfloor nr2\rfloor\rfloor+1)\right]$$

Equation 5

[0083] A threshold comparison could then be based on this equation to test whether or not the Code Rate pair will perform adequately. Equivalently, the triple: {Data Block Size, Size After 1st Stage Rate-Matching, Size After 2nd Stage Rate-Matching} can be used since this triple maps directly into the Code Rate pair.

[0084] The effect of the periodicity of the puncturing pattern can also be mitigated by interleaving the parity bits before rate-matching occurs. In this way, after de-interleaving, periodic sampling of the parity bits is avoided even when periodic sampling is done in rate-matching.

[0085] The functionality of the channel interleaver can be incorporated into the rate-matching block interleaver. This can be accomplished by adding additional constraints to the rate-matching block Interleaver (those defined by the channel interleaver) when the interleaver is designed. This however will require an additional interleaver for the systematic bits.

Ex.1002 / Page 186 of 744
Sandisk Technologies, Inc. et al

[0086] Note that the rearrangement of the non-punctured bits only needs to occur on a range equal to the period of the semi-periodic impulse response of the recursive encoding blocks of the Turbo encoder. Therefore, small time-varying interleavers can be added to the parity bit steams to prevent problematic puncturing patterns. This has the advantage of requiring less memory and helps to minimize the distortion of the uniformity of the puncturing.

[0087] Certain parameters may be made to vary as a function of time (or bit index) to generate puncturing patterns that will mitigate performance degradations. One such example for 3GPP includes modification of the rate-matching parameters X_i, e_ini, e_minus and e_plus parameters throughout the execution of the rate-matching. In this way, the parameters affecting the puncturing pattern may change at one or more locations within the block of bits being punctured. By allowing the parameters to change, the instantaneous code rates may be different from the desired overall code rate (thereby avoiding the puncturing interaction) while still maintaining the same overall code rate. For example: 1) e_plus and e_minus may change frequently throughout the puncturing algorithm thus breaking up any possible long runs of deleterious puncturing. 2) Parameters may be changed so that the code rate is changed so that it is higher in some portions of the code block and lower in others.

[0088] The above methods are examples of modifications to the puncturing pattern used to generate specific code rates. All such methods may be considered special cases and specific implementations of the more general concept of a permutation of the puncturing pattern. Any method of changing the puncturing pattern, i.e., generating some permutation of the original puncturing pattern, has the potential of changing the overall performance of a code.

[0089] One can achieve the desired overall code rate while avoiding problematic puncturing patterns by adjusting the amount of puncturing performed in each stage. This method, when placed in the context of the 3GPP HSDPA Turbo Code / Rate Matching algorithm leads to artificially reducing the size of the virtual Incremental Redundancy (IR) buffer. In general the approach

Ex.1002 / Page 187 of 744
Sandisk Technologies, Inc. et al

implies a modification to the individual puncturing rates of each of the stages such that a desirable overall puncturing pattern results.

[0090] Stage Rate Biasing is easily accomplished by changing the operating point (the *r1*, *r2* pair) to some point on a curve of constant code rate (i.e., where the product *r1*r2* is unchanged) where performance degradation is tolerable.

[0091] The combination of adjusting the stage puncturing rates as well as the relative rates of puncturing in the two parity streams can also mitigate the problematic interaction between the puncturing pattern and the Turbo Code performance.

[0092] This can be envisioned as splitting the operating point in to two points. One of the points corresponds to the rates of 1st and 2nd stage rate matching for the first parity bit stream and the second point for the second parity bit stream. The two points should be selected so that the overall code rate is unchanged, both points lie in regions of acceptable performance, and the total number of surviving bits after the first stage of rate matching does not exceed imposed limits such as an IR buffer.

[0093] The present invention provides several simple and efficient techniques of detecting, avoiding and/or correcting problematic puncturing patterns in the parity bit streams without having to avoid desired code rates. According to the present invention, degradations in punctured error correction coded transmissions are reduced by identifying a puncturing pattern which approximates a particular code rate, and adjusting a value for anticipated degradation in accordance with the matching of the puncturing pattern and the particular code rate.

[0094] When creating high rate codes via puncturing, it is generally desirable to distribute, as uniformly as possible, the bits that are punctured. Using this guideline, it is inevitable that certain code rates will require the resulting non-punctured bit pattern to be less effective. One such example occurs when the non-punctured bit pattern is periodic, with a period equal to the period of the semi-periodic impulse response of the recursive encoding blocks of the

Ex.1002 / Page 188 of 744
Sandisk Technologies, Inc. et al

Turbo encoder. An algorithm may used to identify all puncturing patterns with degraded performance.

[0095] In a particular embodiment of the invention, the capacity of a WTRU is determined, including buffer sizes that are supported by the WTRU. The puncturing is used to reduce the number of bits to fit into the buffer, and the overall code rate is adjusted so as to provide sufficient error correction capability. This provides a first rate in a first stage of puncturing and a second rate in a second stage of puncturing.

[0096] According to the present invention, detecting, avoiding and/or correcting problematic puncture patterns in parity bit streams and punctured Turbo codes is achieved without having to avoid desirable code rates. This is applicable without limitation, to FDD, TDD and other modes of transmission, and enables identification/avoidance of regions of relatively poor Turbo code performance, which might result from the prior art bit-puncturing scheme. The invention provides for forward error correction comprising Turbo coding and puncturing that achieve a smooth functional relationship between any measure of performance and the effective coding rate resulting from combining the lower rate code generated by the Turbo encoder with puncturing of the parity bits. Examples of measures of performance would be BLER, bit error rate (BER), required signal-to-interference ratio (SIR), or required SNR.

[0097] The patterns of the puncturing are created by creating an encoded transmission at a particular data rate and excluding some of the encoded bits. At the receiver end, zeros or other "filler" bits are put in place of the punctured bits, and the decoding operation is based on the received signal with the missing bits. According to the present invention, the puncturing pattern is modified in order to eliminate degradation which would occur under particular puncturing patterns. In particular, if the puncturing pattern is regular or has a periodic nature, it is more likely to result in signal degradation. By eliminating the degradation, it is possible to achieve the desired code rate.

[0098] The present invention is particularly useful for 3GPP-coding, and can be used for low chip rate TDD, high chip rate TDD and FDD.

Ex.1002 / Page 189 of 744
Sandisk Technologies, Inc. et al

[0099] One such example of a periodicity occurs when the non-punctured bit pattern is periodic with period equal to the period of the semi-periodic impulse response of the recursive encoding blocks of the Turbo encoder. An algorithm is provided to identify all puncturing patterns with degraded performance. Additionally, modification of the puncturing patterns is performed such that performance of the Turbo code is restored.

[0100] Modifications to puncturing are used to restore Turbo performance. If it is desirable for a system to be allowed to make unrestricted use of available code rates, then performance may be restored by modification of the puncture pattern. Therefore if it is deemed desirable for a system to be allowed to make unrestricted use of available code rates, then performance may be restored by modification of the puncture pattern.

[0101] Adaptive coding means that on the fly, information is obtained from the WTRU that indicates the channel quality. Based on that, the transmitter adjusts its code rate amongst other parameters. In instances of poor reception, a substantial amount of redundancy is added, which results in a very low code rate, but which allows transfer of data through with the required block error probability.

[0102] The channel quality estimation procedure does not inherently take into account the possibility that problematic puncturing may have been used, nor should it. The channel quality is typically based on signal power and noise power, but it does not take into account the fact that the encoder may have been punctured in a problematic way, therefore AMC will not serve to avoid problematic code rates. With good channel quality, very little coding is needed and it is possible to use a higher order modulation. If the channel quality is poor, the code rate is reduced. Therefore, a reduced data rate is used to provide the desired error performance. The channel quality indicator is used to monitor the channel quality and in turn provides input to the data rate controller.

[0103] Capacity is measured by determining a combined data rate to all users supported. If one or more users in a cell are tuned to a problematic code rate, then those users are presumed to require more power and thereby reduce

capacity. In many cases the problematic coding rates are known or can be determined via this invention and therefore it is possible to avoid these rates.

[0104] The present invention provides an ability to identify problem rates in advance through some analytical means, and responding to this information by avoiding these rates either dynamically or statically. This can be achieved by detecting problematic patterns or by using a formula or table of rates to avoid. If a transmission calls for one of the problem rates, the actual transmission is provided at a rate which is slightly different. The result is, the problem rate is avoided.

[0105] These are representative examples of modifications to the puncturing pattern used to generate specific code rates, as applicable to layer 1 and layer 2 in the TDD and FDD mode of transmission in accordance with the present invention. Variations to the examples described are conceivable. All such variations may be considered special cases and specific implementations of the more general concept of a permutation of the puncturing pattern. Any method of changing the puncturing pattern, i.e., generating some permutation of the original puncturing pattern, has the potential of changing the overall performance of a code, and it is the intent to include permutations and methods of generating such permutations of the puncturing pattern envisaged in the present invention. All modifications and improvements to the basic concept of the invention as set forth hereinabove are envisaged to be within the ambit of the present invention.

[0106] Figure 11 shows the BLER performance of 3 different HS-DSCH transport blocks of increasing size 4554, 4705 and 4858 bits respectively. All of them are rate 1/3 Turbo-coded, and then punctured down to 6072 bits mapped to the High Speed Physical Downlink Shared Channels (HS-PDSCH). This results in an effective coding rate of 0.75 (labeled 7/9-), 0.77 (labeled 7/9) and 0.80 (labeled 7/9+) for the 3 increasing transport blocks sizes.

[0107] When increasing the effective coding rate (or equivalently increasing the puncturing rate) of a Turbo-coded transport block on HS-DSCH, one should expect that BLER performance degrades in a proportioned manner, i.e. gets

increasingly worse for increasingly higher effective coding rates. This unfortunately does not always happen, and it can be seen in this example that the rate 0.77 (7/9) coded transport block needs some 3dB more SNR than a rate 0.80 coded transport block to achieve the same BLER of approximately 10%.

[0108] The reason for the observed degradations is that the periodicity (period = 7) in the output of the two Turbo-coder parity bit streams, as given by the impulse responses of the constituent recursive convolutional encoders, deleteriously interacts with the periodicity of the puncturing patterns applied by the second and/or the first rate matching stages in many cases. This effect is increasingly more frequent for increasingly higher puncturing rates.

[0109] In the analyses that follow, a single R99 rate-matching block is assumed. However, the concepts apply more generally to R4 and R5 with and without incremental redundancy. The main conclusion for both cases is that the critical puncturing rates leading to SNR degradation are predictable in principle, but would require a rather complex set of parameters to be taken into account.

[0110] The setting of critical puncturing rates is avoided in the MAC-hs in the Node B, which has the particular advantage that it would not require a change in current standards and could leave details open to vendor-specific implementation. While this approach is preferable, the look-up tables used are complex and restrictions on possible MAC-hs scheduler decisions are imposed.

[0111] Alternatively, slight modifications are made to the puncturing patterns as generated by the current rate-matching stages for HS-DSCH in TS25.212/222. These modifications constitute only a small change in the way the rate matching parameters are generated in TS25.212/222, e.g., parameters for HARQ second rate matching, and more importantly do not require any additional signaling.

[0112] Due to the apparent simplicity of the second option (slight modification of the generation of the puncturing pattern on Turbo-coded parity bit streams) and its transparency for the MAC-hs scheduler (simply selects whatever transport block size is desired as outcome of the scheduler decision), a corresponding change is made in TS25.212/222 to remedy this problem.

Ex.1002 / Page 192 of 744
Sandisk Technologies, Inc. et al

[0113] The constituent recursive convolutional encoders in the Turbo-coder as by TS25.212/222 have infinite impulse responses that are periodic with period 7 for positive time. The impact of certain patterns of surviving (i.e. non-punctured) bits in the parity bit stream can be qualitatively illustrated by considering how much information remains in these surviving parity bits.

[0114] The Turbo-coder produces 3 bit streams, systematic bits corresponding to the input sequence, parity bit stream 1 (output of 1st constituent encoder) and parity bit stream 2 (output of 2nd constituent encoder).

[0115] Note that the constituent recursive convolutional encoder is an LTI system on GF_2. Therefore, the output of the encoder is a sum of shifted versions of the impulse response of the encoder. Each shifted version of the impulse response corresponds to the locations of the 1's in the TB.

[0116] Certain puncturing patterns may result in a loss of distributed information about a small group of bits across the set of bits in one of the parity bit streams. For example, let the TB consist of 7 bits followed by many zeros. Now consider the problem of trying to determine the seven bits by observation of the output of one encoder after heavy puncturing. Each observation made at the output is a sum of a subset of the 7 bits. The subset for each particular observation is determined by the surviving bit period and the impulse response of the recursive encoding block.

[0117] For most puncture patterns, sums over 7 different subsets of the 7 bits can eventually be observed. Therefore, a system of 7 equations with 7 unknowns can be formed. Assuming that they are linearly independent, the values of the 7 bits may be determined.

[0118] However, when the periodicity of surviving bits is simple (one surviving parity bit per period) and the same as the period of the recursive encoding block, 7, then, due to the periodicity of the signal, every observation for $t > 6$ is the sum of the same subset of the 7 bits and so no new information is obtained by subsequent observations. Thus, a rank-1 system of equations is formed and a unique solution does not exist.

[0119] When the periodicity of surviving bits is not a multiple of 7, making observations of the parity bits will eventually form a rank-7 system of equations after several observations. For our 7-bit group, this is sufficient to uniquely determine the bits, i.e. information sufficient to determine the bits is distributed across the surviving parity bits.

[0120] It is also possible that some distributed information is lost even for non-simple periods, e.g., consider the case where the non-puncture pattern alternates between a simple period of three and four (two surviving parity bits per period with period = 7).

[0121] This results in a period 7 non-puncture pattern but with two surviving parity bits per period. The same argument above leads to a rank-2 system of seven equations. This is still insufficient to uniquely determine the group of bits, but does reduce the dimension of the space they may span. Clearly, some information about the TB may also be lost for non-simple surviving bit patterns.

[0122] At least two alternative approaches exist to avoid the problematic interactions between Turbo-coder output and puncturing in the rate-matching stages.

[0123] The avoidance of problematic puncturing patterns is an approach that requires the MAC-hs to avoid using puncturing patterns that lead to SNR degradation and does not use certain combinations for mapping incoming transport blocks sizes to HS-PDSCHs.

[0124] Avoidance would either require building lookup tables that identify the problematical configurations or evaluating the predicted degradation expression described previously in the MAC-hs scheduler.

[0125] An advantage of the look-up table approach is that no change in current standards would be required and that any particular implementation could be left vendor-specific.

[0126] A disadvantage of the avoidance approach is that it is complicated by the fact that there are several dimensions to the problem that will all play a role in determining if a particular configuration will give rise to degradation.

Ex.1002 / Page 194 of 744
Sandisk Technologies, Inc. et al

Parameters the MAC-hs scheduler would need to take into account in addition include:

(1) transport block set size of HS-DSCH transmission;

(2) number of soft information bits to be stored at the WTRU for HARQ; and

(3) number of physical channel bits allocated for the HS-PDSCHs.

[0127] For FDD, these parameters are implied by the combination indicator for number of channelization codes and modulation format i ($1 < i < 29$), and the transport block size k_i, ($0 < k_i < 62$) in TS25.321.

[0128] Described below is one embodiment of the present invention that includes a combination of previously described techniques.

[0129] The hybrid ARQ functionality matches the number of bits at the output of the channel coder to the total number of bits of the HS-PDSCH set to which the HS-DSCH is mapped. The hybrid ARQ functionality is controlled by the redundancy version (RV) parameters. The exact set of bits at the output of the hybrid ARQ functionality depends on the number of input bits, the number of output bits, and the RV parameters.

[0130] The hybrid ARQ functionality consists of two rate-matching stages and a virtual buffer.

[0131] The first rate matching stage matches the number of input bits to the virtual IR buffer, information about which is provided by higher layers. Note that, if the number of input bits does not exceed the virtual IR buffering capability, the first rate-matching stage is transparent.

[0132] The second rate matching stage matches the number of bits after first rate matching stage to the number of physical channel bits available in the HS-PDSCH set in the TTI.

[0133] Definitions of notations used:

N^{TTI}: Number of bits in a transmission time interval before rate matching.

ΔN_i: An intermediate calculation variable.

$\Delta N_{i,l}^{TTI}$: If positive - number of bits to be repeated in each transmission time interval on TrCH i with transport format l.

If negative - number of bits to be punctured in each transmission time interval on TrCH i with transport format l.

ΔN_{PARITY} : Number of bits to adjust the length of parity streams by, in HARQ 1st stage rate-matching.

N_{data}: Total number of bits that are available for the HS-DSCH in a TTI.

e_{ini} Initial value of variable e in the rate matching pattern determination algorithm.

e_{plus} Increment of variable e in the rate matching pattern determination algorithm.

e_{minus} Decrement of variable e in the rate matching pattern determination algorithm.

b: Indicates systematic and parity bits

b=1: Systematic bit, x_k .

b=2: 1st parity bit (from the upper Turbo constituent encoder), z_k.

b=3: 2nd parity bit (from the lower Turbo constituent encoder), z'_k.

[0134] The HARQ bit separation function shall be performed in the same way as bit separation for turbo encoded TrCHs.

[0135] HARQ first stage rate matching for the HS-DSCH transport channel shall be performed using the method described below with the following specific parameters.

[0136] The maximum number of soft bits available in the virtual IR buffer is N_{IR} which is signaled from higher layers for each HARQ process. The number of coded bits in a TTI before rate matching is N^{TTI} this is deduced from information signaled from higher layers and parameters signaled on the High Speed Synchronization Control Channel (HS-SCCH) for each TTI. Note that HARQ processing and physical layer storage occurs independently for each HARQ process currently active.

Ex.1002 / Page 196 of 744
Sandisk Technologies, Inc. et al

[0137] If N_{IR} is greater than or equal to N^{TTI} (i.e. all coded bits of the corresponding TTI can be stored) the first rate matching stage shall be transparent. This can, for example, be achieved by setting $e_{minus} = 0$. Note that no repetition is performed.

[0138] If N_{IR} is smaller than N^{TTI} the parity bit streams are punctured by setting the rate matching parameter $\Delta N_{il}^{TTI} = N_{IR} - N^{TTI}$ where the subscripts i and l refer to transport channel and transport format in the referenced sub-clause. Note the negative value is expected when the rate matching implements puncturing. Bits selected for puncturing which appear as δ shall be discarded and not counted in the totals for the streams through the virtual IR buffer.

[0139] If first stage puncturing is to be performed, the procedure below shall be applied. Index b is used to indicate systematic (b=1), 1st parity (b=2), and 2nd parity bit (b=3). The parameter ΔN_{PARITY} : is the change in length to the parity streams to circumvent problematic puncturing rates.

The first stage rate matching delta is calculated as follows:

if puncturing is to be performed

$$X_i = \lfloor N^{TTI}/3 \rfloor; \qquad PR = \frac{\lfloor -\Delta N_{il}^{TTI}/2 \rfloor}{X_i}; \qquad \Delta N_{PARITY} = 0; \quad \delta_{PR} = 49; \qquad \alpha = 1$$

$$\text{while}(\alpha \le 4)$$

$$round = \frac{\left\lfloor 7 \cdot \alpha \cdot PR + \frac{1}{2} \right\rfloor}{7 \cdot \alpha}$$

$$\lambda = PR - round$$

$$\text{if}\left(|\lambda| > \frac{1}{\delta_{PR}} \right) or \left(PR > \frac{48}{49} \right) or \left(PR < \frac{1}{49} \right) \text{then}$$

$$\alpha = \alpha + 1$$

$$\delta_{PR} = 98$$

$$\text{else if}(\lambda > 0) \text{ then}$$

Ex.1002 / Page 197 of 744
Sandisk Technologies, Inc. et al

$$\Delta N_{PARITY} = \left\| \left\lfloor -\Delta N_{il}^{TTI} / 2 \right\rfloor - X_i \cdot \left(round - \frac{1}{\delta_{PR}} \right) \right\|$$

$\alpha = 5$

else

$$\Delta N_{PARITY} = \left\| \left\lfloor -\Delta N_{il}^{TTI} / 2 \right\rfloor - X_i \cdot \left(round + \frac{1}{\delta_{PR}} \right) \right\|$$

$\alpha = 5$

end if

end while

$$\Delta N_i = \begin{cases} \left\lfloor -\Delta N_{il}^{TTI} / 2 \right\rfloor - \Delta N_{PARITY}, b = 2 \\ \left\lceil -\Delta N_{il}^{TTI} / 2 \right\rceil + \Delta N_{PARITY}, b = 3 \end{cases}$$

a = 2 when b=2

a = 1 when b=3

[0140] If ΔN_i is calculated as 0 for b=2 or b=3, then the following procedure and the rate matching algorithm shall not be performed for the corresponding parity bit stream.

[0141] For each radio frame, the rate-matching pattern shall be calculated, where:

X_i is as above,

$e_{ini} = X_i$

e_{plus} = a×X_i

e_{minus} = a×$|\Delta N_i|$

[0142] HARQ second stage rate matching for the HS-DSCH transport channel shall be done using one of two possible methods.

[0143] If second stage puncturing is to be performed, and the Composite Puncturing Rate of one of the parity streams, calculated using the formula below falls in any of the intervals [91/128,92/128], [217/256, 222/256], [231/256,

Ex.1002 / Page 198 of 744
Sandisk Technologies, Inc. et al

232/256], [237/256, 238/256] or [487/512, 488/512], then puncturing of that parity stream shall be performed. Note, puncturing maybe carried out on one, both, or neither of the two parity streams, but never on the systematic parity stream.

[0144] Composite Puncturing Ratio, $CPR = \begin{cases} \dfrac{\lfloor N^{TTI}/3 \rfloor - N_{t,p1}}{N^{TTI}/3}, b = 2 \\ \dfrac{\lfloor N^{TTI}/3 \rfloor - N_{t,p2}}{N^{TTI}/3}, b = 3 \end{cases}$

[0145] Otherwise, second stage rate matching for the HS-DSCH transport channel shall be done with the following specific parameters. Bits selected for puncturing which appear as δ shall be discarded and are not counted in the streams towards the bit collection.

[0146] The parameters of the second rate matching stage depend on the value of the RV parameters s and r. The parameter s can take the value 0 or 1 to distinguish between transmissions that prioritize systematic bits (s = 1) and non systematic bits (s = 0). The parameter r (range 0 to r_{max}-1) changes the initial error variable e_{ini} in the case of puncturing. In case of repetition both parameters r and s change the initial error variable e_{ini}. The parameters X_i, e_{plus} and e_{minus} are calculated as per table 1 below.

[0147] Denote the number of bits before second rate matching as N_{sys} for the systematic bits, N_{p1} for the parity 1 bits, and N_{p2} for the parity 2 bits, respectively. Denote the number of physical channels used for the HS-DSCH by P. N_{data} is the number of bits available to the HS-DSCH in one TTI and defined as $N_{data}=P\times3\times N_{data1}$. The rate matching parameters are determined as follows.

[0148] For $N_{data} \le N_{sys} + N_{p1} + N_{p2}$, puncturing is performed in the second rate matching stage. The number of transmitted systematic bits in a transmission is $N_{t,sys} = \min\{N_{sys}, N_{data}\}$ for a transmission that prioritizes systematic bits and $N_{t,sys} = \max\{N_{data} - (N_{p1} + N_{p2}), 0\}$ for a transmission that prioritizes non systematic bits.

[0149] For $N_{data} > N_{sys} + N_{p1} + N_{p2}$ repetition is performed in the second rate matching stage. A similar repetition rate in all bit streams is achieved by setting the number of transmitted systematic bits to $N_{t,sys} = \left\lfloor N_{sys} \cdot \frac{N_{data}}{N_{sys} + 2N_{p1}} \right\rfloor$.

[0150] The number of parity bits in a transmission is: $N_{t,p1} = \max\left(\left\lfloor \frac{N_{data} - N_{t,sys}}{2} \right\rfloor - \Delta N_{PARITY}, 0 \right)$ and $N_{t,p2} = N_{data} - (N_{t,sys} + N_{t,p1})$ for the parity 1 and parity 2 bits, respectively.

	X_i	e_{plus}	e_{minus}
Systematic RM S	N_{sys}	N_{sys}	$\lvert N_{sys} - N_{t,sys} \rvert$
Parity 1 RM P1_2	N_{p1}	$2 \cdot N_{p1}$	$2 \cdot \lvert N_{p1} - N_{t,p1} \rvert$
Parity 2 RM P2_2	N_{p2}	N_{p2}	$\lvert N_{p2} - N_{t,p2} \rvert$

Table 1: Parameters for HARQ second rate matching

[0151] Table 1 above summarizes the resulting parameter choice for the second rate matching stage. The rate matching parameter e_{ini} is calculated for each bit stream according to the RV parameters r and s using
$e_{ini}(r) = \{(X_i - \lfloor r \cdot e_{plus} / r_{\max} \rfloor - 1) \bmod e_{plus}\} + 1$ in the case of puncturing , i.e., $N_{data} \le N_{sys} + N_{p1} + N_{p2}$, and
$e_{ini}(r) = \{(X_i - \lfloor (s + 2 \cdot r) \cdot e_{plus} / (2 \cdot r_{\max}) \rfloor - 1) \bmod e_{plus}\} + 1$ for repetition, i.e., $N_{data} > N_{sys} + N_{p1} + N_{p2}$. Where $r \in \{0,1,\text{L}\ , r_{\max} - 1\}$ and $r_{\max}$ is the total number of redundancy versions allowed by varying r. Note that r_{max} varies depending on the modulation mode, i.e. for 16QAM r_{max} = 2 and for QPSK r_{max} = 4.

[0152] Note: For the modulo operation the following clarification is used: the value of (x mod y) is strictly in the range of 0 to y-1 (i.e. -1 mod 10 = 9).

[0153] HARQ second stage rate matching for the HS-DSCH transport channel shall be done with the method described below, which separates each stream into segments with specific parameters calculated below. Bits selected for

Ex.1002 / Page 200 of 744
Sandisk Technologies, Inc. et al

puncturing which appear as δ shall be discarded and are not counted in the streams towards the bit collection.

[0154] The parity stream will be segmented into 3 segments, the first segment will be made up of the first X_{seg1} bits of the parity stream, the second segment will be made up of the next X_{seg2} bits, and the last segment will be made up of the remaining X_{seg3} bits.

[0155] The first segment is denoted by $x_{i,1}, x_{i,2}, \ldots x_{i,Xseg1}$

The second segment by $x_{i,Xseg1+1}, x_{i,Xseg1+2}, \ldots x_{i,Xseg2}$;

And the final segment by $x_{i,Xseg1+Xseg2+1}, x_{i,Xseg1+Xseg2+2}, \ldots x_{i,X_i}$

Where $X_{seg1} = 42 \cdot \lfloor X_i / 98 \rfloor, X_{seg2} = 56 \cdot \lfloor X_i / 98 \rfloor, X_{seg3} = X_i \bmod 98$

[0156] Note: If X_i is less than 98 only the third segment exists, and if X_i is a multiple of 98 then the third segment will be empty. For the case where a particular segment is empty, then of course no puncturing is performed on that non-existent segment.

[0157] Additional parameters for HARQ Second Rate Matching Stage for problematic puncturing rates

[0158] Parameters P, N_{data}, N_{sys}, N_{p1}, N_{p2}, $N_{p,t1}$, and $N_{p,t2}$ are calculated, additional parameters are defined below.

[0159] Denote the number of parity 1 bits before second rate matching after segmentation as $N_{p1,seg1}$, $N_{p1,seg2}$, $N_{p1,seg3}$ for the parity 1 bits in the 1st, 2nd and 3rd segment respectively. Denote the number of parity 2 bits before second rate matching after segmentation as $N_{p2,seg1}$, $N_{p2,seg2}$, $N_{p2,seg3}$ for the parity 1 bits in the 1st, 2nd and 3rd segment respectively. The rate matching parameters are determined as follows.

[0160] For $N_{data} \le N_{sys} + N_{p1} + N_{p2}$, puncturing is performed in the second rate matching stage.

[0161] The number of parity bits after segmentation is

$$N_{pb,seg1} = 42 \cdot \lfloor N_{pb} / 98 \rfloor$$
$$N_{pb,seg2} = 56 \cdot \lfloor N_{pb} / 98 \rfloor$$

Ex.1002 / Page 201 of 744
Sandisk Technologies, Inc. et al

$$N_{pb,seg3} = N_{pb} \bmod 98$$

[0162] The number of parity bits in each segment for parity 1 bits (b=2), and parity 2bits (b=3) is:

$$N_{t,pb,seg1} = \left\lfloor \frac{49 \cdot N_{t,pb} \cdot \lfloor N_{pb}/98 \rfloor}{N_{pb}} + \frac{1}{2} \right\rfloor;$$

$$N_{t,pb,seg2} = \left\lfloor \frac{49 \cdot N_{t,pb} \cdot \lfloor N_{pb}/98 \rfloor}{N_{pb}} + \frac{1}{2} \right\rfloor;$$

$$N_{t,pb,seg3} = N_{t,pb} - \left(N_{t,pb,seg1} + N_{t,pb,seg2}\right)$$

[0163] The parameters X_i, e_{plus} and e_{minus} are calculated as per table 2 below.

Stream	Segment	X_i	e_{plus}	e_{minus}
Systematic RM S		N_{sys}	N_{sys}	$\lvert N_{sys} - N_{t,sys} \rvert$
Parity 1 RM P1_2	1st	$N_{p1,seg1}$	$2 \cdot N_{p1,seg1}$	$2 \cdot \lvert N_{p1,seg1} - N_{t,p1,seg1} \rvert$
	2nd	$N_{p1,seg2}$	$2 \cdot N_{p1,seg2}$	$2 \cdot \lvert N_{p1,seg2} - N_{t,p1,seg2} \rvert$
	3rd	$N_{p1,seg3}$	$2 \cdot N_{p1,seg3}$	$2 \cdot \lvert N_{p1,seg3} - N_{t,p1,seg3} \rvert$
Parity 2 RM P2_2	1st	$N_{p2,seg1}$	$N_{p2,seg1}$	$\lvert N_{p2,seg1} - N_{t,p2,seg1} \rvert$
	2nd	$N_{p2,seg2}$	$N_{p2,seg2}$	$\lvert N_{p2,seg2} - N_{t,p2,seg2} \rvert$
	3rd	$N_{p2,seg3}$	$N_{p2,seg3}$	$\lvert N_{p2,seg3} - N_{t,p2,seg3} \rvert$

Table 2: Parameters for dithered HARQ second rate matching

[0164] The rate matching parameter e_{ini} is calculated for each segment of bit stream according to the RV parameters r and s using $e_{ini}(r) = \{(X_i - \lfloor r \cdot e_{plus}/r_{\max} \rfloor - 1) \bmod e_{plus}\} + 1$ in the case of puncturing. Where $r \in \{0,1,\mathrm{L}\ ,r_{\max}-1\}$ and $r_{\max}$ is the total number of redundancy versions allowed by

Ex.1002 / Page 202 of 744
Sandisk Technologies, Inc. et al

varying r. Note that r_{max} varies depending on the modulation mode, i.e. for 16QAM $r_{max} = 2$ and for QPSK $r_{max} = 4$.

[0165] Note: For the modulo operation the following clarification is used: the value of (x mod y) is strictly in the range of 0 to y-1 (i.e. -1 mod 10 = 9). The rate matching algorithm is called for each segment of the parity stream in turn.

[0166]

[0167] For both parity streams, after the rate matching algorithm has been called for each of the three segments, the three punctured segments should be concatenated together in their original order.

[0168] The re-concatenated bit stream is denoted by: $x_{o,seg1,1}, x_{o,seg1,2}, \ldots x_{o,seg1,N_{t,p1,seg1}}, x_{o,seg2,1}, x_{o,seg2,2}, \ldots x_{o,seg2,N_{t,p1,seg2}}, x_{o,seg3,1}, x_{o,seg3,2}, \ldots x_{o,seg3,N_{t,p1,seg3}}$

[0169] The HARQ bit collection is achieved using a rectangular interleaver of size $N_{row} \times N_{col}$. The number of rows and columns are determined from:

$$N_{row} = 4 \text{ for 16QAM and } N_{row} = 2 \text{ for QPSK}$$

$$N_{col} = N_{data} / N_{row}$$

where N_{data} is used.

[0170] Data is written into the interleaver column by column, and read out of the interleaver column by column starting from the first column.

[0171] $N_{t,sys}$ is the number of transmitted systematic bits. Intermediate values N_r and N_c are calculated using:

$$N_r = \left\lfloor \frac{N_{t,sys}}{N_{col}} \right\rfloor \text{ and } N_c = N_{t,sys} - N_r \cdot N_{col}.$$

[0172] If N_c=0 and $N_r > 0$, the systematic bits are written into rows 1...N_r.

[0173] Otherwise systematic bits are written into rows 1...N_r+1 in the first N_c columns and, if $N_r > 0$, also into rows 1...N_r in the remaining N_{col}-N_c columns.

[0174] The remaining space is filled with parity bits. The parity bits are written column wise into the remaining rows of the respective columns. Parity 1 and 2 bits are written in alternating order, starting with a parity 2 bit in the first available column with the lowest index number. If the two parity streams are of unequal length, Parity 1 and 2 bits shall be written alternately, again starting

Ex.1002 / Page 203 of 744
Sandisk Technologies, Inc. et al

with a parity 2 bit, until the end of the shorter parity stream, then the remaining parity bits from the longer stream shall be written.

[0175] In the case of 16QAM for each column the bits are read out of the interleaver in the order row 1, row 2, row 3, row 4. In the case of QPSK for each column the bits are read out of the interleaver in the order row1, row2.

[0176] when the analysis of the rate matching issue was restricted to the situation in which the first stage HS-DSCH rate matching is transparent it was shown that the performance degradation arises from an interaction between the periodicity of the rate matching pattern and the inherent periodicity of the turbo encoder. In particular, it was shown that at the problematic code rates, the positions (in the unpunctured bit stream) of survivor (i.e. non punctured) bits lie in the same position or positions within a modulo 7 period for long segments of the unpunctured stream. It has been demonstrated that performance degradations arise via exactly the same mechanism when the first stage is not transparent.

[0177] In the following description, a "puncturing pattern" is defined to be the pattern of positions of bits that are punctured from an original stream. "Puncturing Rate" is defined to be the reciprocal of the number of bits between punctured bit positions (inclusive of the punctured positions) in the original bit stream.

[0178] In the case of two-stage rate matching, puncturing that leads to degradation may occur in two ways:

1) A pattern arising from bits punctured in the first stage only

2) A pattern arising from bits punctured after the second stage; this pattern is dependent upon both the first and second rate matching stages.

[0179] In order to avoid either of these puncturing mechanisms that cause a performance degradation, the following methodology has been adopted:

[0180] In the first stage, test whether the puncturing will cause a problematic puncturing pattern. This is achieved by evaluating an "implicit duration" function, which predicts the length of periodicity in the puncturing pattern. The first stage rate matching is only adjusted if the calculated implicit

duration indicates a necessity. In this case, the amount of bits punctured in the first parity stream is increased slightly, whilst in the second stream the number is correspondingly decreased slightly. This has the effect of shifting the puncturing rates in both streams to break up modulo 7 periodicity, whilst keeping the total amount of punctured bits the same. The adjustment in the puncturing rate in each of the streams is kept sufficiently small to leave the coding performance in each stream unaffected.

[0181] In the second stage, a composite puncturing rate for each stream, taking into account adjustments to the puncturing rate carried out at the first stage is calculated. The composite puncturing rate in each stream is compared with a look up table, which predicts rates at which modulo-7 periodic patterns, and hence coding degradations will arise. If the composite rate of a stream falls within a range of problematic rates, a "dithering" algorithm is applied.

[0182] Differing puncturing rates are applied in the two parity streams if the puncturing pattern in the first stage will lead to a performance degradation, or is likely to contribute to an unwanted pattern after second stage rate matching. In this case, the number of bits punctured in the first parity stream is reduced by an amount ΔN_{PARITY}, whilst the number of bits punctured in the second stream is increased by the same amount. The decision as to whether such a change should take place is based upon a calculated parameter, which estimates the inverse of the length of runs of survivor bits which hold the same modulo 7 position in the unpunctured stream. This length becomes greater than 49 bits (i.e. 7 modulo 7 periods), then the adjustment to the lengths of the streams is applied.

[0183] The dithered puncturing algorithm may be applied to either or both of the two parity streams in the second stage. When processing a parity stream, the algorithm employs two puncturing rates, one higher than the original, and one lower, it then switches between these two puncturing rates whilst applying puncturing to the parity bit stream. The total number of bits punctured is kept the same as it would have been with the original algorithm. Generically it is possible to have one or many switching points between the two puncturing rates

within a parity stream. The number of points is not critical to the performance, since the fraction of the bit stream punctured at each rate is independent of the number of switching points.

[0184] The number of switching points is set to one (or two), meaning that the punctured stream consists of two (or three) segments in which the first segment uses a lower puncturing rate than the original, the second uses a higher puncturing rate. If the two segments do not account for the whole of the parity stream, a third segment is allowed at the end that is punctured at the original rate. The third segment is kept short, and allows for puncturing exactly the correct amount of bits to maintain the overall puncturing rate. By breaking the parity stream into a low number of segments in this way, the original rate matching algorithm can easily be used with some minor parameter changes.

[0185] The difference between the higher puncturing rate in the first segment and the original rate, and also between the lower puncturing rate in the second segment and the original has been set to 1/49. This selection allows for problematic rates to be avoided in all cases whilst not representing a significant change in coding rate.

[0186] In the majority of cases, the modified rate matching algorithm operates in exactly the same manner as previously defined. In the minority of cases in which either the first or second stage rate matching would lead to a performance degradation, unequal length streams and/or dithering are applied. Both of these modifications do not change the body of the rate matching procedure in any way, but rather the rate matching parameters that are applied. Furthermore, in an implementation, the modifications can be achieved with a modification to the Rate Matching DSP software without affecting any other part of the system.

[0187] The foregoing is a description of a preferred embodiment and alternative solutions to providing detecting, avoiding/correcting problematic Turbo code puncturing patterns. While this invention has been particularly shown and described with reference to preferred embodiments, it will be understood by those skilled in the art that various changes in form and details

may be made therein without departing from the scope of the invention as described hereinabove.

* * *

Ex.1002 / Page 207 of 744
Sandisk Technologies, Inc. et al

CLAIMS

What is claimed is:

1. In a communications system employing a plurality of rate matching stages, a method of avoiding problematic Turbo code puncturing patterns, the method comprising:

(a) adjusting the number of bits punctured in each stage of rate matching; and

(b) adjusting the number of bits punctured in each of the plurality of parity streams, wherein the problematic puncturing patterns are avoided.

2. The method of claim 1 wherein degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits are identified, and step (b) further comprises:

(b1) adding punctured bits to the first group of P1 bits;

(b2) removing punctured bits from the second group of P2 bits; and

(b3) biasing the punctured rates of the P1 and P2 bits to avoid problematic puncturing patterns by:

(i) adding a number of non-punctured P1 bits to the first group; and

(ii) decreasing the number of non-punctured P2 bits in the second group

by the number of non-punctured P1 bits added to the first group.

3. The method of claim 2 further comprising:

(c) determining a number of bits $\hat{N}$ using $\hat{N} = \left\lfloor \frac{4I}{7P} + \frac{1}{2} \right\rfloor$ wherein I is the

number of bits at the input to each branch of rate matching and P is the total number of the P1 and P2 bits at the output of rate matching; and

Ex.1002 / Page 208 of 744
Sandisk Technologies, Inc. et al

(d) if $\left|\frac{I}{(P/2)}-\frac{7\hat{N}}{2}\right|<1-\frac{\hat{N}}{2}+\left\lfloor\frac{\hat{N}}{2}\right\rfloor$, calculate the bias

$$\Delta=\left\lceil \max\left\{\frac{I}{\left\lfloor\frac{7\hat{N}-1}{2}\right\rfloor}-\frac{P}{2}\,,\ \frac{P}{2}-\frac{I}{\left\lceil\frac{7\hat{N}+1}{2}\right\rceil}\right\}\right\rceil\,,\ \text{otherwise set}\ \Delta=0\,.$$

4. The method of claim 3, wherein non-puncturing patterns with a period of $7\hat{N}/2$ cause degradation in performance results and $\hat{N}$ is a whole number.

5. The method of claim 4 wherein the periods will be employed whenever the average non-puncturing period of P1 or P2 bits is within ±1 or ±½ of $7\hat{N}/2$ for even and odd $\hat{N}$ respectively.

6. In a communications system employing a plurality of rate matching stages, a method of avoiding problematic Turbo code puncturing patterns, the method comprising:

(a) adjusting the number of bits punctured in each stage of rate matching; and

(b) adjusting the puncturing rates of each of the individual parity streams while maintaining a constant overall effective coding rate.

7. The method of claim 6 wherein degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits are identified, and step (b) further comprises:

(b1) adding punctured bits to the first group of P1 bits;

(b2) removing punctured bits from the second group of P2 bits; and

(b3) biasing the punctured rates of the P1 and P2 bits to avoid problematic puncturing patterns by:

(i) adding a number of non-punctured P1 bits to the first group; and

(ii) decreasing the number of non-punctured P2 bits in the second group

by the number of non-punctured P1 bits added to the first group.

8. The method of claim 7 further comprising:

(c) determining a number of bits $\hat{N}$ using $\hat{N} = \left\lfloor \frac{4I}{7P} + \frac{1}{2} \right\rfloor$ wherein I is the

number of bits at the input to each branch of rate matching and P is the total number of the P1 and P2 bits at the output of rate matching; and

(d) if $\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| < 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$, calculate the bias

$$\Delta = \left\lceil \left| \max\left\{ \frac{I}{\left\lfloor \frac{7\hat{N}-1}{2} \right\rfloor} - \frac{P}{2}, \frac{P}{2} - \frac{I}{\left\lceil \frac{7\hat{N}+1}{2} \right\rceil} \right\} \right| \right\rceil$$, otherwise set $\Delta = 0$.

9. The method of claim 8, wherein non-puncturing patterns with a period of $7\hat{N}/2$ cause degradation in performance results and $\hat{N}$ is a whole number.

10. The method of claim 9 wherein the periods will be employed whenever the average non-puncturing period of P1 or P2 bits is within ±1 or ±½ of $7\hat{N}/2$ for even and odd $\hat{N}$ respectively.

11. A method of identifying degradations in quality of punctured error correction coded transmissions, the method comprising:

(a) identifying a puncturing pattern which approximates a

Ex.1002 / Page 210 of 744
Sandisk Technologies, Inc. et al

particular code rate; and

(b) adjusting a value for anticipated degradation in accordance with the matching of the puncturing pattern and the particular code rate.

12. The method of claim 11 wherein degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits are identified, and step (b) further comprises:

(b1) adding punctured bits to the first group of P1 bits;

(b2) removing punctured bits from the second group of P2 bits; and

(b3) biasing the punctured rates of the P1 and P2 bits to avoid problematic puncturing patterns by:

(i) adding a number of non-punctured P1 bits to the first group; and

(ii) decreasing the number of non-punctured P2 bits in the second group by the number of non-punctured P1 bits added to the first group.

13. The method of claim 12 further comprising:

(c) determining a number of bits $\hat{N}$ using $\hat{N} = \left\lfloor \frac{4I}{7P} + \frac{1}{2} \right\rfloor$ wherein I is the number of bits at the input to each branch of rate matching and P is the total number of the P1 and P2 bits at the output of rate matching; and

(d) if $\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| < 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$, calculate the bias

$$\Delta = \left\lceil \left| \max\left\{ \frac{I}{\left\lfloor \frac{7\hat{N}-1}{2} \right\rfloor} - \frac{P}{2}, \frac{P}{2} - \frac{I}{\left\lceil \frac{7\hat{N}+1}{2} \right\rceil} \right\} \right| \right\rceil , \text{ otherwise set } \Delta = 0 .$$

14. The method of claim 11 further comprising:

(c) using Turbo code to implement the error correction coded transmissions.

15. The method of claim 14 further comprising:

(d) identifying when a non-punctured bit pattern of the transmissions

exhibits a periodic characteristic, with a period equal to a period of a semi-periodic impulse response of recursive encoding blocks of the Turbo code; and

(e) using the identified non-punctured bit patterns which exhibit a periodic characteristic to identify puncturing patterns with degraded performance.

16. A method of identifying degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits, the method comprising:

(a) adding punctured bits to the first group of P1 bits;

(b) removing punctured bits from the second group of P2 bits; and

(c) biasing the punctured rates of the P1 and P2 bits to avoid problematic

puncturing patterns by:

(i) adding a number of non-punctured P1 bits to the first group; and

(ii) decreasing the number of non-punctured P2 bits in the second group

Ex.1002 / Page 212 of 744
Sandisk Technologies, Inc. et al

by the number of non-punctured P1 bits added to the first group.

17. The method of claim 16 further comprising:

(d) using Turbo code to implement the error correction coded transmissions.

18. The method of claim 17 further comprising:

(e) identifying when a non-punctured bit pattern of the transmissions

exhibits a periodic characteristic, with a period equal to a period of a semi-periodic impulse response of recursive encoding blocks of the Turbo code; and

(f) using the identified non-punctured bit patterns which exhibit a periodic characteristic to identify puncturing patterns with degraded performance.

19. A method for reducing degradations in quality of punctured error corrected code transmissions, the method comprising:

(a) identifying a puncturing pattern which approximates a particular code

rate; and

(b) adjusting the parameters of the transmissions sufficiently to cause a

mismatch in the puncturing pattern and the particular code rate.

20. The method of claim 19 further comprising:

(c) determining a capacity of a wireless transmit and receive unit (WTRU), including buffer sizes that are supported by the WTRU;

(d) using puncturing to remove sufficient bits to fit into the buffer; and

(e) adjusting an overall code rate so as to provide sufficient error

Ex.1002 / Page 213 of 744
Sandisk Technologies, Inc. et al

correction capability, thereby providing a first rate in a first stage of puncturing and providing a second rate in a second stage of puncturing.

21. The method of claim 20, further comprising:

(f) increasing non-punctured bits in one of the first stage and second stage of puncturing, and decreasing non-punctured bits in another of the first stage and second stage of puncturing, thereby adding additional puncturing to one stage and removing it from the other stage.

22. The method of claim 20 further comprising:

(f) increasing non-punctured bits in the first stage and decreasing non-punctured bits in the second stage.

23. The method of claim 20 further comprising:

(f) decreasing non-punctured bits in the first stage and increasing non-punctured bits in the second stage.

24. The method of claim 20 further comprising:

(f) interleaving parity bits before rate matching occurs; and

(g) subsequently de-interleaving the parity bits, thereby avoiding a need

to periodically sample the parity bits when performing periodic sampling in rate matching, thereby mitigating the effect of the periodicity of the puncturing pattern.

25. A communications system for avoiding problematic Turbo code puncturing patterns, the system comprising:

(a) a plurality of rate matching stages;

(b) means for adjusting the number of bits punctured in each stage of rate matching; and

(c) means for adjusting the number of bits punctured in each of the

Ex.1002 / Page 214 of 744
Sandisk Technologies, Inc. et al

plurality of parity streams, wherein the problematic puncturing patterns are avoided.

26. The system of claim 25 wherein degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits are identified, the means (c) for adjusting the number of bits punctured in each of the plurality of parity streams further comprising:

(c1) means for adding punctured bits to the first group of P1 bits;

(c2) means for removing punctured bits from the second group of P2 bits;

and

(c3) means for biasing the punctured rates of the P1 and P2 bits to avoid

problematic puncturing patterns, the biasing means including:

(i) means for adding a number of non-punctured P1 bits to the first group;

and

(ii) means for decreasing the number of non-punctured P2 bits in the second group by the number of non-punctured P1 bits added to the first group.

27. The system of claim 26 further comprising:

(d) means for determining a number of bits $\hat{N}$ using $\hat{N} = \left\lfloor \frac{4I}{7P} + \frac{1}{2} \right\rfloor$

wherein

I is the number of bits at the input to each branch of rate matching and P is the total number of the P1 and P2 bits at the output of rate matching; and

(e) means for calculating the bias

$$\Delta = \left\lceil \max\left\{ \frac{I}{\left\lfloor \frac{7\hat{N}-1}{2} \right\rfloor} - \frac{P}{2}, \frac{P}{2} - \frac{I}{\left\lceil \frac{7\hat{N}+1}{2} \right\rceil} \right\} \right\rceil \text{ if}$$

$$\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| < 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor ; \text{ and}$$

(e) means for setting bias $\Delta = 0$ if $\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| \geq 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$.

28. The system of claim 27, wherein non-puncturing patterns with a period of $7\hat{N}/2$ cause degradation in performance results and $\hat{N}$ is a whole number.

29. The system of claim 28 wherein the periods will be employed whenever the average non-puncturing period of P1 or P2 bits is within ±1 or ±½ of $7\hat{N}/2$ for even and odd $\hat{N}$ respectively.

30. A communications system for avoiding problematic Turbo code puncturing patterns, the system comprising:

(a) a plurality of rate matching stages;

(b) means for adjusting the number of bits punctured in each stage of rate matching; and

(c) means for adjusting the puncturing rates of each of the individual parity streams while maintaining a constant overall effective coding rate.

31. The system of claim 30 wherein degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits are identified, the means (c) for adjusting the puncturing rates of each of the individual parity streams further

comprising:

(c1) means for adding punctured bits to the first group of P1 bits;

(c2) means for removing punctured bits from the second group of P2 bits;

and

(c3) means for biasing the punctured rates of the P1 and P2 bits to avoid

problematic puncturing patterns, the biasing means (c3) including:

(i) means for adding a number of non-punctured P1 bits to the first group;

and

(ii) means for decreasing the number of non-punctured P2 bits in the second group by the number of non-punctured P1 bits added to the first group.

32. The system of claim 31 further comprising:

(d) means for determining a number of bits $\hat{N}$ using $\hat{N} = \left\lfloor \frac{4I}{7P} + \frac{1}{2} \right\rfloor$

wherein

I is the number of bits at the input to each branch of rate matching and P is the total number of the P1 and P2 bits at the output of rate matching;

(e) means for calculating the bias

$$\Delta = \left\lceil \max\left\{ \frac{I}{\left\lfloor \frac{7\hat{N}-1}{2} \right\rfloor} - \frac{P}{2}, \frac{P}{2} - \frac{I}{\left\lceil \frac{7\hat{N}+1}{2} \right\rceil} \right\} \right\rceil \text{ if}$$

$$\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| < 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor ; \text{ and}$$

(e) means for setting bias $\Delta = 0$ if $\left|\frac{I}{(P/2)} - \frac{7\hat{N}}{2}\right| \geq 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$.

33. The system of claim 32, wherein non-puncturing patterns with a period of $7\hat{N}/2$ cause degradation in performance results and $\hat{N}$ is a whole number.

34. The system of claim 33 wherein the periods will be employed whenever the average non-puncturing period of P1 or P2 bits is within ±1 or ±½ of $7\hat{N}/2$ for even and odd $\hat{N}$ respectively.

35. A communications system for identifying degradations in quality of punctured error correction coded transmissions, the system comprising:

(a) means for identifying a puncturing pattern which approximates a particular code rate; and

(b) means for adjusting a value for anticipated degradation in accordance with the matching of the puncturing pattern and the particular code rate.

36. The system of claim 35 wherein degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits are identified, the means (b) for adjusting a value for anticipated degradation further comprising:

(b1) means for adding punctured bits to the first group of P1 bits;

(b2) means for removing punctured bits from the second group of P2 bits; and

(b3) means for biasing the punctured rates of the P1 and P2 bits to avoid problematic puncturing patterns, the biasing means (b3) including:

(i) means for adding a number of non-punctured P1 bits to the first group;

and

(ii) means for decreasing the number of non-punctured P2 bits in the second group by the number of non-punctured P1 bits added to the first group.

37. The system of claim 36 further comprising:

(c) means for determining a number of bits $\hat{N}$ using $\hat{N} = \left\lfloor \frac{4I}{7P} + \frac{1}{2} \right\rfloor$

wherein

I is the number of bits at the input to each branch of rate matching and P is the total number of the P1 and P2 bits at the output of rate matching;

(d) means for calculating the bias $\Delta = \left\lceil \max\left\{ \frac{I}{\left\lfloor \frac{7\hat{N}-1}{2} \right\rfloor} - \frac{P}{2}, \frac{P}{2} - \frac{I}{\left\lceil \frac{7\hat{N}+1}{2} \right\rceil} \right\} \right\rceil$ if

$\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| < 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$; and

(e) means for setting bias $\Delta = 0$ if $\left| \frac{I}{(P/2)} - \frac{7\hat{N}}{2} \right| \geq 1 - \frac{\hat{N}}{2} + \left\lfloor \frac{\hat{N}}{2} \right\rfloor$.

38. The system of claim 35 further comprising:

(c) means for using Turbo code to implement the error correction coded transmissions.

39. The system of claim 38 further comprising:

(d) means for identifying when a non-punctured bit pattern of the

transmissions exhibits a periodic characteristic, with a period equal to a period of a semi-periodic impulse response of recursive encoding blocks of the Turbo code; and

(e) means for using the identified non-punctured bit patterns which exhibit a periodic characteristic to identify puncturing patterns with degraded performance.

40. A communications system for identifying degradations in the quality of punctured error correction coded transmissions having a first group of parity 1 (P1) bits and a second group of parity 2 (P2) bits, the system comprising:

(a) means for adding punctured bits to the first group of P1 bits;

(b) means for removing punctured bits from the second group of P2 bits;

and

(c) means for biasing the punctured rates of the P1 and P2 bits to avoid

problematic puncturing patterns, the biasing means (c) including:

(i) means for adding a number of non-punctured P1 bits to the first group;

and

(ii) means for decreasing the number of non-punctured P2 bits in the second group by the number of non-punctured P1 bits added to the first group.

41. The system of claim 40 further comprising:

(d) means for using Turbo code to implement the error correction coded transmissions.

42. The system of claim 41 further comprising:

(e) means for identifying when a non-punctured bit pattern of the transmissions exhibits a periodic characteristic, with a period equal to a period of a semi-periodic impulse response of recursive encoding blocks of the Turbo code;

Ex.1002 / Page 220 of 744
Sandisk Technologies, Inc. et al

and

(f) means for using the identified non-punctured bit patterns which exhibit a periodic characteristic to identify puncturing patterns with degraded performance.

43. A communications system for reducing degradations in quality of punctured error corrected code transmissions, the system comprising:

(a) means for identifying a puncturing pattern which approximates a

particular code rate; and

(b) means for adjusting the parameters of the transmissions sufficiently to

cause a mismatch in the puncturing pattern and the particular code rate.

44. The system of claim 43 further comprising:

(c) means for determining a capacity of a wireless transmit and receive

unit (WTRU), including buffer sizes that are supported by the WTRU;

(d) means for using puncturing to remove sufficient bits to fit into the

buffer; and

(e) means for adjusting an overall code rate so as to provide sufficient

error correction capability, thereby providing a first rate in a first stage of puncturing and providing a second rate in a second stage of puncturing.

45. The system of claim 44, further comprising:

(f) means for increasing non-punctured bits in one of the first stage and second stage of puncturing, and decreasing non-punctured bits in another of the first stage and second stage of puncturing, thereby adding additional puncturing to one stage and removing it from the other stage.

46. The system of claim 44 further comprising:

(f) means for increasing non-punctured bits in the first stage and decreasing non-punctured bits in the second stage.

47. The system of claim 44 further comprising:

(f) means for decreasing non-punctured bits in the first stage and increasing non-punctured bits in the second stage.

48. The system of claim 44 further comprising:

(f) means for interleaving parity bits before rate matching occurs; and

(g) means for subsequently de-interleaving the parity bits, thereby avoiding a need to periodically sample the parity bits when performing periodic sampling in rate matching, thereby mitigating the effect of the periodicity of the puncturing pattern.

Ex.1002 / Page 222 of 744
Sandisk Technologies, Inc. et al

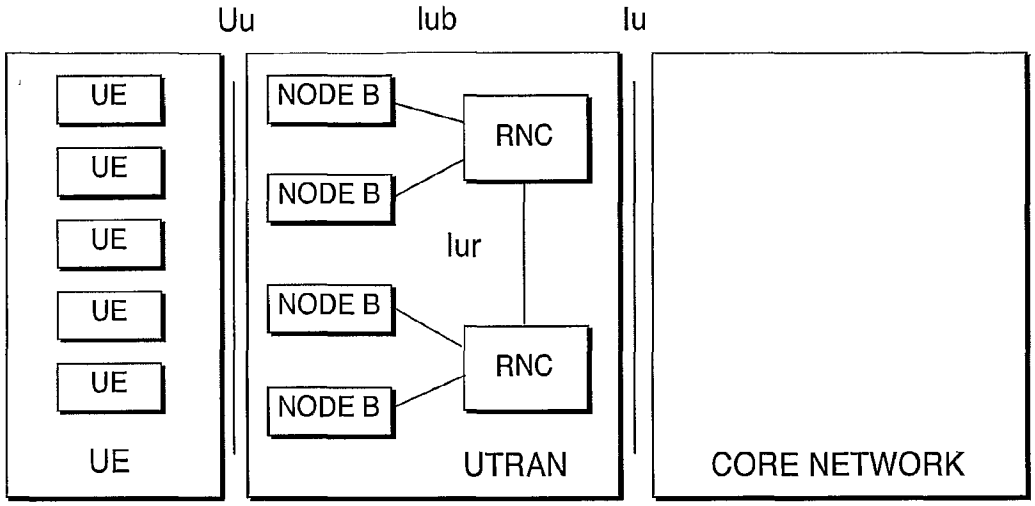


FIG. 1
PRIOR ART

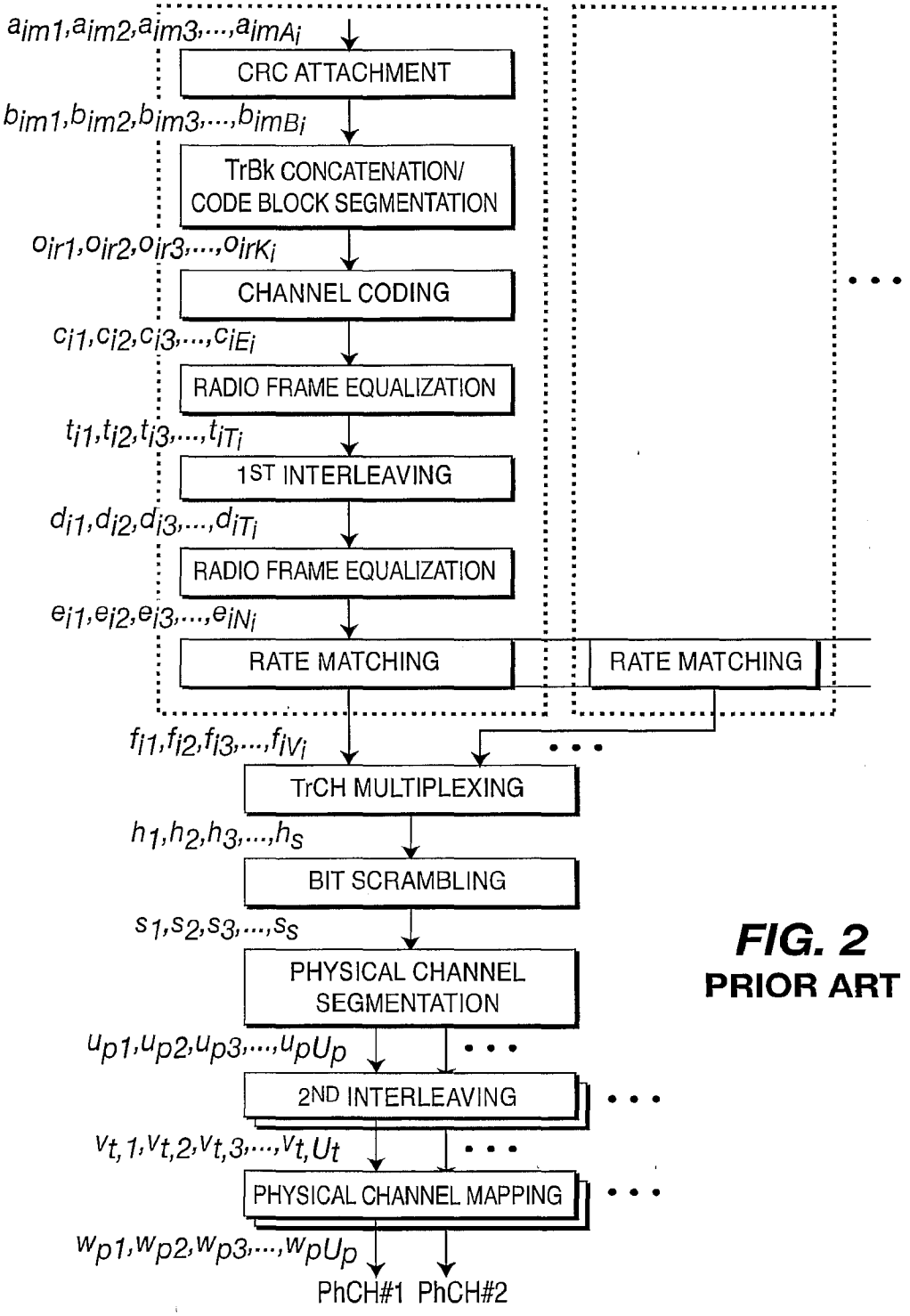


FIG. 2
PRIOR ART

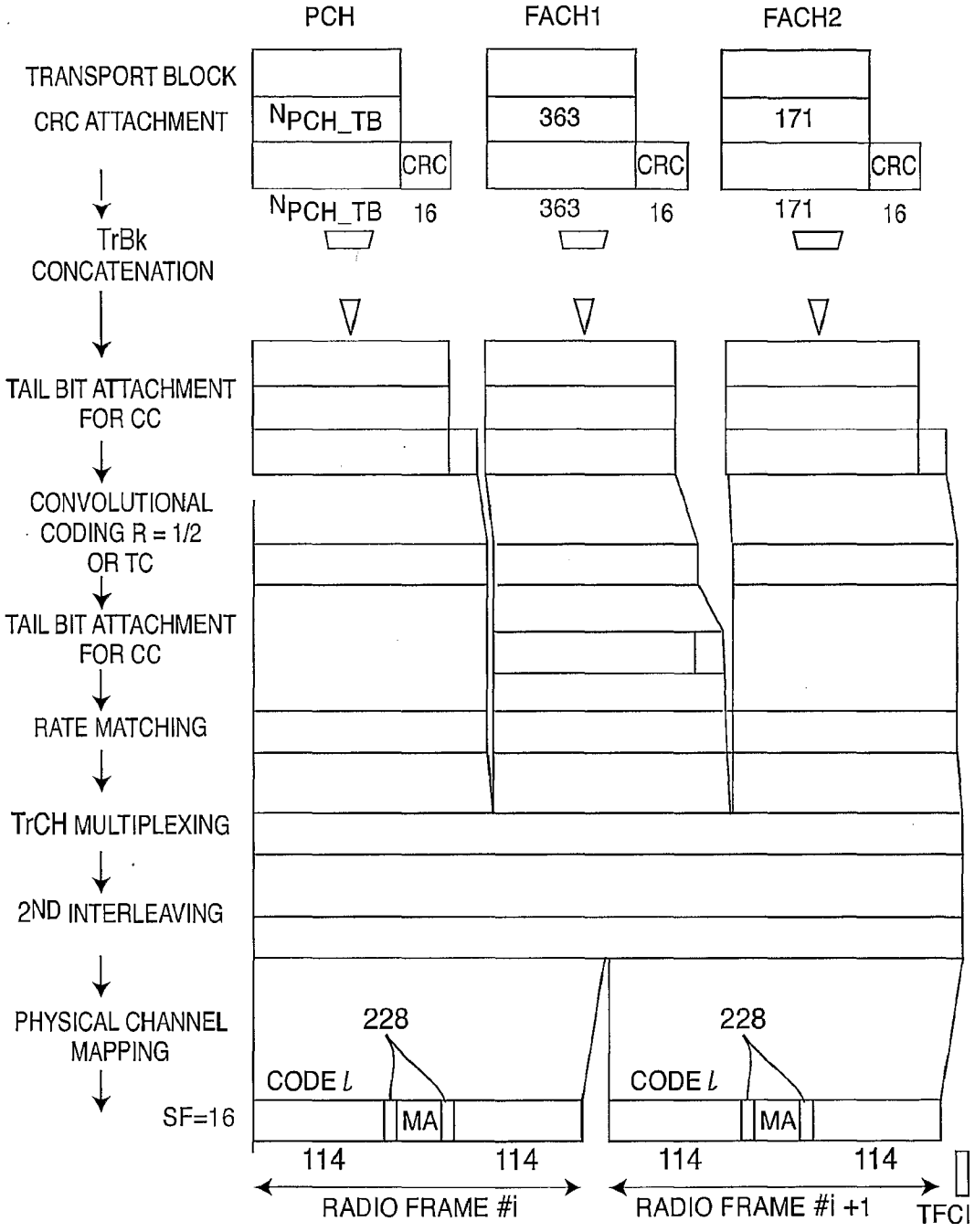


FIG. 3
PRIOR ART

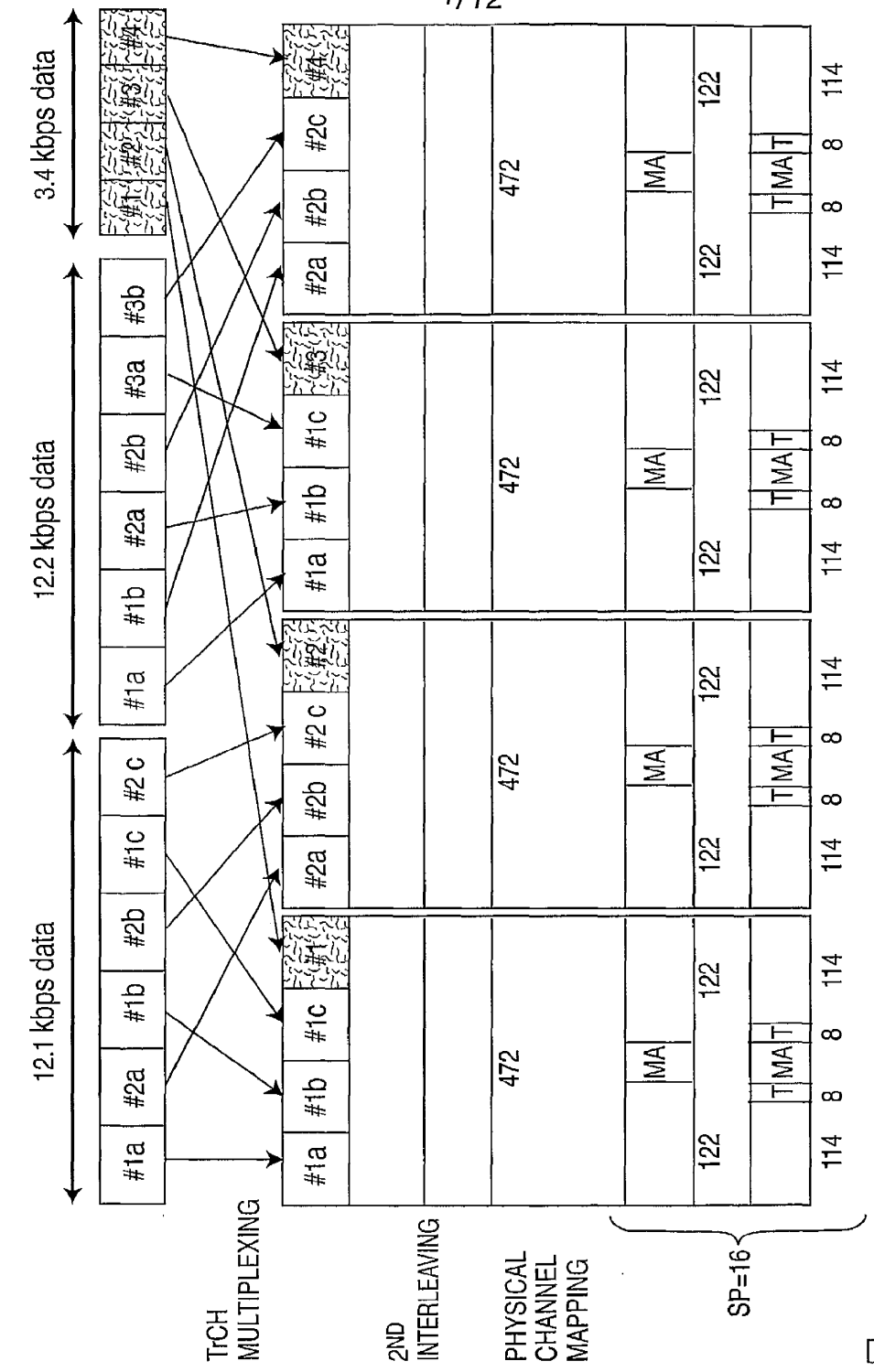
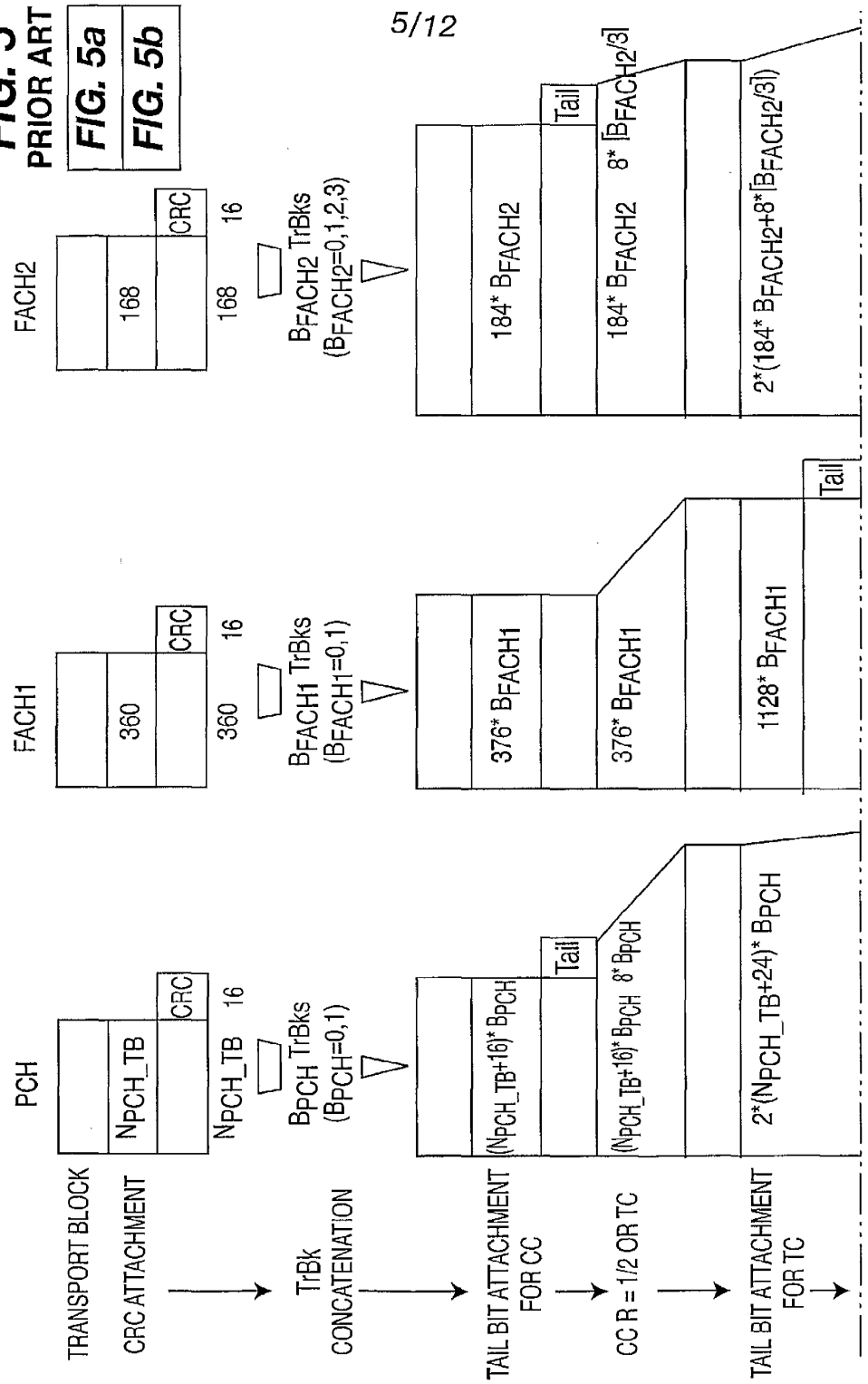


FIG. 4 PRIOR ART

FIG. 5 PRIOR ART

FIG. 5a

FIG. 5b



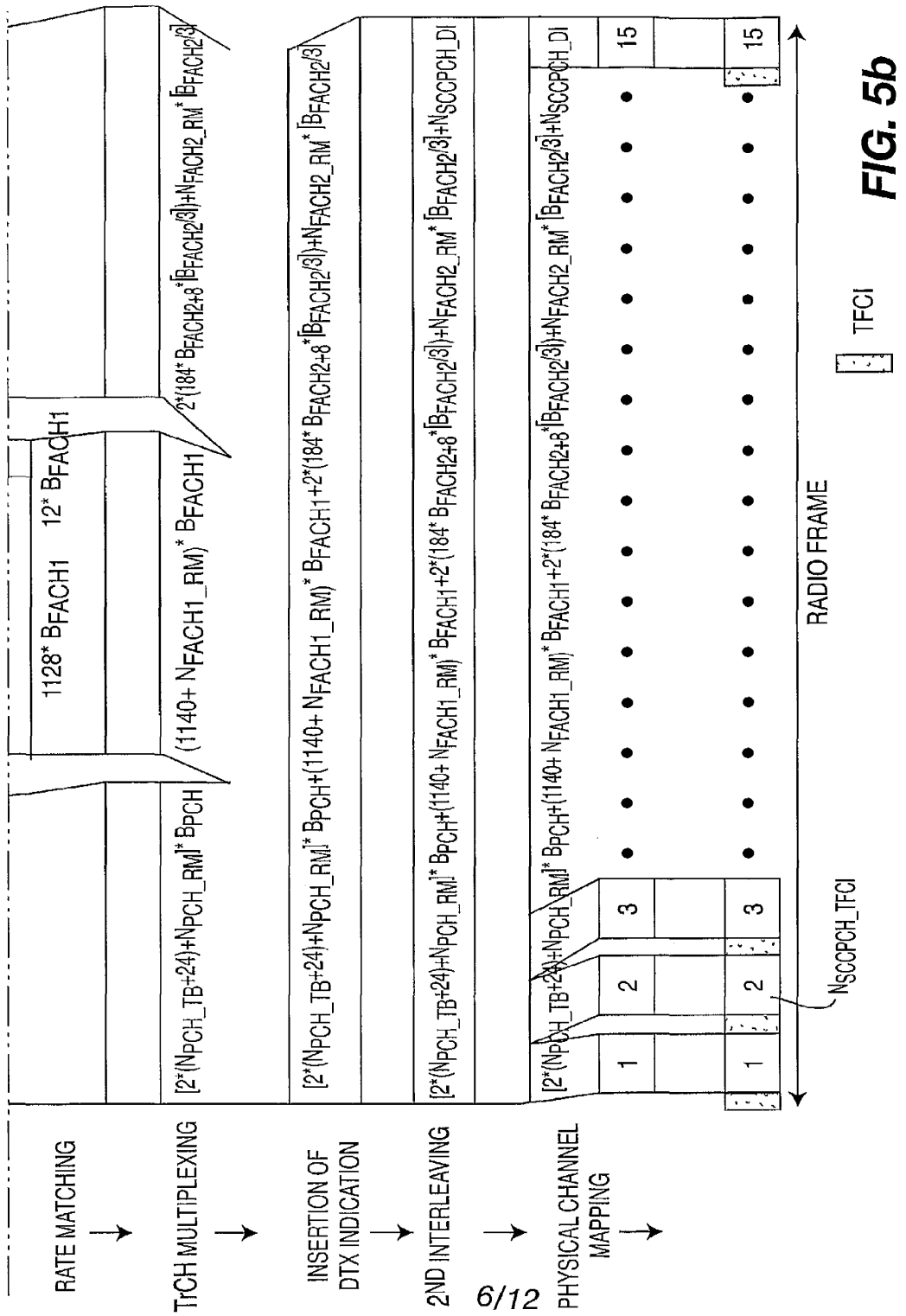

RATE MATCHING
TrCH MULTIPLEXING
INSERTION OF DTX INDICATION
2ND INTERLEAVING
PHYSICAL CHANNEL MAPPING
1128* BFACH1 12* BFACH1
(1140+ NFACH1_RM)* BFACH1
2*(184* BFACH2+8 *⌈BFACH2/3⌉)+NFACH2_RM* ⌈BFACH2/3⌉
[2*(NPCH_TB+24)+NPCH_RM]* BPCH
[2*(NPCH_TB+24)+NPCH_RM]* BPCH+(1140+ NFACH1_RM)* BFACH1+2*(184* BFACH2+8 *⌈BFACH2/3⌉)+NFACH2_RM* ⌈BFACH2/3⌉
[2*(NPCH_TB+24)+NPCH_RM]* BPCH+(1140+ NFACH1_RM)* BFACH1+2*(184* BFACH2+8 *⌈BFACH2/3⌉)+NFACH2_RM* ⌈BFACH2/3⌉+NSCCPCH_DI
[2*(NPCH_TB+24)+NPCH_RM]* BPCH+(1140+ NFACH1_RM)* BFACH1+2*(184* BFACH2+8 *⌈BFACH2/3⌉)+NFACH2_RM* ⌈BFACH2/3⌉+NSCCPCH_DI
1
2
3
15
NSCCPCH_TFCI
RADIO FRAME
TFCI

FIG. 5b

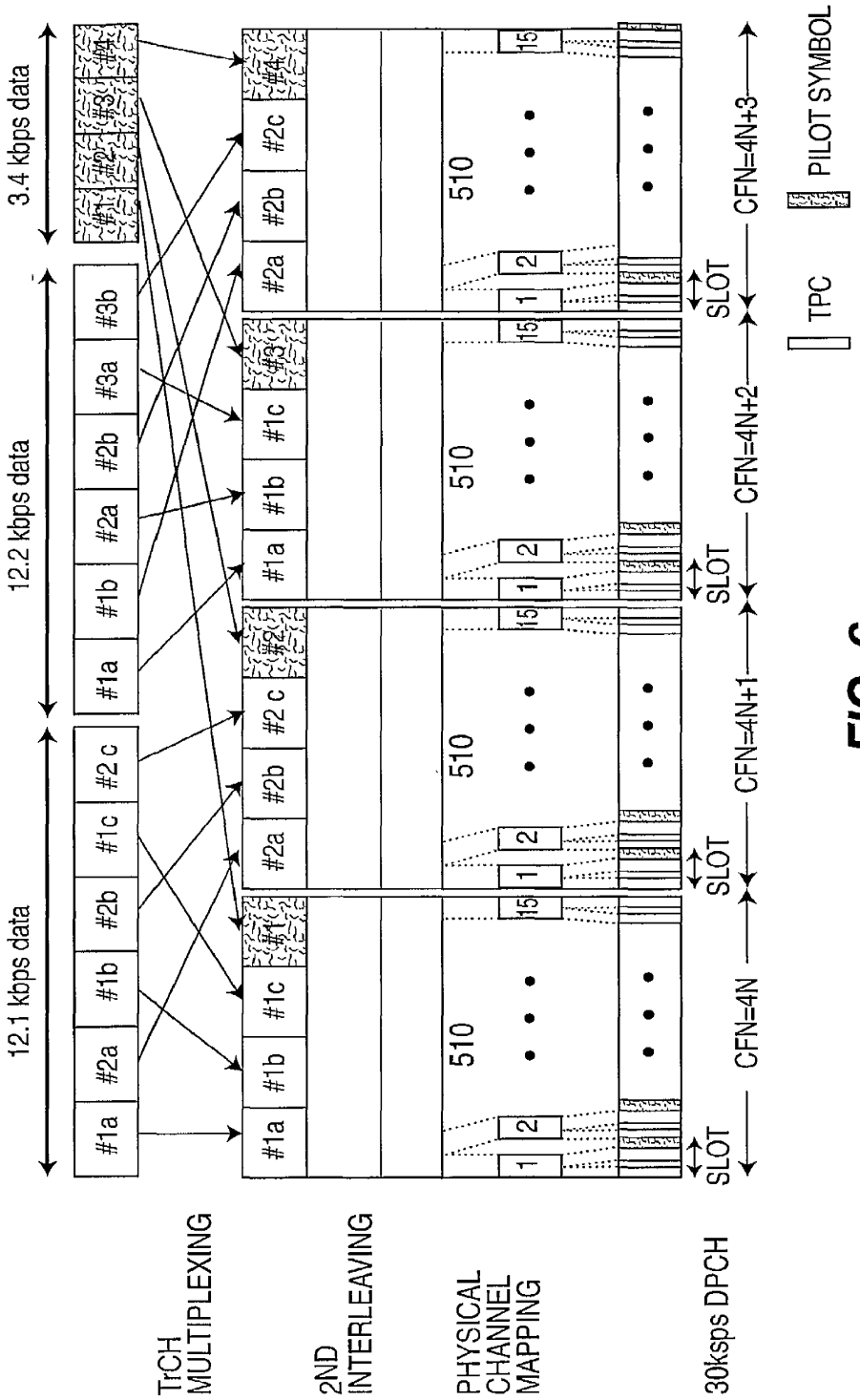


FIG. 6
PRIOR ART

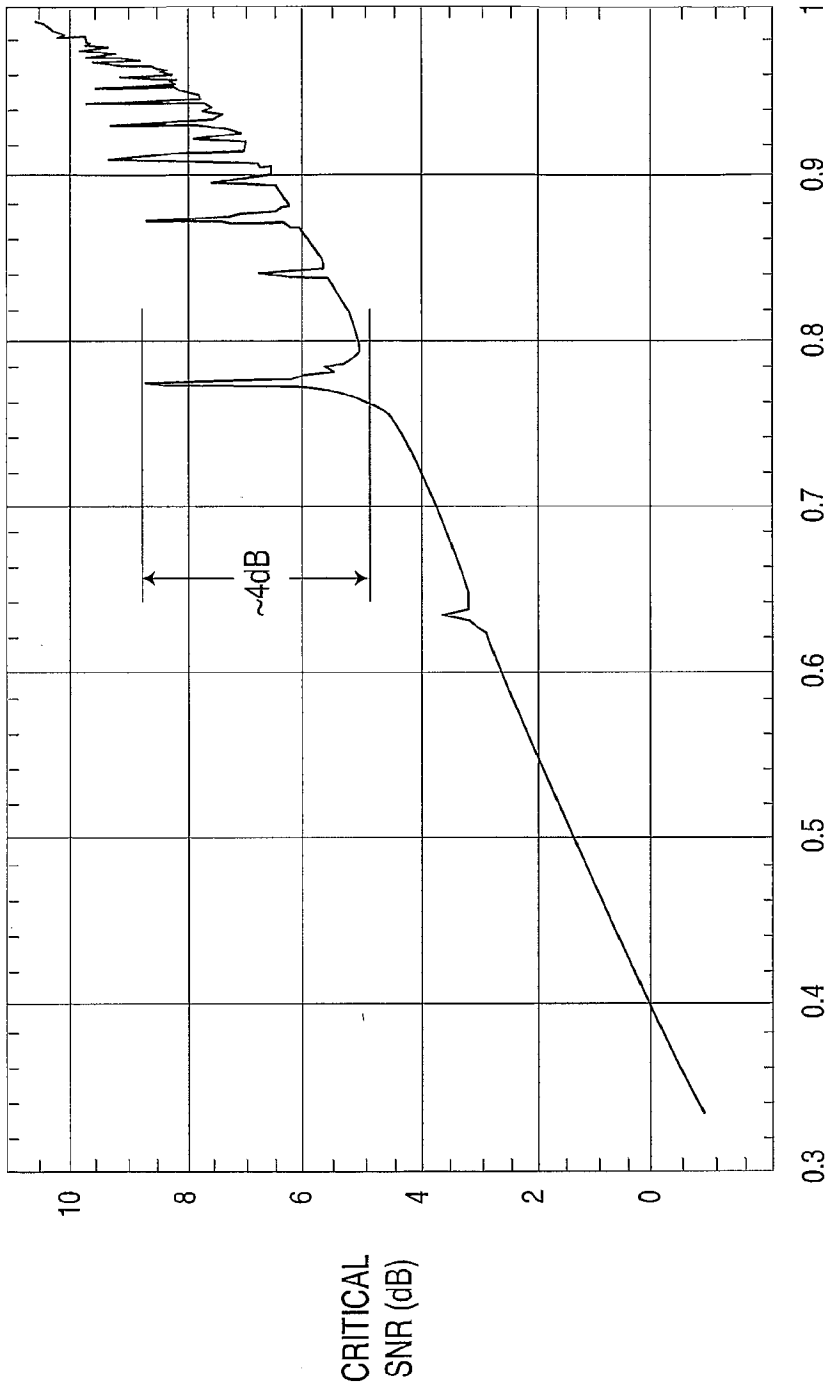


FIG. 7
PRIOR ART

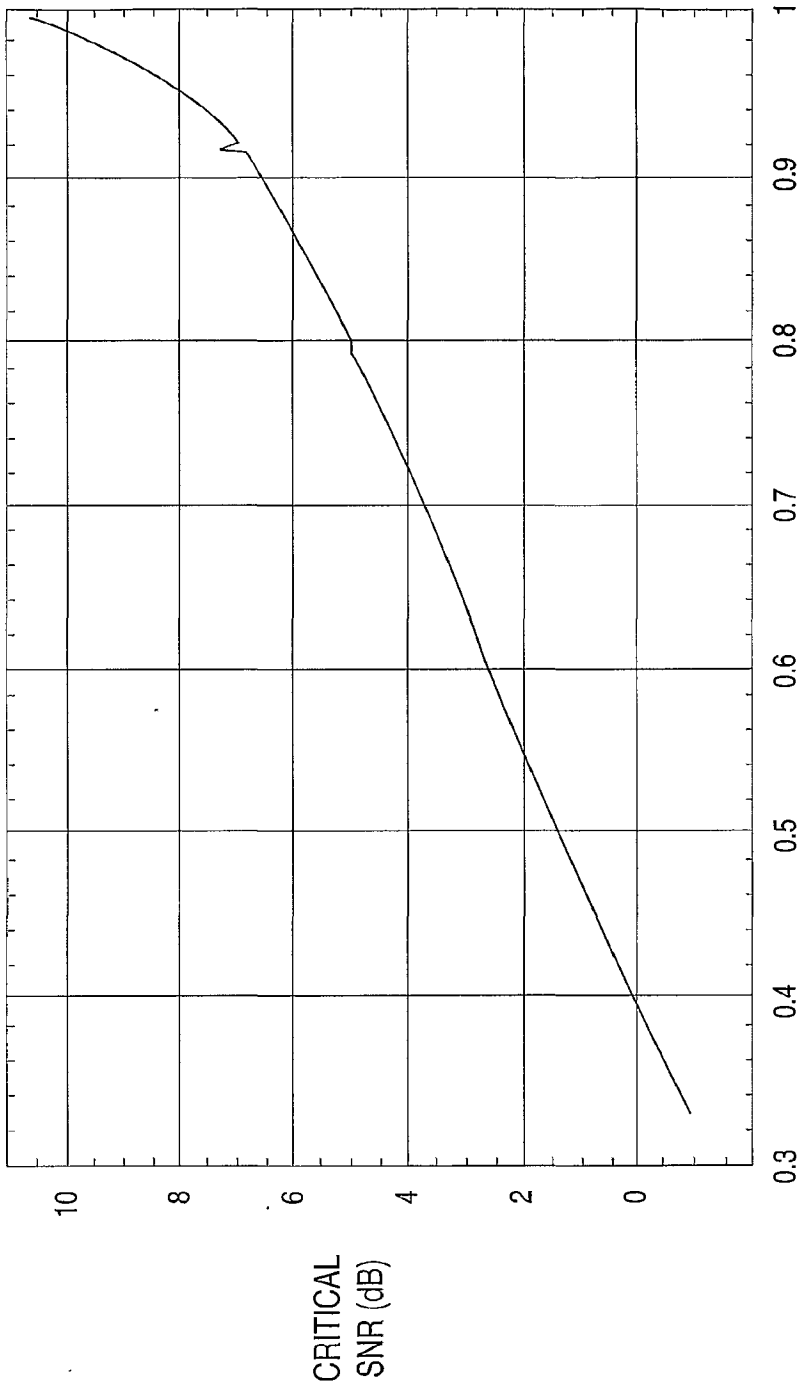


FIG. 8

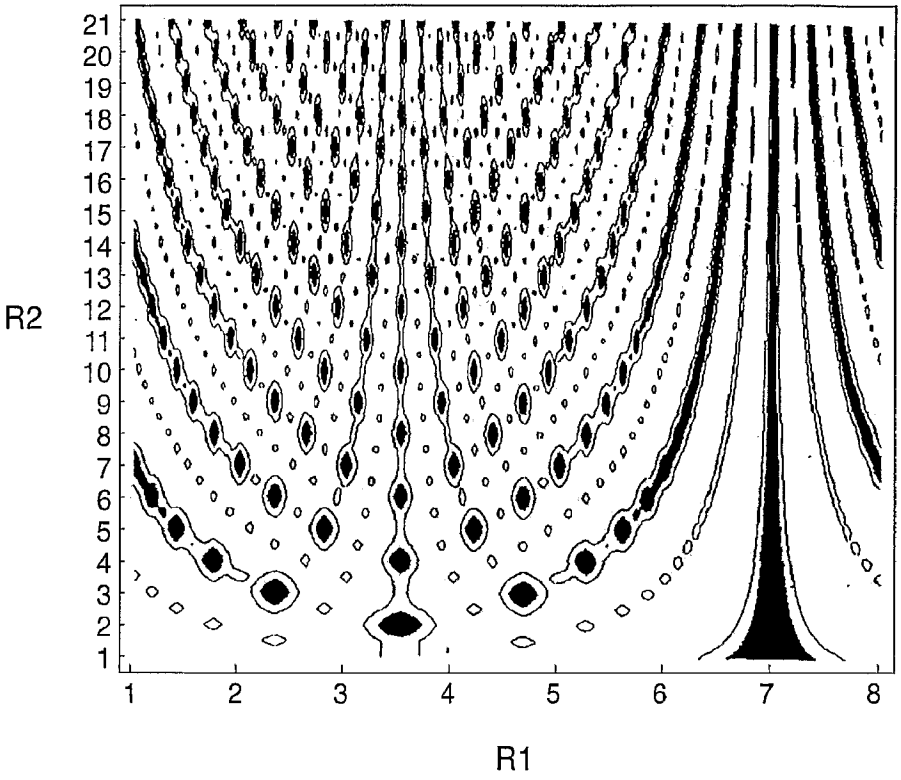


FIG. 9

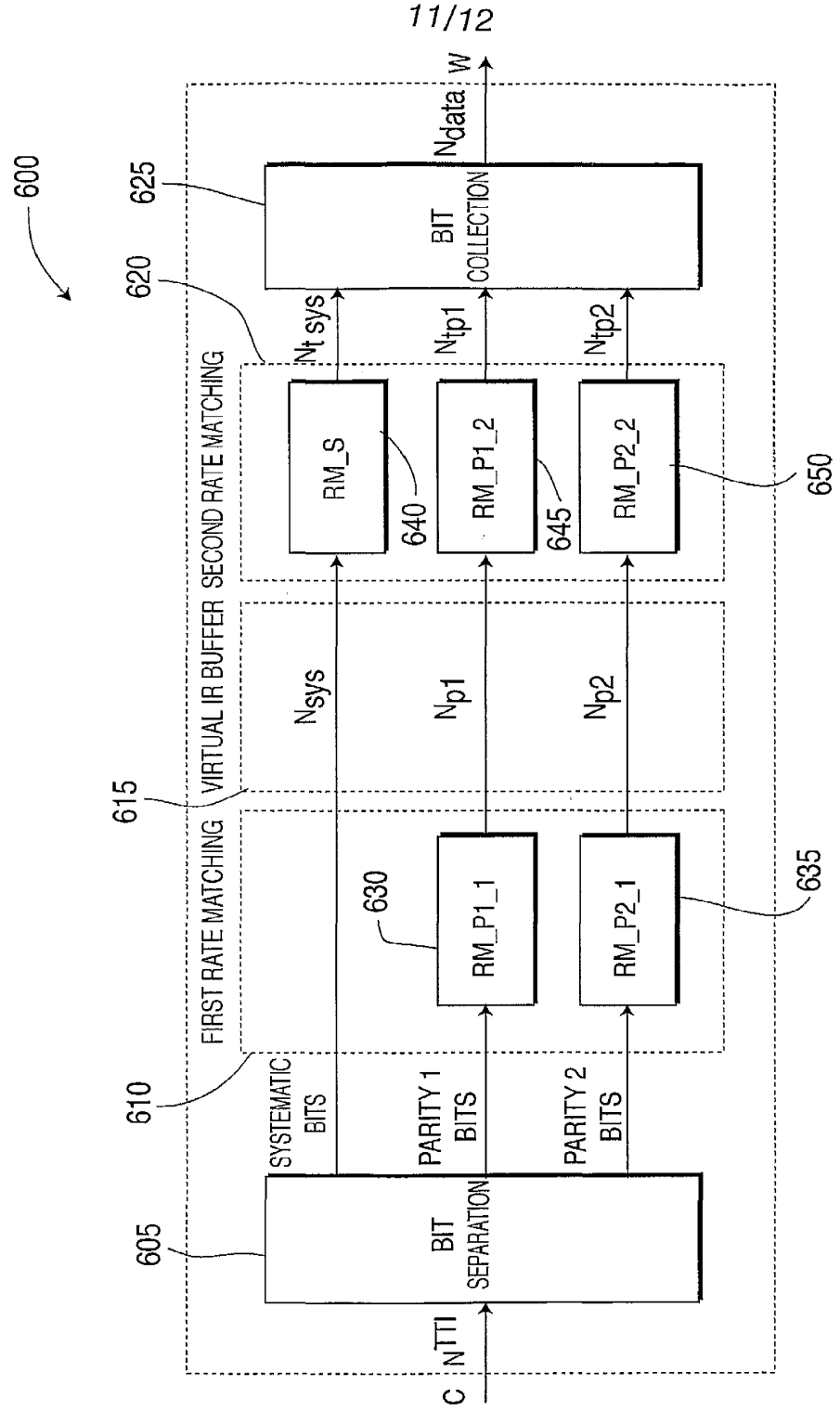

600
625
620
615
610
605
630
640
645
650
635
BIT COLLECTION
BIT SEPARATION
RM_S
RM_P1_2
RM_P2_2
RM_P1_1
RM_P2_1
SECOND RATE MATCHING
VIRTUAL IR BUFFER
FIRST RATE MATCHING
SYSTEMATIC BITS
PARITY 1 BITS
PARITY 2 BITS
Ndata
Nt sys
Ntp1
Ntp2
Nsys
Np1
Np2
NTTI
W
C

FIG. 10

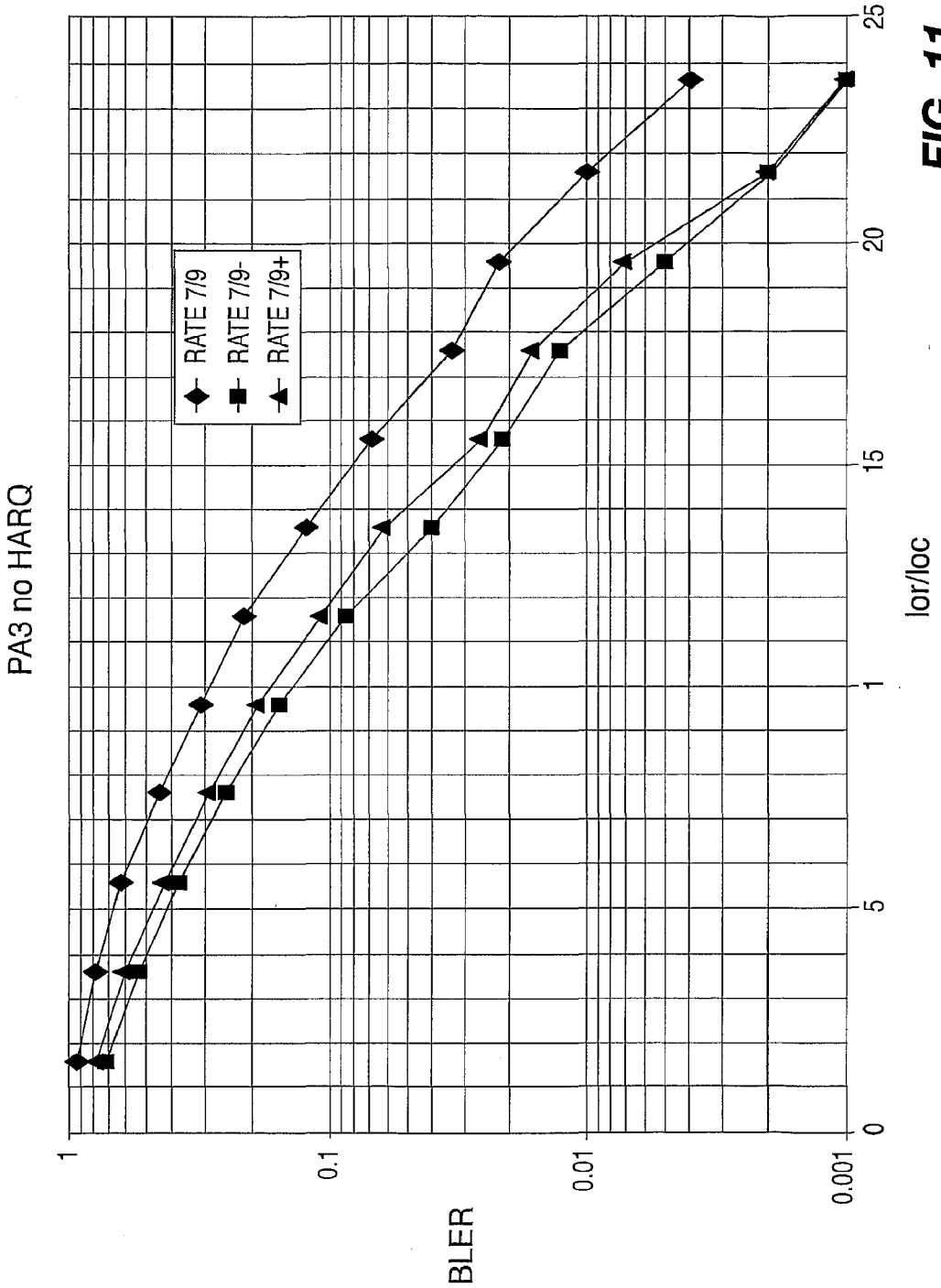

PA3 no HARQ
RATE 7/9
RATE 7/9-
RATE 7/9+
BLER
1
0.1
0.01
0.001
lor/loc
0
5
1
15
20
25

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/38223

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H03M 13/35, 13/27
US CL : 714/786, 790; 375/265

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 714/786, 790; 375/265

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
East, IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6,370,669 B1 (EROZ et al) 09 April 2002 (09.04.2002), abstract, column 3, lines 1-52.	1, 6, 11, 16, 19, 25, 30, 35, 40, 43
Y,P	US 6,601,214 B1 (HAMMONS, JR.) 29 July 2003 (29.07.2003), abstract, column 5, lines 23-67.	2-5, 7-10, 12-15, 17, 18, 20-24, 26-29, 31-34, 36-39, 41, 42, 44-48

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
30 March 2004 (30.03.2004)

Date of mailing of the international search report
13 APR 2004

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
Facsimile No. (703)305-3230

Authorized officer
Albert DeCady
Telephone No. (703) 305-9595

Form PCT/ISA/210 (second sheet) (July 1998)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 July 2007 (26.07.2007)

PCT

(10) International Publication Number
WO 2007/084751 A2

(51) International Patent Classification:
G11C 11/34 (2006.01) *G11C 16/06* (2006.01)
G11C 16/04 (2006.01)

(21) International Application Number:
PCT/US2007/001623

(22) International Filing Date: 19 January 2007 (19.01.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/760,622	20 January 2006 (20.01.2006)	US
60/761,888	25 January 2006 (25.01.2006)	US
60/771,621	8 February 2006 (08.02.2006)	US
11/598,178	8 November 2006 (08.11.2006)	US

(71) Applicant *(for all designated States except US)*: **MARVELL WORLD TRADE LTD.** [BB/BB]; L'Horizon, Gunsite Road, BB14027 Brittons Hill, St. Michael (BB).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **WU, Zining** [CN/US]; 836 South Springer Road, Los Altos, CA 94024 (US). **SUTARDJA, Pantas** [US/US]; 18690 Blythswood Drive, Los Gatos, CA 95030 (US).

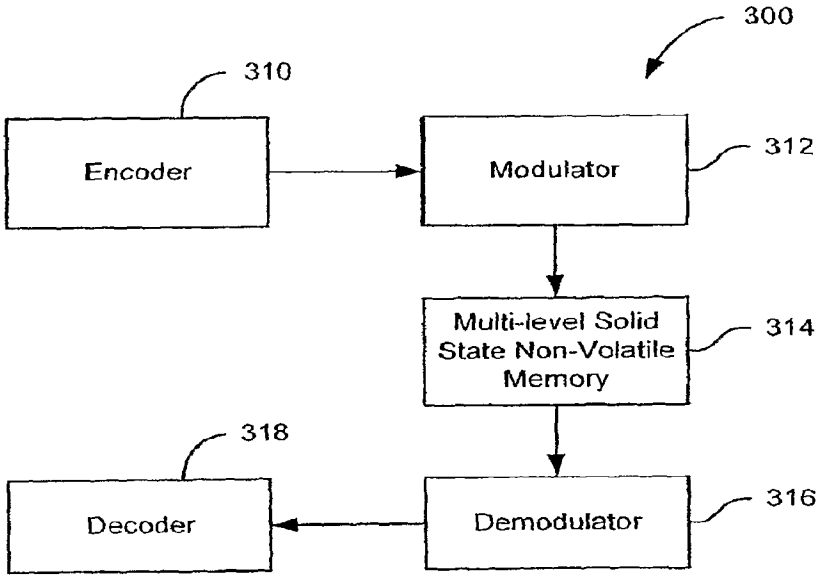
(74) Agent: **WHITE, Jason, C.**; Brinks Hofer Gilson & Lione, PC, NBC Tower, Suite 3600, 455 North Cityfront Plaza Drive, Chicago, IL 60611-5599 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,

[Continued on next page]

(54) Title: FLASH MEMORY WITH CODING AND SIGNAL PROCESSING



(57) Abstract: A solid state non-volatile memory unit includes, in part, an encoder, a multi-level solid state non-volatile memory array adapted to store data encoded by the encoder, and a decoder adapted to decode the data retrieved from the memory array. The memory array may be a flash EEPROM array. The memory unit optionally includes a modulator and a demodulator. The data modulated by the modulator is stored in the memory array. The demodulator demodulates the modulated data retrieved from the memory array.

WO 2007/084751 A2

RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

FLASH MEMORY WITH CODING AND SIGNAL PROCESSING

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] The present application claims benefit under 35 U.S.C. § 119(e) of the following U.S. provisional applications, all of which are commonly assigned and incorporated herein by reference in their entirety:

Provisional Application No. 60/760,622, entitled "Flash Memory - Error Correction Issues" filed on January 20, 2006;
Provisional Application No. 60/761,888, entitled "Increase Storage Capacity of Flash Memory through Coding and Signal Processing" filed on January 25, 2006; and
Provisional Application No. 60/771,621, entitled "Increase Storage Capacity of Flash Memory through Coding and Signal Processing" filed on February 8, 2006.

[0002] The present application is also related to concurrently filed U.S. Patent Application No. _____, attorney docket no. MP0909, entitled "Method And System For Error Correction In Flash Memory," commonly assigned and incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

[0003] The present invention relates generally to integrated circuits. More particularly, the invention relates to a method and system for performing error correction in multi-level solid state non-volatile memories.

[0004] Solid state non-volatile memories, such as flash EEPROM memories, are used in a variety of electronics applications. Flash memories are used in a number of memory card formats, such as CompactFlash (CF), MultiMediaCard (MMC) and Secure Digital (SD). Electronic systems in which such cards are used include personal and notebook computers, hand-held computing devices, cameras, MP3 audio players, and the like. Flash EEPROM memories are also utilized as bulk mass storage in many host systems.

[0005] Conventional solid state memories store information as a series of binary digits or "bits," which can take on one of two different values (0 or 1). Bits are grouped together to represent larger numbers.

[0006] As with most solid state non-volatile memory devices, flash EEPROMs are susceptible to defects and failures. Errors result from several factors including the gradual shifting of the threshold level of the memory states as a result of ambient conditions and stress from normal operations of the memory device including program, erase, and read operations. In order to prevent errors during operation, error correction code (ECC) techniques are utilized in flash memory devices. Typically, a controller generates redundant bits (parity bits) that are appended to the end of data sectors during program operations. For example, a 512 byte data sector may have 16 bytes of ECC data appended, resulting in a 528 byte page. During read operations, the redundant data included in the 16 bytes of ECC data is utilized to detect and correct errors in the data read out from the flash memory.

[0007] For a conventional memory, the maximum storage density is determined by the size of the individual storage elements and the number of storage elements that can be integrated onto a single integrated circuit chip. Typically, increases in memory density have been provided by shrinking the linewidth of the process geometry used to fabricate the memory cells.

[0008] Another technique used to increase solid state non-volatile memory density is storing more than one bit per memory cell, also referred to as a multi-level memory cell. Rather than sensing whether or not charge is stored in a given memory cell (i.e., a binary cell), multi-level memories utilize a sense amplifier that senses the amount of charge stored in a capacitive storage cell. By quantizing information into units greater than binary, e.g., 4-level (2 bits/cell), 8-level (3 bits/cell), 16-level (4 bits/cell) units, and the like, and storing these multi-level units, the memory density can be increased. As an example, a cell may be programmed to produce four distinct threshold levels, which results in four distinct read-back levels. With a four level signal available per cell, two data bits can be encoded into each solid state non-volatile memory cell. Multi-level memories enable the manufacturing of higher density memories without increasing the number of memory cells since each memory cell can store more than a single bit. Merely by way of example, for a memory cell capable of storing 2 bits/cell, there may be three programmed states and an erased state. FIG. 1 is a simplified probability distribution function (PDF) as a function of voltage for a solid state non-volatile memory cell having a 4-level quantization. In the memory cell illustrated in FIG. 1, four programmed states are utilized. As illustrated, in some solid state non-volatile memories, the PDF of programming characteristics has a wider distribution at lower voltage levels.

Ex.1002 / Page 239 of 744
Sandisk Technologies, Inc. et al

[0009] However, increasing the number of quantization levels in a cell results in a reduction in the voltage difference between adjacent levels. In multi-level encoding systems, this reduction is sometimes referred to as reduced signal distance (reduced D_{min}). Reduced signal distance may impact non-volatile memory performance in both write (program) as well as read operations. During programming, it is more difficult to transfer multiple discrete units of charge to a capacitive cell than it is simply to fully charge or fully discharge the cell. Thus, uncertainty in the amount of charge transferred to a given cell may result in a level shift, resulting in a "program disturb" in which the wrong level is stored in the cell. During reading, "read disturbs" occur when the distribution of one signal level overlaps the distribution of an adjacent signal level. Because the signal distance is reduced, the increase in the number of discrete values stored in the cell reduces the noise margin of the cell as compared to a binary storage cell, making the storage element more prone to erroneous readout. Read disturbs are more common for low-level signals, which are characterized by larger noise distributions as shown in FIG. 1.

[0010] The reduction in voltage separation between adjacent levels in a multi-level solid state non-volatile memory may lead to an increase in the number of errors in comparison with conventional solid state non-volatile memory cells. Thus, it would be desirable to have improved methods and techniques for operating solid state non-volatile memory with multi-level cells.

SUMMARY OF THE INVENTION

[0011] In accordance with one embodiment of the present invention, a solid state non-volatile memory unit includes, in part, an encoder, a multi-level solid state non-volatile memory array adapted to store data encoded by the encoder, and a decoder adapted to decode data retrieved from the memory array. In one embodiment, the memory unit is disposed in an integrated circuit. The memory array may be a flash EEPROM array.

[0012] In one embodiment, the memory unit may further include, a second encoder adapted to supply encoded data to the first encoder, and a second decoder adapted to decode the data supplied by the first decoder. The first encoder may execute a first ECC different from a second ECC that the second encoder is adapted to execute. In some embodiments, the second encoder comprises a Reed-Solomon encoder.

Ex.1002 / Page 240 of 744
Sandisk Technologies, Inc. et al

[0013] In some embodiments, the memory unit further includes, in part, a modulator and a demodulator. The data modulated by the modulator is subsequently stored in the memory array. The demodulator demodulates the modulated data retrieved from the memory array.

[0014] In some embodiments, the first encoder is a binary encoder, encoding data in accordance with, for example, Hamming code, BCH code, Reed-Muller code and Array Code. In yet other embodiments, the first encoder is a non-binary encoder, encoding data in accordance with, for example, Reed-Solomon (RS) code.

[0015] In some embodiments, the first encoder is a convolutional encoder. In yet other embodiments, the first encoder and the modulator form a trellis-coded modulator. The first encoder may be an iterative encoder encoding data in accordance with, for example, low-density parity check code and Turbo code. The first encoder may also be a symbol-based iterative encoder.

[0016] A method of operating a solid state non-volatile memory unit, in accordance with one embodiment of the present invention includes, encoding first data, storing the first encoded data in a multi-level solid state non-volatile memory array; and decoding the first data retrieved from the memory array. The memory array may be a flash EEPROM array.

[0017] In one embodiment, the method further includes, encoding a second data to generate the first data, and decoding the first decoded data to generate the second data. Encoding of the first data may be carried out in accordance with a first ECC different from a second ECC in accordance with which encoding of the second data is carried out. The second data may be encoded in accordance with a Reed-Solomon code.

[0018] In some embodiments, the method further includes, in part, modulating the encoded data; storing the modulated data in the memory array; and demodulating the data retrieved from the memory array. In some embodiments, the data may be encoded in accordance with a binary code, such as a Hamming code, BCH code, Reed-Muller code or Array Code. In yet other embodiments, the data may be encoded in accordance with a non-binary code, such as a Reed-Solomon code.

[0019] In some embodiments, the data may be encoded in accordance with a convolutional code. In yet other embodiments, the encoding and modulation of data may be performed in accordance with a trellis-coded modulation. In some embodiments, data may be encoded in

accordance with an iterative code, such as, a low-density parity check code or Turbo code. The iterative code may be a symbol-based iterative code.

[0020] In accordance with one embodiment of the present invention, a solid state non-volatile memory unit includes, in part, means for encoding first data, means for storing the first encoded data in a multi-level solid state non-volatile memory array; and means for decoding the first data retrieved from the memory array. The memory array may be a flash EEPROM array.

[0021] In one embodiment, the memory unit further includes, means for encoding a second data to generate the first data, and means for decoding the first decoded data to generate the second data. The means for encoding of the first data may execute a first ECC different from a second ECC technique executed by the second encoding means. The second encoding means may be a Reed-Solomon encoder.

[0022] In some embodiments, the memory unit further includes, in part, means for modulating the encoded data; and means for demodulating the data retrieved from the memory array. In some embodiments, the encoding means may perform binary encoding using codes, such as Hamming code, BCH code, Reed-Muller code and Array Code. In yet other embodiments, the encoding means may perform non-binary encoding using codes, such as a Reed-Solomon code.

[0023] In some embodiments, the encoding means may perform convolutional encoding. In yet other embodiments, the encoding means and the modulation means may perform trellis-coded modulation. In some embodiments, the encoding means may perform iterative encoding using codes, such as, low-density parity check code or Turbo code. The iterative code may be a symbol-based iterative code.

[0024] In accordance with one embodiment of the present invention, a solid state non-volatile memory unit includes, in part, code for encoding first data, code for storing the first encoded data in a multi-level solid state non-volatile memory array; and code for decoding the first data retrieved from the memory array. The memory array may be a flash EEPROM array.

[0025] In one embodiment, the memory unit further includes, code for encoding a second data to generate the first data, and code for decoding the first decoded data to generate the second data. The code for encoding of the first data may be an ECC code different from a

second ECC code used to encode the second data. The code used to encode the second data may be a Reed-Solomon code.

[0026] In some embodiments, the memory unit further includes, in part, code for modulating the encoded data; and code for demodulating the data retrieved from the memory array. In some embodiments, the code for encoding the data may a binary code, such as Hamming code, BCH code, Reed-Muller code and Array Code. In yet other embodiments, the code for encoding the data may be a non-binary code, such as a Reed-Solomon code.

[0027] In some embodiments, the code for encoding of the data may be a convolutional code. In yet other embodiments, the encoding code and the modulation code may be a trellis-coded modulation code. In some embodiments, the code used to encode the data may be an iterative code, such as, a low-density parity check code or Turbo code. The iterative code may be a symbol-based iterative code.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] FIG. 1 is a simplified probability distribution function as a function of voltage for a solid state non-volatile memory cell having a 4-level quantization;

[0029] FIG. 2A is a constellation diagram for an uncoded one bit PAM scheme;

[0030] FIG. 2B is a constellation diagram for a two-bit data modulated using a PAM scheme having an average power of 1/2;

[0031] FIG. 2C is a constellation diagram for a two-bit data modulated using a PAM scheme having a peak limit of $\pm$ 1/2 for use in a solid state non-volatile memory cell;

[0032] FIG. 3A is a simplified block diagram of an exemplary solid state non-volatile memory unit with error correction code according to an embodiment of the present invention;

[0033] FIG. 3B is a simplified block diagram of a solid state non-volatile memory unit incorporating an analog-to-digital converter according to an embodiment of the present invention;

[0034] FIG. 4 is a simplified block diagram of a solid state non-volatile memory unit with error correction according to another embodiment of the present invention;

[0035] FIG. 5 shows a word error rate (WER) of a memory unit, in accordance with one exemplary embodiment of the present invention, as compared to a conventional uncoded system;

[0036] FIG. 6 shows various blocks of an exemplary rate 1/2 convolutional encoder according to an embodiment of the present invention;

[0037] FIG. 7 shows an example of 2-D set partitioning according to an embodiment of the present invention;

[0038] FIG. 8 is a simplified block diagram of a conventional TCM encoder;

[0039] FIG. 9 illustrates an example of combining set-partitioning with iterative code according to an embodiment of the present invention;

[0040] FIG. 10A is a simplified illustration of a two-level encoding that includes an inner encoder and an outer encoder according to an embodiment of the present invention;

[0041] FIG. 10B is a simplified block diagram of a two-level encoding according to another exemplary embodiment of the present invention;

[0042] FIGS. 11A-11C show a number of exemplary sector and codeword sizes, in accordance with the present invention;

[0043] FIG. 12 illustrates an exemplary 3-way interleaved cell according to an embodiment of the present invention;

[0044] FIG. 13A is a simplified schematic diagram of an interleaving technique according to an exemplary embodiment of the present invention;

[0045] FIG. 13B is a plot of SER as a function of SNR for an interleaved system as illustrated in FIG. 13A; and

[0046] FIGS. 14A-14H show various devices in which the present invention may be embodied.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

[0047] FIG. 2A is a constellation diagram for an uncoded one bit PAM (pulse amplitude modulation) scheme according to which, information is stored as either a 0 (-1 volts) or a 1 (+1 volts). FIG. 2B is a constellation diagram for a two-bit data modulated using a PAM

scheme. The four states defined by the two bits, namely states (00, 01, 10, and 11) are mapped to one of four possible levels, for example, $-3/\sqrt{5}$ volts, $-1/\sqrt{5}$ volts, $+1/\sqrt{5}$ volts, and $+3/\sqrt{5}$ volts. Given these voltages, both the 2-points PAM (2-PAM) and the 4-points PAM (4-PAM) with 1/2 code rate provide a 1 bit/cell spectral efficiency and are characterized by equal power. For the modulation scheme illustrated by FIG. 2B, the 4-state code with Gray mapping reduces bit error rate over the 4-state natural mapping.

[0048] In solid state non-volatile memory devices, the maximum voltage applied at the floating gate limits the maximum voltage available for mapping of multi-level symbols. This voltage limitation results in a peak constraint on the constellation values of the applied modulation and encoding scheme. Thus, for a solid state non-volatile memory device, the constellation points must account for this limitation. The maximum voltage constraint characteristic of a solid state non-volatile memory unit constrasts with other channels in which additional power increases are available. Accordingly, embodiments of the present invention utilize modulation and encoding schemes for multi-level non-volatile solid state memories designed to be operable in spite of such constraints.

[0049] FIG. 2C is a constellation diagram for a two-bit data modulated using a PAM scheme having a peak value limited to ± 1 for use in a solid state non-volatile memory cell. Symbol 00 is mapped to a signal amplitude of -1 volts and symbol 10 is mapped to a signal amplitude of +1 volts, which in this example, correspond to the maximum voltages stored by a given solid state non-volatile memory cell. Applying power scaling to account for the maximum allowable voltage across the non-volatile solid state memory cells, it is seen that the squared free distance is reduced by a factor of 5/9, resulting in a 0 dB coding gain over an uncoded 2-PAM system.

[0050] As the number of levels increases, the signal to noise ratio (SNR) decreases. However, the slope of the SNR decrease is not the same for these two systems. Referring to FIGS. 2B and 2C, the SNR is related to the minimum distance between the nearest neighbors in the constellation map. Consequently, voltage constraints in solid state non-volatile memory devices reduce the available minimum distance and the SNR. Therefore, for a given number of levels, memory systems generally have a lower SNR compared to other known systems. As a result of these differences, the encoding and modulation techniques, in accordance with various embodiments of the present invention, are adapted to the environment characteristic of solid state non-volatile memories.

Ex.1002 / Page 245 of 744
Sandisk Technologies, Inc. et al

[0051] Assuming equal energy constellations, for an uncoded system, the error probability ($P_{uncoded}$) is upper bounded by :

$$P_{uncoded} \le A_{\min} Q\left(\sqrt{\frac{d_{\min}^2}{2N_0}}\right) \approx \frac{A_{\min}}{2} \exp\left(\frac{-d_{\min}^2}{4N_0}\right) \quad (1)$$

where A_{min} is the number of nearest neighbors, $d_{\min}^2$ is the minimum distance squared between two points in a constellation, and $Q(x)$ is the complementary error function (co-error function).

[0052] For a coded system:

$$P_{coded} \le A_{dfree} Q\left(\sqrt{\frac{d_{dfree}^2}{2N_0}}\right) \approx \frac{A_{dfree}}{2} \exp\left(\frac{-d_{dfree}^2}{4N_0}\right) \quad (2)$$

where d_{free} is the minimum distance.

[0053] The asymptotic coding gain is defined by:

$$\gamma = \frac{d_{dfree}^2}{d_{\min}^2} \quad (3)$$

[0054] For the coded system, the minimum distance d_{free} is generally increased in comparison to the uncoded system, resulting in an asymptotic coding gain greater than one. However, the number of nearest neighbors also increase so the real coding gain is somewhat reduced.

[0055] FIG. 3A is a simplified block diagram of an exemplary solid state non-volatile memory unit 300 with an ECC according to an embodiment of the present invention. User data to be written into the multi-level memory cells of multi-level solid state non-volatile memory 314 are ECC encoded by encoder 310 to add redundant symbols. The encoded data is passed to modulator 312 for channel encoding. According to embodiments of the present invention, any one of a number of encoding and modulation techniques may be used.

[0056] Multi-level solid state non-volatile memory 314 receives encoded and modulated data from modulator 312. Multi-level solid state non-volatile memory 314 may be a flash EEPROM, or the like. Generally, a multi-level flash memory includes one or more flash memory cell arrays and read and write (program) circuitry. In addition to a multi-level flash memory, there are other types of solid state non-volatile memory technologies that are included within the scope of the present invention. Floating gate memory cells such as flash

Ex.1002 / Page 246 of 744
Sandisk Technologies, Inc. et al

memories are discussed herein merely by way of example. Embodiments of the present invention as described throughout the present specification also apply to other memory technologies other than floating gate technology with the appropriate modifications. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0057] Data is read from multi-level solid state non-volatile memory 314 and passes to demodulator 316 and decoder 318. The demodulator in embodiments of the present invention includes signal processing logic adapted to extract soft information related to the data stored in the multi-level solid state non-volatile memory 314. In conventional binary non-volatile memory units, a threshold detector is utilized to determine if a voltage value associated with a particular cell is less than or greater than a given threshold value. This threshold-based approach is also utilized in multi-level memory units, where the threshold detection circuit merely utilizes a greater number of threshold values. Although some multi-level memory units include circuitry that tracks process or other variations and adjusts the threshold values accordingly, these units output a value from a number of possible values that is equal to the number of levels in the multi-level unit. Thus, for example, for a conventional four-level non-volatile memory, the sense amplifier will produce a data signal representing one of the four levels.

[0058] In contrast with conventional multi-level memory units, embodiments of the present invention utilize a demodulator that produces an output having a number of possible values greater than the number of levels provided by the multi-level solid state non-volatile memory 314. This information is sometimes referred to as soft information since the output includes information in addition to the value stored in the multi-level memory. Merely by way of example, an analog-to-digital (A/D) converter or detector provided as part of or working in conjunction with demodulator 316 provides an output signal with, for example, 32 possible values in response to the values detected by a sense amplifier disposed in communication with, for example, a four-level memory. Embodiments of the present invention are not limited to using an output signal with 32 levels, as other output signals, with, for example, 8, 16 or more levels are included within the scope of the present invention. In some applications, the soft information is passed to a soft input decoder (not shown) for processing. In accordance with the present invention, the soft information provided by the demodulator 316 is utilized during signal processing operations to improve the reliability with which data from the solid state non-volatile memory 314 is read.

[0059] In an embodiment of the present invention, encoder 310, modulator 312, demodulator 316, and decoder 318 are components of a controller in communication with multi-level solid state non-volatile memory 314. Memory devices typically include one or more memory chips that are mounted on a card. Each of the memory chips may include an array of memory cells as well as integrated circuits performing such operations as program, read, and erase. According to embodiments of the present invention, a controller circuit performing these operations may or may not be disposed in the integrated circuits (IC) in which the memory chip (s) are also disposed. Controllers provided herein are not limited to performing encoding/decoding and modulation/demodulation processes, but may also provide for other functionality such as wear leveling and interfacing processes.

[0060] Embodiments of the present invention enable designers to increase the memory density of existing solid state non-volatile memories. As described more fully throughout the present specification, in comparison with conventional units, increased levels of read and write errors are corrected utilizing the techniques and methods provided herein. Thus, although attempting to utilize, for example, a four-level memory system for, for example, eight-level applications by introducing additional levels may produce a greater number of errors during read operations than is otherwise acceptable under a given performance measure, techniques described herein may be utilized to correct such errors during the demodulation process and enable the use, for instance, of a four-level memory system in an eight-level application. Thus, in accordance with the present invention, the memory density of existing solid state non-volatile memory systems may be increased while still using some of the same components, including memory arrays, sense amplifiers, and the like.

[0061] FIG. 3B is a simplified block diagram of a solid state non-volatile memory unit 350 incorporating an A/D converter according to an embodiment of the present invention. As illustrated in FIG. 3B, an encoder 360 and a modulator 362 provide encoded and modulated data to the multi-level non-volatile solid state memory 364. A/D converter 366 receives signals from the multi-level non-volatile solid state memory 364. The digital signal output by the A/D converter 366 is of higher resolution (characterized by more levels) than the number of levels associated with the multi-level non-volatile solid state memory 364. Merely by way of example, in a particular embodiment, the multi-level non-volatile solid state memory 364 is, for example, a four-level memory, providing storage for 2 bits in each cell. During a read operation, the A/D converter 366 converts an analog signal associated with one or more cells of the memory 364 into one of, for example, 8, 16, 32, or 64 levels depending on the

particular application. Other numbers of levels greater than four levels are used in other embodiments. Signal processing algorithms resident in demodulator 368 utilize the output of the A/D converter 366 to determine the likelihood that the cell contains data associated with one of the four levels stored in the cell. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0062] In conventional solid state memories, ECC techniques are utilized to detect and correct errors in data as the data is read out from the memory. Such ECC techniques simply operate on binary or multi-level digital data produced by a sense amplifier. On the other hand, in accordance with the present invention, soft information produced and utilized by the demodulator 316 does not merely include the binary or multi-level digital data, but additional information as well. Soft information is typically represented by distributions that are useful in performing signal processing techniques not generally applicable once the data has been reduced to threshold-based digital values equal in number to the number of levels in the multi-level system.

[0063] Utilizing embodiments of the present invention, positive coding gain is achieved for multi-level solid state non-volatile memory systems in comparison to uncoded systems. Table 1 illustrates an exemplary coding gain from convolutional coded 2bit/cell multi-level non-volatile memory over an uncoded 1 bit/cell non-volatile memory as a function of the number of states of the code. As shown in the first row entry, for a 4-state system (illustrated by FIGS. 2A-2C), the coding gain in comparison to an uncoded system is 0 dB. However, as the number of states is increased, the coding gain in comparison to an uncoded system is positive.

Table 1

Number of States	Coding Gain (dB)
4	0
8	0.46
16	0.87
32	1.50

Ex.1002 / Page 249 of 744
Sandisk Technologies, Inc. et al

[0064] Table 2 illustrates the coding gain for convolutional coded 3bit/cell multi-level non-volatile memory over uncoded 2 bit/cell multi-level non-volatile memory as a function of the number of states of the code. As shown for 1 bit/cell multi-level non-volatile memory with four or more states, the coding gain in comparison to an uncoded system is positive for systems with four or more states.

Table 2

Number of States	Coding Gain (dB)
4	2.18
8	2.64
16	3.05
32	3.78

[0065] FIG. 4 is a simplified block diagram of a solid state non-volatile memory unit with error correction according to another embodiment of the present invention. As illustrated in FIG. 4, an outer encoder 410 provides encoded data to an inner encoder 412. As an example, outer encoder 410 may be a Reed-Solomon encoder and inner encoder may be a LDPC encoder. These encoding techniques are used merely as examples and are not intended to limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, modifications, and alternatives. Modulator 414 receives encoded data from inner encoder 412 and modulates the data prior to storage in multi-level solid state non-volatile memory 416 during a program operation. During a read operation, the data stored in the multi-level solid state non-volatile memory 416 is retrieved and provided to demodulator 418, inner decoder 420, and outer decoder 422.

[0066] Any number of Error Correcting Codes (ECCs) including Forward Error Correction (FEC) codes may be used according to embodiments of the present invention to improve the bit error rate (BER) performance of power-limited and/or bandwidth-limited channels by adding structured redundancy to the transmitted data. For example, block codes may be used to encode a block of data for channels with additive burst noise (random multi-bit errors). It is understood that the present invention is applicable to both systematic encoders that do not manipulate the user data prior to encoding and storage, as well as to non-systematic encoders.

[0067] Any one of a number of different linear block codes including, for example, binary codes such as Hamming code, BCH code, Reed-Muller code and Array Code, and non-binary codes such as Reed-Solomon (RS) code may be used. Choice of block size depends on SNR and the code used. For example, assume that the voltage levels are increased from 4 to 8 per cell, and that each three cells are grouped together to form a 9-bit symbol. Applying a (511, 451) Reed-Solomon code, based on GF(2^9), the codeword length is 511 *9 = 4599 bits, and the code rate is 451/511~=0.883. Therefore, the storage capacity for the coded system is 3*451/511~=2.6 bits/cell, which represents a 32% capacity increase over uncoded 4-level system. The word error rate (WER) of such a coded system is compared to a 4-level uncoded system in FIG. 5. It can be seen at WER<10^{-8}, the RS coded system outperforms the uncoded system. Therefore, with the above RS code, better reliability and higher capacity is achieved.

[0068] In accordance with other exemplary embodiments of the present invention, for example, when the noise is independent from symbol to symbol, convolutional codes are used to encode the data. Convolutional codes introduce correlation into coded data and thus increase the minimum distance at the decoder. Convolution codes are applied to serially-transmitted data stored in or read from solid state non-volatile memories, which are subject to Gaussian noise. Convolution codes are progressive codes. At any point in time, the output of a convolutional encoder may depend upon both the past and present input values. Thus, convolution codes are generally directed to correcting errors that span an ordered progression of data values. Accordingly, such codes may be used in multi-level solid state non-volatile memories that store and read out data in the form of an ordered, progressive stream (i.e., a data stream).

[0069] The decoder receives either hard decision inputs or multi-level quantized inputs. Soft inputs are known to cause fewer errors at the detector. FIG. 6 shows various blocks of an exemplary rate 1/2 convolutional encoder 600. Output C2 generated by modulo-2 adder 610 is defined by shift registers 602, 606 and input U. For example, if the voltage levels are increased from four to eight, applying a rate 3/4 convolutional code to get 3*3/4=2.25 bits/cell, results in a 10% increase in storage capacity. To achieve the same error rate as uncoded 4-level system, the free distance of the convolutional code must be greater than $(7/3)^2$=5.44. A 3/4 convolutional encoder with six memory units would require a Viterbi decoder with 2^6=64 states.

[0070] Trellis coded modulation (TCM) combines convolutional code with set partitioning to achieve high code rate, high coding gain, and low decoding complexity. A set of constellation points can be partitioned to a smaller subset, where points in each subset are separated further than in the original constellation. FIG. 7 shows an example of 2-D set partitioning where dots, "•", represent one subset, and crosses, "x", represent the other. If the minimum distance between the original constellation points is d, then the minimum distance between points in each subset is $\sqrt{2}d$. In systems with additive white Gaussian Noise (AWGN) channels, such as flash memory systems read path channels, the parameter governing the performance of the system is not the free Hamming distance of the convolutional code, but rather the free Euclidean distance between the transmitted signal sequences. Accordingly, the optimization of the TCM design is based on the Euclidean distances rather than the Hamming distances.

[0071] An example of TCM 800 is shown in FIG. 8. Input bits are separated into two groups -- a first group with k_1 bits, going through a rate $k_1/(k_1+1)$ encoder 802 to select subsets; a second group with $k-k_1$ bits, which select constellation points, using constellation mapper 804, within each subset. Below is a description of an exemplary increase in capacity from 2bits/cell to 2.5bits/cell. Assume that there are 8 voltage levels and that each two adjacent cells is combined to form a 64 QAM constellation. The 64 QAM is partitioned into 4 cosets. Distance between any two points in each coset is 8*d0. Assume a rate 3/ 4 convolutional code is used to select a coset, and two uncoded bits are used to select a point within any given coset. The overall code rate would thus become 5/6, leading to 2.5 bits/cell. The overall coding gain is 0.43 dB. Therefore, such a TCM coded system has a better performance than the uncoded 4 level system while increasing the storage capacity by 25%.

[0072] Some codes based on random construction may be effectively decoded by iterative detection methods. These codes include Turbo codes, i.e., serially concatenated convolutional codes, or parallel concatenated convolutional codes, low-density parity-check (LDPC) codes, Turbo Product code, and their variations.

[0073] Coding gain of TCM comes from two areas -- set partitioning to increase the distance between constellation points within each subset, and convolutional code to achieve high Euclidean distance between different subsets, the latter of which can also be achieved if other high gain codes are substituted for convolutional code. An example of combining set-partitioning with iterative code is shown in FIG. 9. Assume a 16-PAM system is partitioned

into 4 subsets. Then the minimum distance between points in each subset is $4d_0$, hence providing a 12dB gain compared to uncoded system. However, between different subsets, the minimum distance remains at d_0. Since the selection of subset depends on LDPC coded bits supplied by LDPC encoder 902, then the overall system has an approximate gain of 12dB if an LDPC code having a gain of 12dB is used. In general, the overall system gain is the minimum of the set-partitioning gain and the iterative coding gain. LDPC encoder 902 shown in FIG. 9 is required to operate on a whole codeword defined by the iterative code block size. In addition to iterative codes, other high gain codes including RS code and BCH code may be used to code a portion of the input for subset selection.

[0074] In accordance with some embodiments, to further improve coding gain, multi-level coding may be used. A two-level encoding that includes an inner encoder and an outer encoder is shown in FIG. 10A. In one exemplary embodiment, outer code encoder 1002 may be a RS encoder, and inner code encoder 1004 may be a TCM encoder. Bursty errors caused by TCM decoder 1006 are corrected by RS decoder 1008.

[0075] In another exemplary embodiment, inner encoder 1004 is adapted to perform iterative codes, such as LDPC codes or Turbo codes, and outer encoder 1002 is adapted to perform RS code. Iterative codes may be binary codes or symbol-based codes. Each symbol may contain multiple bits. Iterative codes may be decoded using a soft-input soft-output (SISO) decoder, while RS codes may be decoded using either a SISO or a hard-decision decoder. The outer RS decoder 1008 may iterate with inner decoder 1006 to exchange soft information. Such iteration would improve the quality of soft information and thereby decrease the number of bit errors after each iteration.

[0076] The descriptions of the various embodiments provided herein are provided merely by way of example, and are not intended to limit the scope of the present invention. Various other coding techniques, interleaving techniques, modulation techniques, demodulation techniques, decoding techniques, mapping techniques, and the like are included within the scope of the present invention.

[0077] FIG. 10B is a simplified block diagram of a two-level encoding 1020, in accordance with another exemplary embodiment of the present invention. The outer encoder 1022 is a Reed-Solomon encoder with a correction power *t*. Inner encoder 1030 includes a TCM 1024 having a code rate 3/4, and a 16-PAM constellation mapper 1026 having a spacing between any two points of 2/15. For calculations performed for the encoding channel shown in FIG.

Ex.1002 / Page 253 of 744
Sandisk Technologies, Inc. et al

10B, convolutional encoders with polynomials of order 3, 4, and 5 are utilized. It is understood that although the exemplary embodiment shown in FIG. 10B includes a Reed-Solomon encoder 1022, a TCM 1024, and a 16-PAM constellation mapper 1026, other embodiments of the present invention may include other encoders, modulators, and mappers. Moreover, for all the exemplary embodiments described herein and shown in the drawings, modulators including multi-dimensional ones such as those proposed by Wei, "Trellis-Coded Modulation with Multidimensional Constellations" IEEE Transactions on Information Theory, Volume IT-33, No. 4, (July 1987), pgs. 483-501; and multilevel codes such as those proposed by Imai and Hirakawa, "A New Multilevel Coding Method Using Error-Correcting Codes" IEEE Transactions on Information Theory, Volume IT-23, No. 3, (May 1977) pgs. 371-377, any combinations thereof, and the like may be used.

[0078] Referring to FIG. 10B, approximately 2 kbits (2048 bits) are represented by 228 9-bit symbols. Merely by way of example, the 9-bit symbols are formed by combining three adjacent 8-level cells. The 228 9-bit symbols are supplemented with $2t$ symbols by the Reed-Solomon encoder 1022 to provide 230 symbols that are input to the inner encoder 1030. In an exemplary embodiment, user data represented by the 228 9-bit symbols is written into a portion of a memory array characterized by a width equal to thirty 3-bit cells. After the user data is written into the memory array in a serial manner, three adjacent 3-bit cells are grouped to form a 9-bit symbol. The ten columns of such 9-bit symbols are then appended with column parity values provided, for example, by the RS encoder 1022. The number of appended parity values will depend, in part, on the correction power selected for the RS encoder 1022. TCM encoding is performed on each 3-bit cell to provide encoded 4-bit symbols, which are subsequently provided to the 16-PAM constellation mapper 1026 and then written to the solid state non-volatile memory. It will be appreciated that encoding may also be performed in parallel, for example, by providing a number of TCMs 1024 operating in parallel , thereby increasing the processing speed. Decoding of user data is performed by reversing the operations discussed in relation to FIG. 10B.

[0079] Depending on the encoding technique selected, one of several generator polynomials may be used. Merely by way of example, generator polynomials as illustrated in Table 3 are used with systematic encoders with a rate 1/2 code for some applications. The number of branches per state is equal to two.

Table 3

Generator Polynomials	
Order	Polynomial
2	[1 D/D^2+1]
3	[1 D^2/D^3+D+1]
4	[1 D^2/D^4+D+1]
5	[1 D^2/D^5+D^2+1]

[0080] Set partitioning for Level 1 is:

Q(0) - {-15;-11,-7,-3,+1,+,5+9,+13}
Q(1) - {-13,-9,-5,-1,+3,+7,+11,+15}

[0081] Set partitioning for Level 2 is:

Q(00) - {-15,-7,+1,+9},
Q(10) - {-11,-3,+5,+13}
Q(01) - {-13,-5,+3,+11}
Q(11) - {-9,-1,+7,+15}

[0082] As illustrated in FIG. 10B, some embodiments of the present invention utilize an RS encoder as outer encoder 1022. RS encoders provide a functionality of being well suited for applications in which bursts of errors are present. For solid state non-volatile memory applications, errors may occur in bursts for several reasons. First, defects in a portion of a memory array media may impact the error caused by the cells disposed in such portions. Additionally, error bursts may result from operation of the inner encoder 1024. A convolutional decoder may produce bursts of errors since the output at any given time depends, in part, on previous outputs. Thus, some embodiments of the present invention utilize RS encoders and decoders suited for handling error bursts.

[0083] As seen from FIG. 1, the PDF of the programmed cells is different depending on the cell's threshold voltage. If the four levels are spaced equally, then the level corresponding to cells with a PDF defined by a threshold voltage ranging between 2 and 4 volts is more prone to errors than other levels due to its wider distribution. Accordingly, in some embodiments, constrained coding is used to inhibit certain patterns or reduce their frequency. For example, lowering the frequency of data corresponding to cells with a PDF

defined by a threshold voltage ranging between 2 and 4 volts lowers the overall error probability.

[0084] In some embodiments, the codeword size is aligned with the sector size. For example, if the sector size is, for example, 256 kbits, the inner code and outer code may be configured such that one outer codeword is 256 kbits. Smaller or larger codeword sizes relative to the sector size may also be used. In FIG. 11A, the sector size is shown as being equal to the codeword size. In the case of smaller codeword size, each sector includes several codewords, as shown in FIG. 11B. In the case of larger codeword size, each codeword includes several sectors, as shown in FIG. 11C. In general, the larger the codeword size, the larger the coding gain, the longer decoding delay, and the higher is the decoder complexity.

[0085] Codewords may be interleaved before being stored. FIG. 12 shows an exemplary 3-way interleaved cell in which cells 1202 form codeword 1, cells 1204 form codeword 2, and cells 1206 form codeword 3. If a defect spans no more than three cells, it causes only one symbol error in each codeword, which is easier to correct than a burst of three symbol errors.

[0086] In accordance with other exemplary embodiments of the present invention, coding of data, as described above may be applied across a multitude of non-volatile solid-state semiconductor memories, that in some embodiments are physically stacked on top of one another. For example, if 8 such non-volatile solid-state semiconductor memories are stacked together, a $GF(2^8)$-based RS code may be applied across these memories, where each bit of a RS code symbol comes from one of these memories. Coding across such memories improves error recovery in the event one of these memories has large defects.

[0087] FIG. 13A is a simplified schematic diagram of an interleaving technique provided according to an exemplary embodiment of the present invention. As illustrated in FIG. 13A, an inner encoder, for example, a TCM encoder, is utilized on rows of data and an outer encoder, for example, an RS encoder, is utilized on block columns of data. This exemplary embodiment may be used, for example, in solid state non-volatile memories, in which the data is written to the memory cells in a rectangular format, i.e., blocks. In situations in which a multi-bit error of significant length is present on the inner TCM code, embodiments of the present invention provide that several independent outer RS codes process data corrupted by the multi-bit error. Accordingly, the number of errors impacting an individual RS code is limited. Referring to FIG. 13A, the number of columns in a particular implementation is

determined, in part, by the maximum error burst length. The number of rows is determined, in part, by the number of sectors per block codeword.

[0088] According to some embodiments of the present invention, the number of columns is predetermined depending on the particular application. For example, if the number of columns (related to the interleaving depth) is larger than the maximum error burst length, then the outer encoder operating on the columns is similar to a memoryless channel characterized by a particular symbol error rate. Accordingly, error bursts that occur affect different outer encoder codes. The symbol error rate is typically determined by using TCM simulations independently, and the error probability may be estimated by independent modeling. The percentage of overhead of the outer encoder, for example, an RS encoder, may be reduced by increasing the row dimension of the block codeword. Alternatively, one can increase the row dimension of the RS code while keeping the overhead percentage constant, thereby allowing for a higher correction power per column.

[0089] FIG. 13B is a plot of SER as a function of SNR for an interleaved system as illustrated in FIG. 13A. To compute the data presented in FIG. 13B, 10 columns per block codeword were utilized. Other embodiments will utilize a varying number of columns and sectors per block codeword depending on the particular application. The SER for an uncoded 2 bits/cell 4-PAM system is illustrated for purposes of comparison. The SER values for implementations in which the strength of the outer encoder (an RS encoder in this example) is varied over a range of correction powers are shown (t_{RS}=12, 14, and 16, respectively). As the correction power or strength of the outer encoding increases, the SNR at which the coded system drops to a level equal to the uncoded system decreases. Referring to FIG. 13B, this cross-over point is at approximately 22.4 dB, 22.2 dB, and 22.0 dB for t_{RS}=12, 14, and 16, respectively.

[0090] In a particular exemplary embodiment, a multi-level solid state non-volatile memory includes, for example, 2.5 bits/cell. In such embodiments, two adjacent 8-level cells (3 bits/cell) form a 64-QAM modulation symbol. Of the six bits in the 64-QAM modulation symbol, five bits are utilized for data and one bit is used for encoding. Thus, in such exemplary embodiments, the code rate is 5/6 and the number of branches per state is equal to four. Such a system provides 2.5 bits/cell as 5 data bits are stored between two adjacent cells. In such exemplary embodiments, the coding gain in comparison with an uncoded 4-PAM system may be, for example, 0.423 dB for 16 states. It should be noted that calculation

results will be modified as multiplicities are included in such calculations. For example, losses of approximately 0.2 dB are expected with the doubling of multiplicities. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0091] In another particular exemplary embodiment of the present invention, multi-level solid state non-volatile memory systems with, for example, 3.5 bits/cell are provided. In such embodiments, two adjacent 16-level cells (4 bits/cell) form a 256-QAM modulation symbol. Of the eight bits in the 256-QAM modulation symbol, seven bits are utilized for data and one bit is used for encoding. Thus, in such exemplary embodiments the code rate is 7/8 and the number of branches per state is equal to four. Such a system provides 3.5 bits/cell as 7 data bits are stored between two adjacent cells. The coding gain in comparison with an uncoded 8-PAM system may be, for example, 0.527 dB for 8 states and 1.317 dB for 16 states. It should be noted that calculation results will be modified as multiplicities are included in such calculations. For example, losses of approximately 0.2 dB are expected with the doubling of multiplicities. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0092] Referring now to FIGS. 14A-14H, various exemplary implementations of the present invention are shown. Referring to FIG. 14A, the present invention may be embodied in a hard disk drive 1400. The present invention may implement either or both signal processing and/or control circuits, which are generally identified in FIG. 14A at 1402. In some implementations, signal processing and/or control circuit 1402 and/or other circuits (not shown) in HDD 1400 may process data, perform coding and/or encryption, perform calculations, and/or format data that is output to and/or received from a magnetic storage medium 1406.

[0093] HDD 1400 may communicate with a host device (not shown) such as a computer, mobile computing devices such as personal digital assistants, cellular phones, media or MP3 players and the like, and/or other devices via one or more wired or wireless communication links 1408. HDD 1400 may be connected to memory 1409, such as random access memory (RAM), a low latency nonvolatile memory such as flash memory, read only memory (ROM) and/or other suitable electronic data storage.

[0094] Referring now to FIG. 14B, the present invention may be embodied in a digital versatile disc (DVD) drive 1410. The present invention may implement either or both signal processing and/or control circuits, which are generally identified in FIG. 14B at 1412, and/or

Ex.1002 / Page 258 of 744
Sandisk Technologies, Inc. et al

mass data storage 1418 of DVD drive 1410. Signal processing and/or control circuit 1412 and/or other circuits (not shown) in DVD 1410 may process data, perform coding and/or encryption, perform calculations, and/or format data that is read from and/or data written to an optical storage medium 1416. In some implementations, signal processing and/or control circuit 1412 and/or other circuits (not shown) in DVD 1410 can also perform other functions such as encoding and/or decoding and/or any other signal processing functions associated with a DVD drive.

[0095] DVD drive 1410 may communicate with an output device (not shown) such as a computer, television or other device via one or more wired or wireless communication links 1417. DVD 1410 may communicate with mass data storage 1418 that stores data in a nonvolatile manner. Mass data storage 1418 may include a hard disk drive (HDD) such as that shown in FIG. 14A. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". DVD 1410 may be connected to memory 1419, such as RAM, ROM, low latency nonvolatile memory such as flash memory, and/or other suitable electronic data storage.

[0096] Referring now to FIG. 14C, the present invention may be embodied in a high definition television (HDTV) 1420. The present invention may implement either or both signal processing and/or control circuits, which are generally identified in FIG. 14C at 1422, a WLAN interface and/or mass data storage of the HDTV 1420. HDTV 1420 receives HDTV input signals in either a wired or wireless format and generates HDTV output signals for a display 1426. In some implementations, signal processing circuit and/or control circuit 1422 and/or other circuits (not shown) of HDTV 1420 may process data, perform coding and/or encryption, perform calculations, format data and/or perform any other type of HDTV processing that may be required.

[0097] HDTV 1420 may communicate with mass data storage 1427 that stores data in a nonvolatile manner such as optical and/or magnetic storage devices. At least one HDD may have the configuration shown in FIG. 14A and/or at least one DVD may have the configuration shown in FIG. 14B. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". HDTV 1420 may be connected to memory 1428 such as RAM, ROM, low latency nonvolatile memory such as flash memory and/or other suitable electronic data storage. HDTV 1420 also may support connections with a WLAN via a WLAN network interface 1429.

Ex.1002 / Page 259 of 744
Sandisk Technologies, Inc. et al

[0098] Referring now to FIG. 14D, the present invention implements a control system of a vehicle 1430, a WLAN interface and/or mass data storage of the vehicle control system. In some implementations, the present invention implements a powertrain control system 1432 that receives inputs from one or more sensors such as temperature sensors, pressure sensors, rotational sensors, airflow sensors and/or any other suitable sensors and/or that generates one or more output control signals such as engine operating parameters, transmission operating parameters, and/or other control signals.

[0099] The present invention may also be embodied in other control systems 1440 of vehicle 1430. Control system 1440 may likewise receive signals from input sensors 1442 and/or output control signals to one or more output devices 1444. In some implementations, control system 1440 may be part of an anti-lock braking system (ABS), a navigation system, a telematics system, a vehicle telematics system, a lane departure system, an adaptive cruise control system, a vehicle entertainment system such as a stereo, DVD, compact disc and the like. Still other implementations are contemplated.

[0100] Powertrain control system 1432 may communicate with mass data storage 1446 that stores data in a nonvolatile manner. Mass data storage 1446 may include optical and/or magnetic storage devices for example hard disk drives HDD and/or DVDs. At least one HDD may have the configuration shown in FIG. 14A and/or at least one DVD may have the configuration shown in FIG. 14B. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". Powertrain control system 1432 may be connected to memory 1447 such as RAM, ROM, low latency nonvolatile memory such as flash memory and/or other suitable electronic data storage. Powertrain control system 1432 also may support connections with a WLAN via a WLAN network interface 1448. The control system 1440 may also include mass data storage, memory and/or a WLAN interface (all not shown).

[0101] Referring now to FIG. 14E, the present invention may be embodied in a cellular phone 1450 that may include a cellular antenna 1451. The present invention may implement either or both signal processing and/or control circuits, which are generally identified in FIG. 14E at 1452, a WLAN interface and/or mass data storage of the cellular phone 1450. In some implementations, cellular phone 1450 includes a microphone 1456, an audio output 1458 such as a speaker and/or audio output jack, a display 1460 and/or an input device 1462 such as a keypad, pointing device, voice actuation and/or other input device. Signal processing

and/or control circuits 1452 and/or other circuits (not shown) in cellular phone 1450 may process data, perform coding and/or encryption, perform calculations, format data and/or perform other cellular phone functions.

[0102] Cellular phone 1450 may communicate with mass data storage 1464 that stores data in a nonvolatile manner such as optical and/or magnetic storage devices for example hard disk drives HDD and/or DVDs. At least one HDD may have the configuration shown in FIG. 14A and/or at least one DVD may have the configuration shown in FIG. 14B. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". Cellular phone 1450 may be connected to memory 1466 such as RAM, ROM, low latency nonvolatile memory such as flash memory and/or other suitable electronic data storage. Cellular phone 1450 also may support connections with a WLAN via a WLAN network interface 1468.

[0103] Referring now to FIG. 14F, the present invention may be embodied in a set top box 1480. The present invention may implement either or both signal processing and/or control circuits, which are generally identified in FIG. 14F at 1484, a WLAN interface and/or mass data storage of the set top box 1480. Set top box 1480 receives signals from a source such as a broadband source and outputs standard and/or high definition audio/video signals suitable for a display 1488 such as a television and/or monitor and/or other video and/or audio output devices. Signal processing and/or control circuits 1484 and/or other circuits (not shown) of the set top box 1480 may process data, perform coding and/or encryption, perform calculations, format data and/or perform any other set top box function.

[0104] Set top box 1480 may communicate with mass data storage 1490 that stores data in a nonvolatile manner. Mass data storage 1490 may include optical and/or magnetic storage devices for example hard disk drives HDD and/or DVDs. At least one HDD may have the configuration shown in FIG. 14A and/or at least one DVD may have the configuration shown in FIG. 14B. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". Set top box 1480 may be connected to memory 1494 such as RAM, ROM, low latency nonvolatile memory such as flash memory and/or other suitable electronic data storage. Set top box 1480 also may support connections with a WLAN via a WLAN network interface 1496.

[0105] Referring now to FIG. 14G, the present invention may be embodied in a media player 1472. The present invention may implement either or both signal processing and/or

Ex.1002 / Page 261 of 744
Sandisk Technologies, Inc. et al

control circuits, which are generally identified in FIG. 14G at 1471, a WLAN interface and/or mass data storage of the media player 1472. In some implementations, media player 1472 includes a display 1476 and/or a user input 1477 such as a keypad, touchpad and the like. In some implementations, media player 1472 may employ a graphical user interface (GUI) that typically employs menus, drop down menus, icons and/or a point-and-click interface via display 1476 and/or user input 1477. Media player 1472 further includes an audio output 1475 such as a speaker and/or audio output jack. Signal processing and/or control circuits 1471 and/or other circuits (not shown) of media player 1472 may process data, perform coding and/or encryption, perform calculations, format data and/or perform any other media player function.

[0106] Media player 1472 may communicate with mass data storage 1470 that stores data such as compressed audio and/or video content in a nonvolatile manner. In some implementations, the compressed audio files include files that are compliant with MP3 format or other suitable compressed audio and/or video formats. The mass data storage may include optical and/or magnetic storage devices for example hard disk drives HDD and/or DVDs. At least one HDD may have the configuration shown in FIG. 14A and/or at least one DVD may have the configuration shown in FIG. 14B. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". Media player 1472 may be connected to memory 1473 such as RAM, ROM, low latency nonvolatile memory such as flash memory and/or other suitable electronic data storage. Media player 1472 also may support connections with a WLAN via a WLAN network interface 1474.

[0107] Referring to FIG. 14H, the present invention may be embodied in a Voice over Internet Protocol (VoIP) phone 1483 that may include an antenna 1439. The present invention may implement either or both signal processing and/or control circuits, which are generally identified in FIG. 14H at 1482, a wireless interface and/or mass data storage of the VoIP phone 1483. In some implementations, VoIP phone 1483 includes, in part, a microphone 1487, an audio output 1489 such as a speaker and/or audio output jack, a display monitor 1491, an input device 1492 such as a keypad, pointing device, voice actuation and/or other input devices, and a Wireless Fidelity (Wi-Fi) communication module 1486. Signal processing and/or control circuits 1482 and/or other circuits (not shown) in VoIP phone 1483 may process data, perform coding and/or encryption, perform calculations, format data and/or perform other VoIP phone functions.

[0108] VoIP phone 1483 may communicate with mass data storage 1402 that stores data in a nonvolatile manner such as optical and/or magnetic storage devices, for example hard disk drives HDD and/or DVDs. At least one HDD may have the configuration shown in FIG. 14A and/or at least one DVD may have the configuration shown in FIG. 14B. The HDD may be a mini HDD that includes one or more platters having a diameter that is smaller than approximately 1.8". VoIP phone 1483 may be connected to memory 1485, which may be a RAM, ROM, low latency nonvolatile memory such as flash memory and/or other suitable electronic data storage. VoIP phone 1483 is configured to establish communications link with a VoIP network (not shown) via Wi-Fi communication module 1486. Still other implementations in addition to those described above are contemplated.

[0109] The above embodiments of the present invention are illustrative and not limitative. Various alternatives and equivalents are possible. The invention is not limited by the type of comparator, counter, pulse-width modulator, driver, or filter used. The invention is not limited by the type of amplifier used to establish the reference charging and discharging currents. The invention is not limited by the of oscillator. The invention is not limited by the type of integrated circuit in which the present disclosure may be disposed. Nor is the invention limited to any specific type of process technology, e.g., CMOS, Bipolar, or BICMOS that may be used to manufacture the present disclosure. Other additions, subtractions or modifications are obvious in view of the present disclosure and are intended to fall within the scope of the appended claims.

Ex.1002 / Page 263 of 744
Sandisk Technologies, Inc. et al

WHAT IS CLAIMED IS:

1. A solid state non-volatile memory unit comprising:
a first encoder;
a multi-level solid state non-volatile memory array adapted to store data encoded by the first encoder; and
a first decoder adapted to decode data retrieved from the memory array.

2. The solid state non-volatile memory unit of claim 1 wherein the memory array is a flash EEPROM array.

3. The solid state non-volatile memory unit of claim 1 further comprising:
a second encoder in communication with the first encoder; and
a second decoder in communication with the first decoder.

4. The solid state non-volatile memory unit of claim 3 wherein the first encoder is adapted to execute a first ECC different from a second ECC executed by the second encoder.

5. The solid state non-volatile memory unit of claim 3 wherein the second encoder comprises a Reed-Solomon encoder.

6. The solid state non-volatile memory unit of claim 1 further comprising:
a modulator adapted to modulate the encoded data; and
a demodulator adapted to demodulate the modulated data retrieved from the memory array.

7. The solid state non-volatile memory unit of claim 1 wherein said first encoder is a binary encoder.

8. The solid state non-volatile memory unit of claim 7 wherein said binary encoder encodes in accordance with a binary code selected from a group consisting of Hamming code, BCH code, Reed-Muller code and Array Code.

9. The solid state non-volatile memory unit of claim 1 wherein said first encoder is selected from a group consisting of non-binary encoder and convolutional encoder.

Ex.1002 / Page 264 of 744
Sandisk Technologies, Inc. et al

10. The solid state non-volatile memory unit of claim 1 wherein said first encoder is disposed in a trellis-coded modulator.

11. The solid state non-volatile memory unit of claim 1 wherein said first encoder is an iterative encoder.

12. The solid state non-volatile memory unit of claim 11 wherein said iterative encoder encodes in accordance with a code selected from a group consisting of low-density parity check code and Turbo code.

13. The solid state non-volatile memory unit of claim 11 wherein said iterative encoder is a symbol-based iterative encoder.

14. The solid state non-volatile memory unit of claim 1 wherein the solid state non-volatile memory unit is disposed in an integrated circuit.

15. A method of operating a solid state non-volatile memory unit, the method comprising:

encoding a first data;

storing the encoded first data in a multi-level solid state non-volatile memory array;

retrieving the first encoded data from the memory array; and

decoding the first data retrieved from the memory array.

16. The method of claim 15 wherein the memory array is a flash EEPROM array.

17. The method of claim 15 further comprising:

encoding a second data to generate the first data; and

decoding the decoded first data to generate the second data.

18. The method of claim 17 wherein the encoding of the first data is performed in accordance with a first ECC different from a second ECC in accordance with which the second data is encoded.

19. The method of claim 18 wherein the second ECC is a Reed-Solomon code.

Ex.1002 / Page 265 of 744
Sandisk Technologies, Inc. et al

20. The method of claim 15 further comprising:
modulating the encoded data;
storing the modulated data in the memory array;
retrieving the modulated data from the memory array; and
demodulating the data retrieved from the memory array.

21. The method of claim 15 further comprising:
encoding the first data in accordance with a binary code.

22. The method of claim 21 wherein said binary code is selected from a group consisting of Hamming code, BCH code, Reed-Muller code and Array Code.

23. The method of claim 15 further comprising:
encoding the first data in accordance with one of non-binary code and convolutional code.

24. The method of claim 20 wherein said encoding comprises trellis-coded modulation.

25. The method of claim 15 wherein said encoding is an iterative encoding.

26. The method of claim 25 wherein said iterative encoding is performed in accordance with a code selected from a group consisting of low-density parity check code and Turbo code.

27. The method of claim 25 wherein said iterative encoding is a symbol-based iterative encoding.

28. A solid state non-volatile memory unit comprising:
means for encoding a first data;
means for storing the encoded first data in a multi-level solid state non-volatile memory array;
means for retrieving the first encoded data from the memory array; and
means for decoding the first data retrieved from the memory array.

Ex.1002 / Page 266 of 744
Sandisk Technologies, Inc. et al

29. The solid state non-volatile memory unit of claim 28 wherein the memory array is a flash EEPROM array.

30. The solid state non-volatile memory unit of claim 28 further comprising:

means for encoding a second data to generate the first data; and

means for decoding the decoded first data to generate the second data.

31. The solid state non-volatile memory unit of claim 30 wherein the means for encoding the first data is adapted to execute a first ECC different from a second ECC executed by the means for encoding the second data.

32. The solid state non-volatile memory unit of claim 31 wherein the second ECC is a Reed-Solomon code.

33. The solid state non-volatile memory unit of claim 28 further comprising:

means for modulating the encoded data;

means for storing the modulated data in the memory array;

means for retrieving the modulated data from the memory array; and

means for demodulating the data retrieved from the memory array.

34. The solid state non-volatile memory unit of claim 28 wherein the means for encoding the first data encodes in accordance with a binary code.

35. The solid state non-volatile memory unit of claim 34 wherein the means for encoding the first data encodes in accordance with a binary code selected from a group consisting of Hamming code, BCH code, Reed-Muller code and Array Code.

36. The solid state non-volatile memory unit of claim 28 wherein said means for encoding the first data is selected from a group consisting of non-binary encoder and convolutional encoder.

37. The solid state non-volatile memory unit of claim 33 wherein said means for encoding the first data is disposed in a trellis-coded modulator.

Ex.1002 / Page 267 of 744
Sandisk Technologies, Inc. et al

38. The solid state non-volatile memory unit of claim 28 wherein said means for encoding the first data is an iterative encoder.

39. The solid state non-volatile memory unit of claim 38 wherein said iterative encoder encodes in accordance with a code selected from a group consisting of low-density parity check code and Turbo code.

40. The solid state non-volatile memory unit of claim 38 wherein said iterative encoder is a symbol-based iterative encoder.

Ex.1002 / Page 268 of 744
Sandisk Technologies, Inc. et al

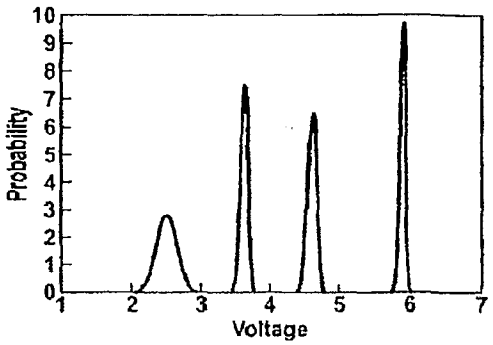


FIG. 1

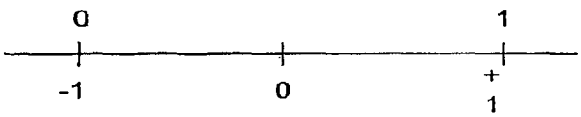


FIG. 2A

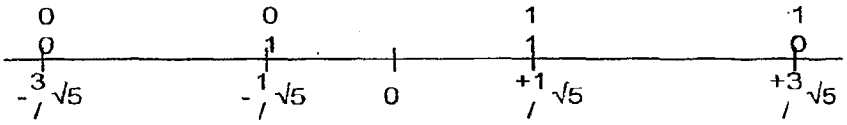


FIG. 2B

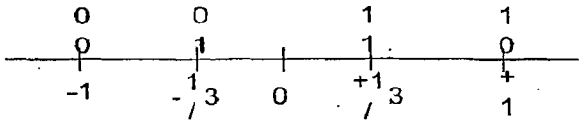


FIG. 2C

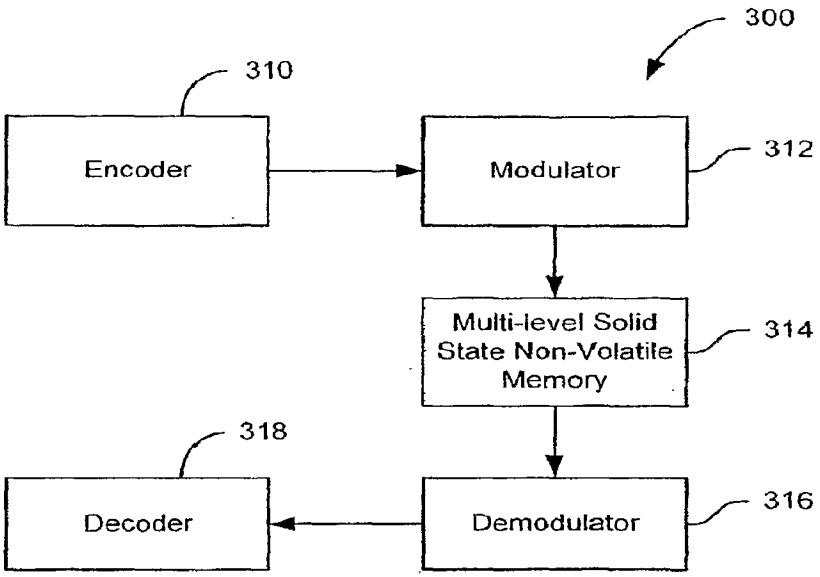

300
310
Encoder
Modulator
312
Multi-level Solid State Non-Volatile Memory
314
318
Decoder
Demodulator
316

FIG. 3A

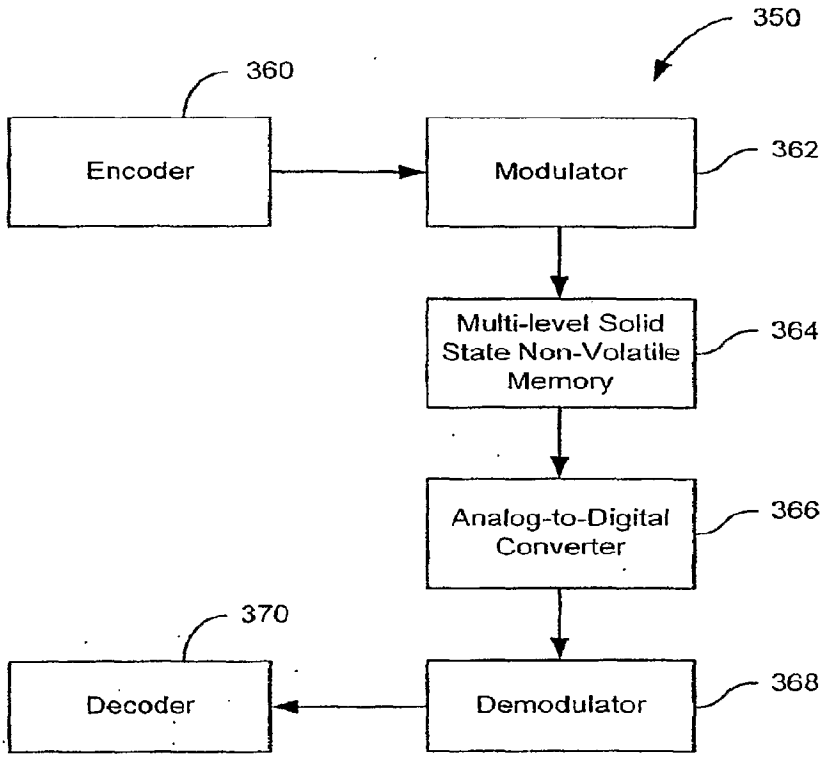

350
360
Encoder
Modulator
362
Multi-level Solid State Non-Volatile Memory
364
Analog-to-Digital Converter
366
370
Decoder
Demodulator
368

FIG. 3B

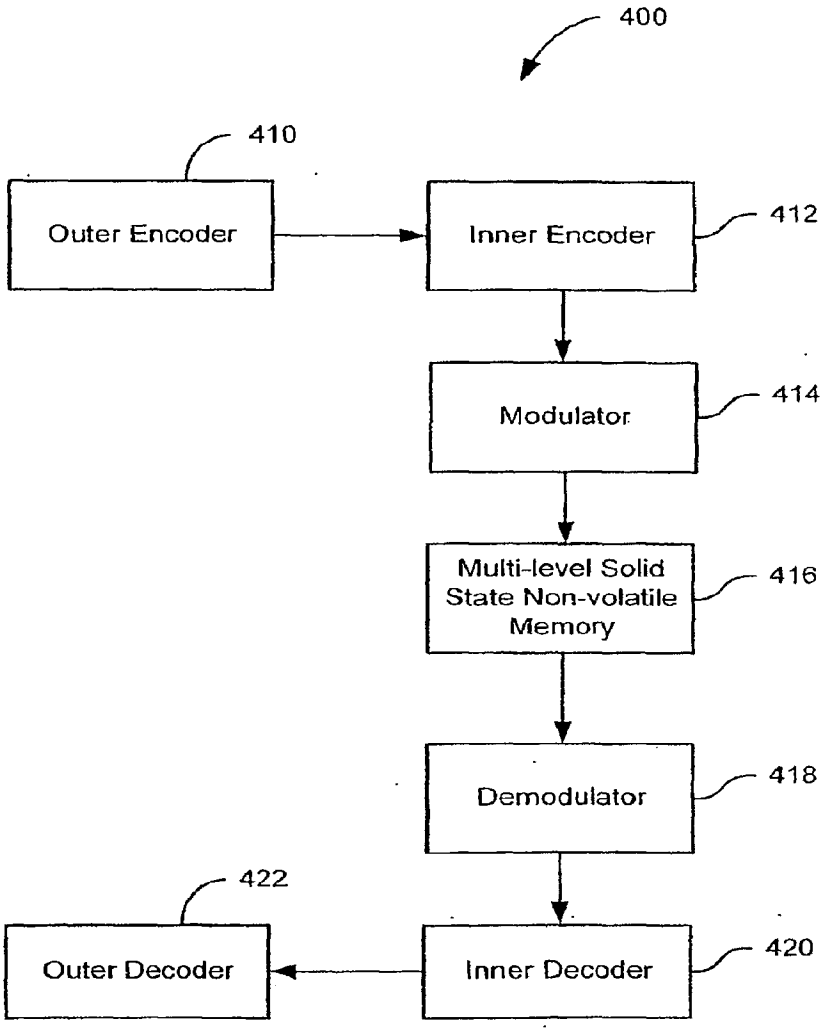

400
410
Outer Encoder
Inner Encoder
412
Modulator
414
Multi-level Solid State Non-volatile Memory
416
Demodulator
418
422
Outer Decoder
Inner Decoder
420

FIG. 4

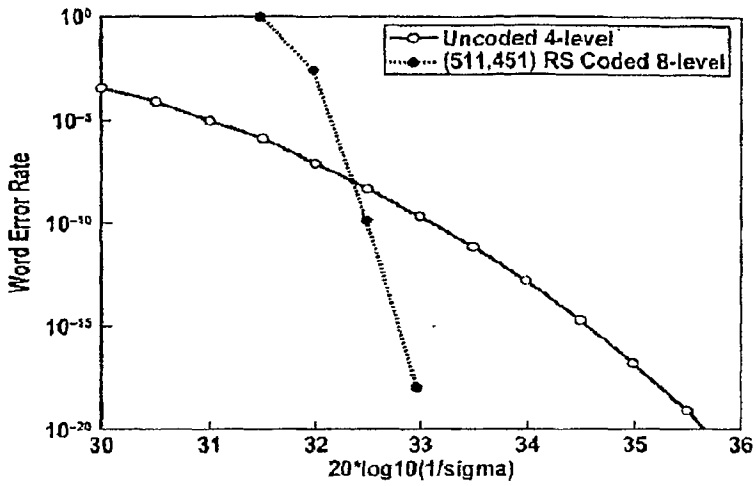

Uncoded 4-level
(511,451) RS Coded 8-level
Word Error Rate
10^0
10^-5
10^-10
10^-15
10^-20
30
31
32
33
34
35
36
20*log10(1/sigma)

FIG. 5

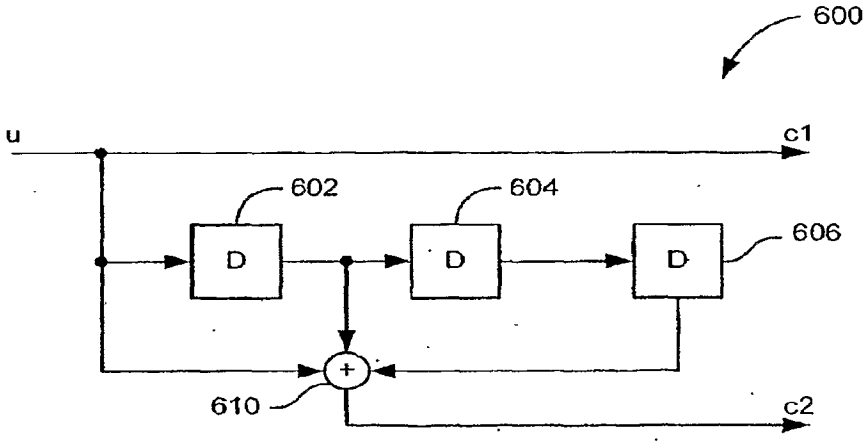

600
u
c1
602
604
606
D
D
D
610
c2

FIG. 6

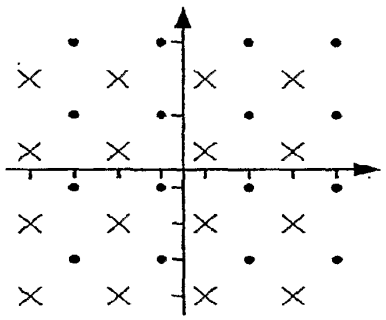


FIG. 7

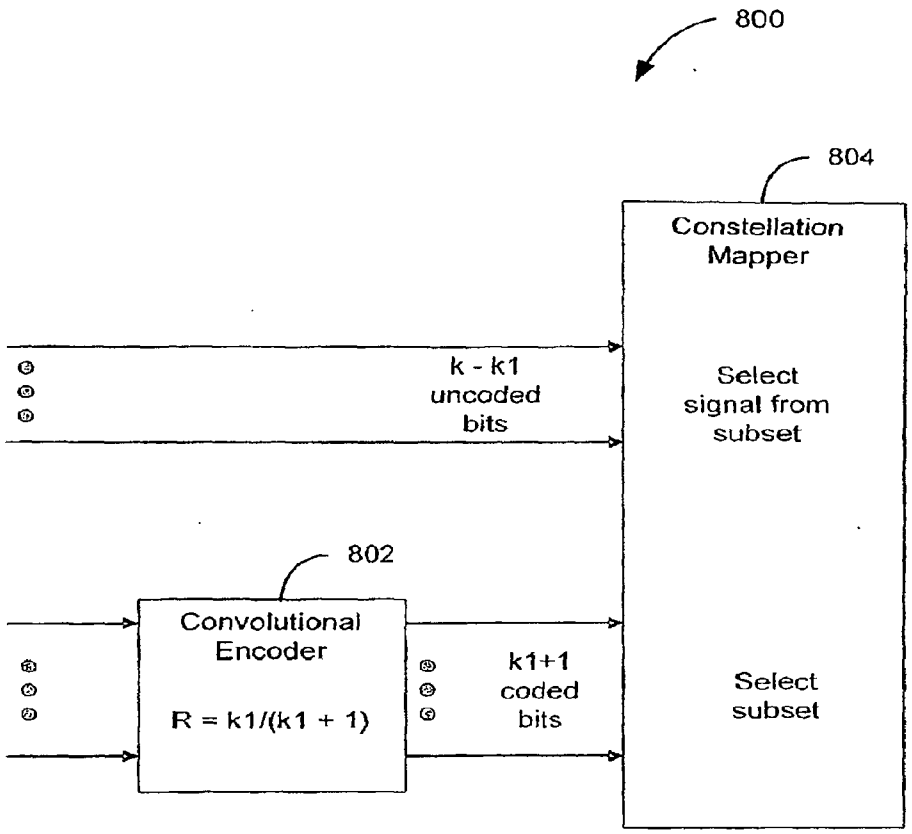

800
804
Constellation Mapper
k - k1 uncoded bits
Select signal from subset
802
Convolutional Encoder
R = k1/(k1 + 1)
k1+1 coded bits
Select subset

FIG. 8

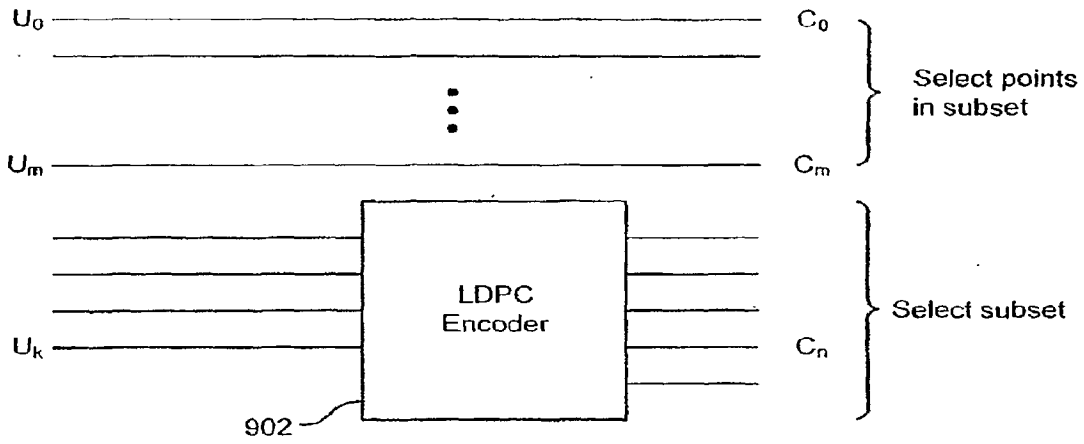

U_0
U_m
U_k
LDPC
Encoder
902
C_0
C_m
C_n
Select points
in subset
Select subset

FIG. 9

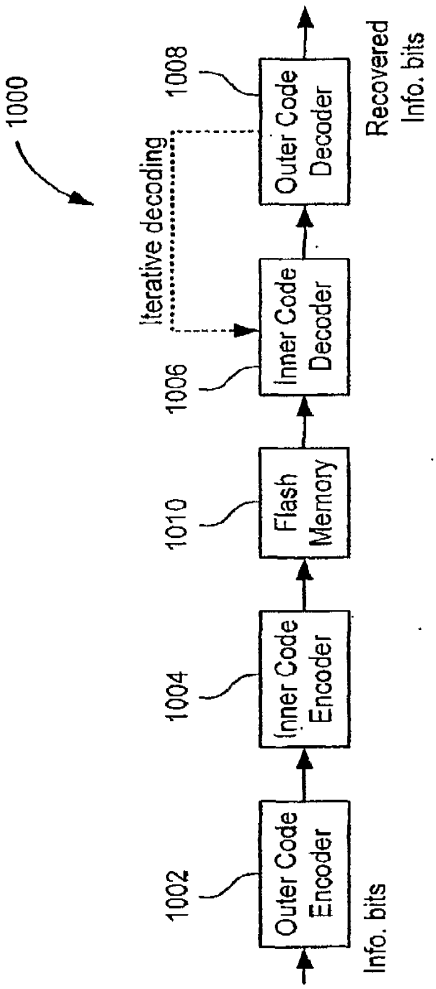

1000
1002
Outer Code Encoder
Info. bits
1004
Inner Code Encoder
1010
Flash Memory
1006
Inner Code Decoder
Iterative decoding
1008
Outer Code Decoder
Recovered Info. bits

FIG. 10A

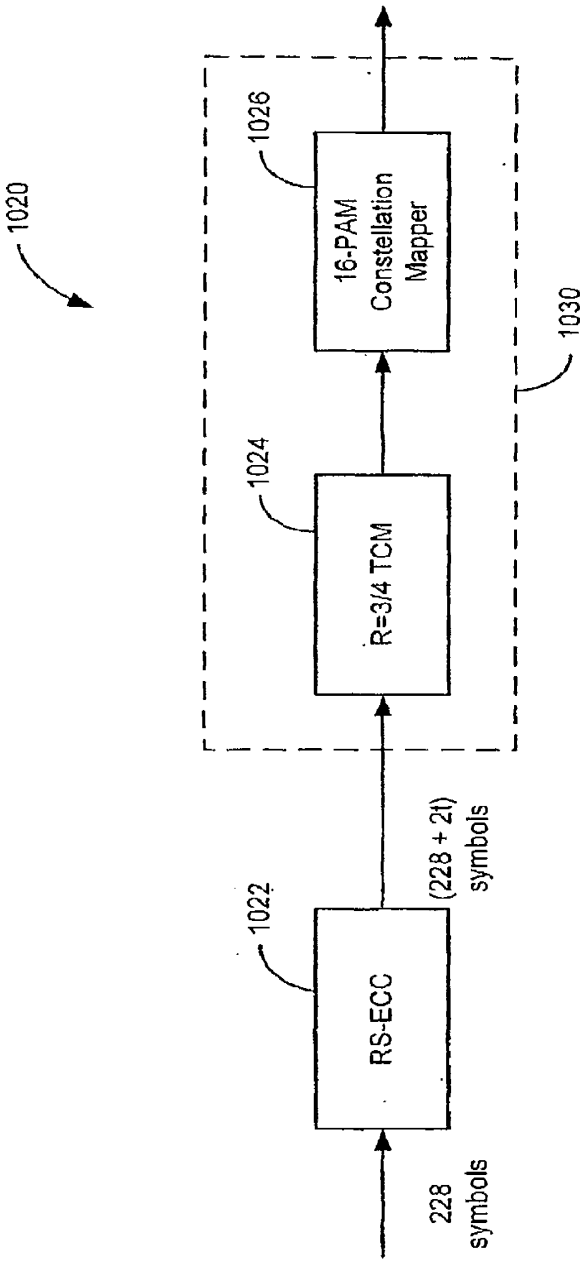

1020
1022
RS-ECC
228 symbols
(228 + 2t) symbols
1024
R=3/4 TCM
1026
16-PAM Constellation Mapper
1030

FIG. 10B

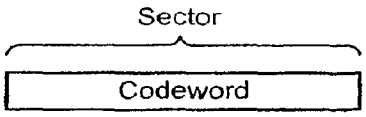


FIG. 11A

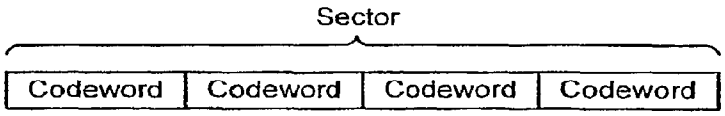


FIG. 11B

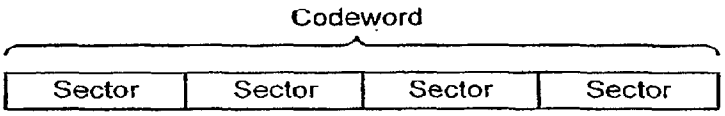


FIG. 11C

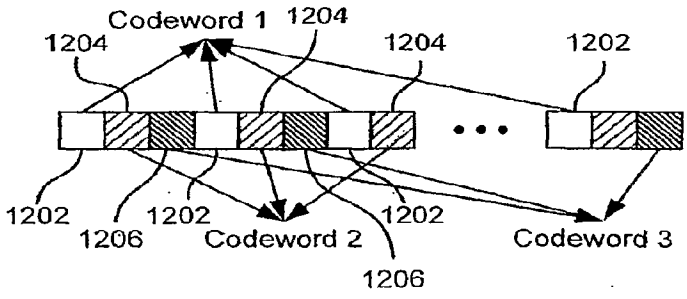


FIG. 12

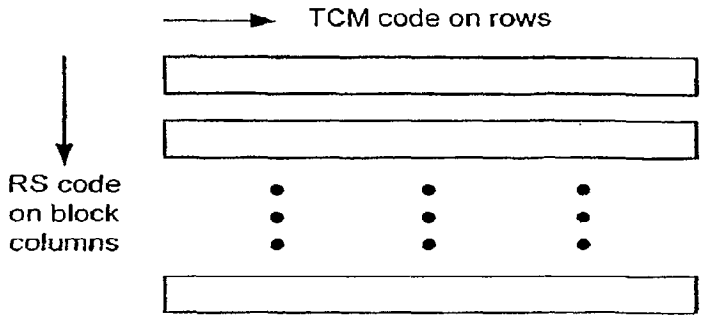


FIG. 13A

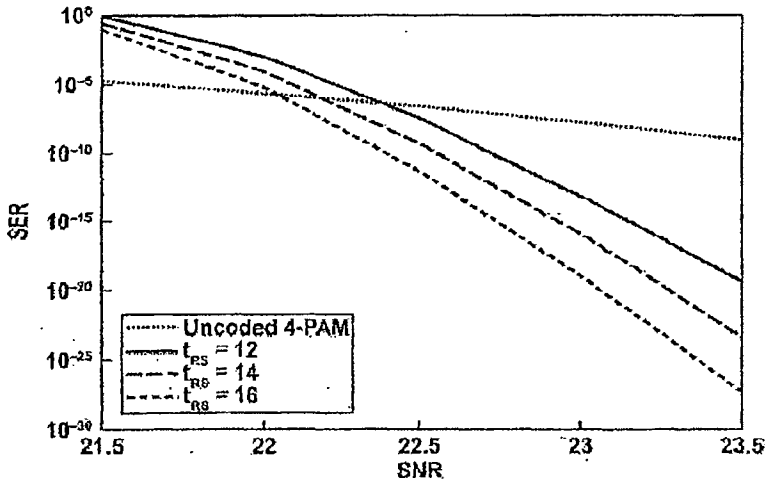


FIG. 13B

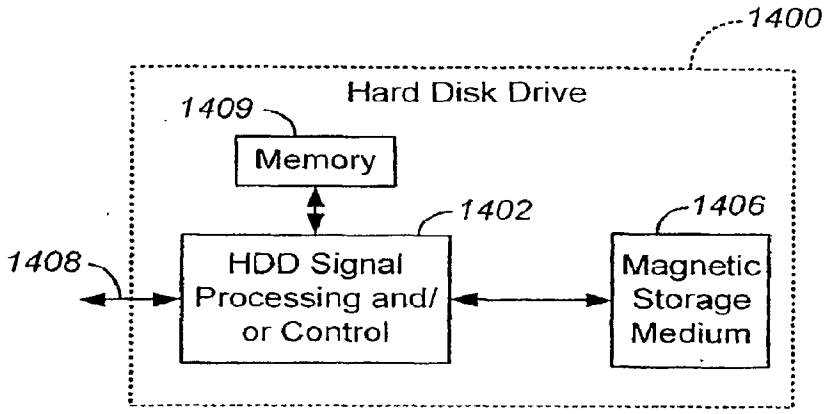


FIG. 14A

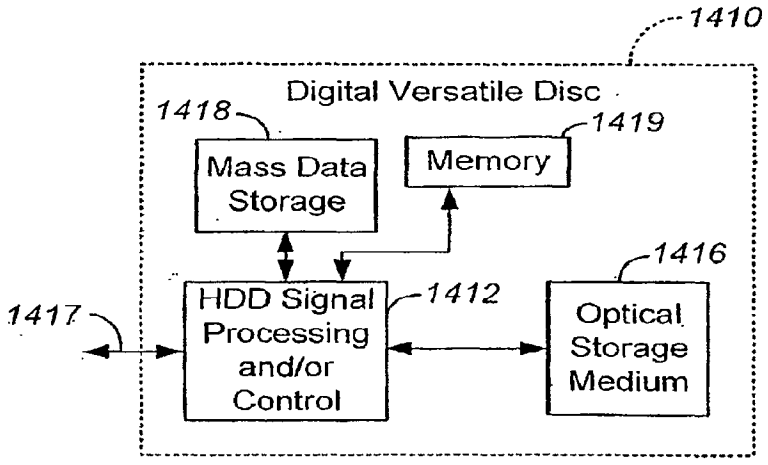


FIG. 14B

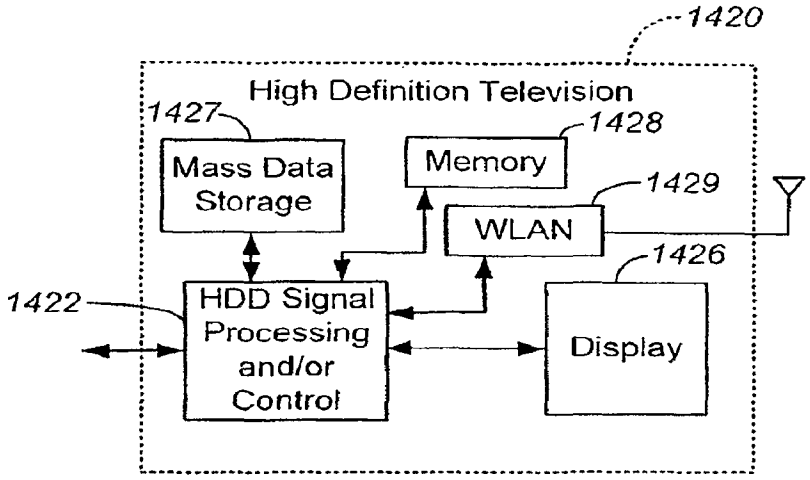


FIG. 14C

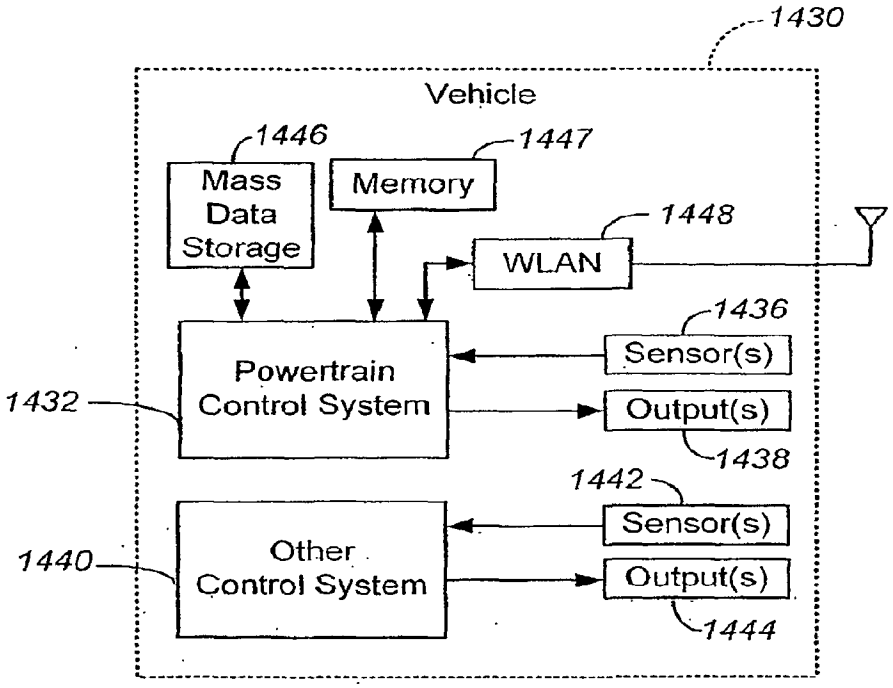


FIG. 14D

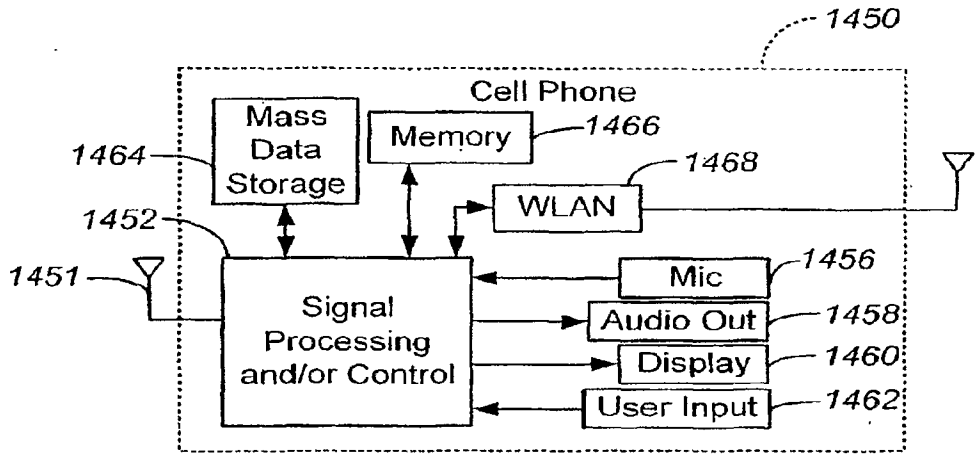


FIG. 14E

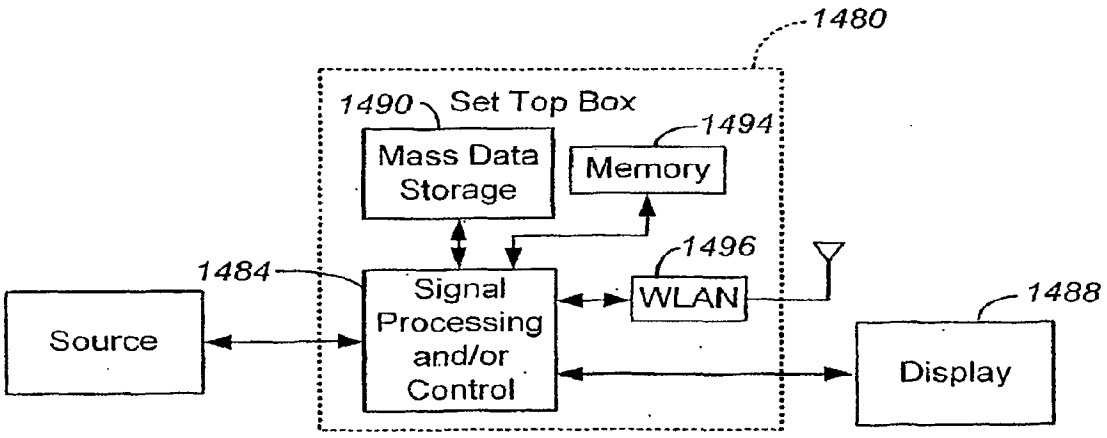


FIG. 14F

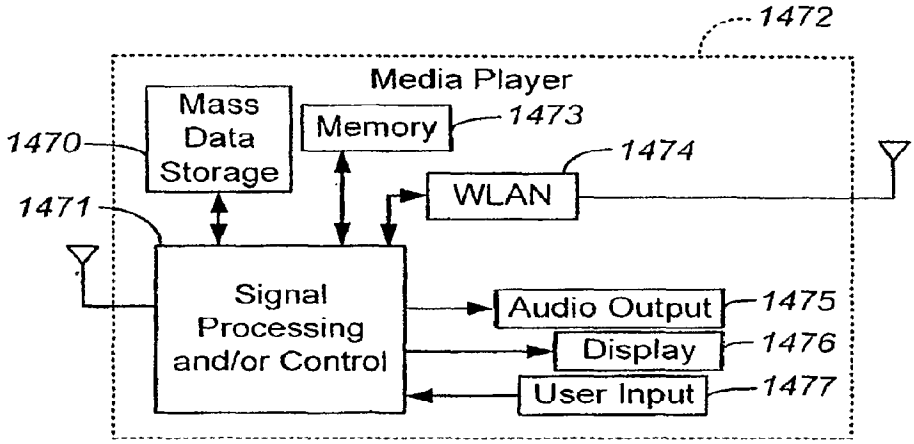


FIG. 14G

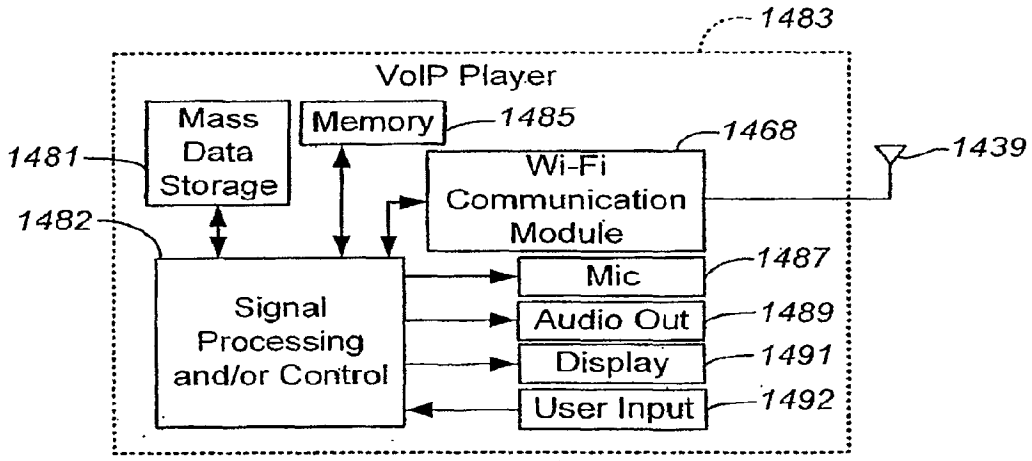


FIG. 14H

INFORMATION DISCLOSURE STATEMENT

Applicant	:	Northcott
App. No	:	Unknown
Filed	:	On Even Date Herewith
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
Examiner	:	Unknown
Art Unit	:	Unknown
Conf No.	:	Unknown

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Submitted herewith in the above-identified application is an Information Disclosure Statement listing references for consideration. Copies of any listed foreign and non-patent literature references are being submitted.

Timing of Disclosure

This Information Disclosure Statement is being filed within three months of the filing date and no fee is required.

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: May 22, 2012

By: Michael Okamoto

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

13056046_1
052212

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1

CONFIRMATION NO. 2672

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

FORMALITIES LETTER



Date Mailed: 06/06/2012

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  *Applicant must submit $**380** to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing.
  A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
  Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Additional claim fees of $**600** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.
- A surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of $**130** for a non-small entity, must be submitted.

SUMMARY OF FEES DUE:

Total fee(s) required within **TWO MONTHS** from the date of this Notice is $**1980** for a non-small entity
- $**380** Statutory basic filing fee.
- $**130** Surcharge.
- The application search fee has not been paid. Applicant must submit $**620** to complete the search fee.

- The application examination fee has not been paid. Applicant must submit $**250** to complete the examination fee for a non-small entity.
- Total additional claim fee(s) for this application is $**600**
  - $**600** for **10** total claims over 20.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web.
https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at **1-866-217-9197** or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

/skiflemariam/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/477,600	05/22/2012	2827	0.00	DATUMTE.076A1	30	3

CONFIRMATION NO. 2672

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

FILING RECEIPT

OC000000054609605

Date Mailed: 06/06/2012

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Philip L. Northcott, Coquitlam, CANADA;

Assignment For Published Patent Application

PMC-Sierra, Inc., Santa Clara, CA

Power of Attorney: None

Domestic Priority data as claimed by applicant

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.)

If Required, Foreign Filing License Granted: 06/01/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/477,600**

Projected Publication Date: Request for Non-Publication Acknowledged

Non-Publication Request: Yes

Early Publication Request: No

Title

SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE

Preliminary Class

365

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where

the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 13/477,600

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	380
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	620
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	250
TOTAL CLAIMS (37 CFR 1.16(i))	30 minus 20 =	* 10			OR	x 60 =	600
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*				x 250 =	0.00
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0.00
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	1850

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

Please Direct All Correspondence to Customer Number 20995

RESPONSE TO FORMALITIES NOTICE

Applicant	:	Philip L. Northcott
App. No.	:	13/477600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
Art Unit	:	2827
Conf No.	:	2672

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The above-captioned application was filed without a Declaration and/or filing fees. Enclosed in compliance with 37 CFR 1.53(f) are the following.

(X) A Declaration in 1 page.

(X) General Power of Attorney and Statement Under 37 CFR 3.73(b).

(X) A Notice to File Missing Parts dated June 6, 2012.

(X) Fees will be paid via EFS Web.

The Commissioner is hereby authorized to charge any additional fees which may be required, now or in the future, or credit any overpayment, to Account No. 11-1410.



Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

MPARTS
13502088 / 062212

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1

CONFIRMATION NO. 2672

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

FORMALITIES LETTER

OC000000054609607

Date Mailed: 06/06/2012

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  *Applicant must submit $**380** to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing.
  A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
  Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Additional claim fees of $**600** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.
- A surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of $**130** for a non-small entity, must be submitted.

SUMMARY OF FEES DUE:

Total fee(s) required within **TWO MONTHS** from the date of this Notice is $**1980** for a non-small entity
- $**380** Statutory basic filing fee.
- $**130** Surcharge.
- The application search fee has not been paid. Applicant must submit $**620** to complete the search fee.

- The application examination fee has not been paid. Applicant must submit $**250** to complete the examination fee for a non-small entity.
- Total additional claim fee(s) for this application is $**600**
  - $**600** for **10** total claims over 20.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web.
https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at **1-866-217-9197** or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

/skiflemariam/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

DECLARATION FOR UTILITY OR DESIGN APPLICATION UNDER 37 CFR 1.63

Docket No.: DATUMTE.076A1

Title: SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE

Inventor: Philip L. Northcott

Please Direct All Correspondence to Customer Number 20995

This Declaration is directed to the invention described in the application that:

Was filed as Application No. 13/477,600 on May 22, 2012.

As a below named inventor:

I believe the inventor named below to be the original and first inventor of the subject matter which is described and claimed and for which a patent is sought;

I have reviewed and understand the contents of the above-identified application, including the claims, and any amendment filed herewith or identified above;

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56;

The application was originally filed with an Application Data Sheet (ADS). The ADS sets forth any applicable Foreign Priority Claims under 35 USC § 119, and sets forth the full mailing address and residence of each inventor whose signature appears below as allowed under 37 CFR 1.63(e). The ADS also sets forth any Domestic Priority Claims under 35 USC §§ 119(e), 120, 121, and 365.

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

Full name of sole inventor: Philip L. Northcott

Signature: _____ Date: June 12, 2012

Citizenship: Canadian

Send Correspondence To:
KNOBBE, MARTENS, OLSON & BEAR, LLP
Customer No. 20995

13380486_2 / 060512

Docket No.: DATUMTE.076A1 **Customer No. 20995**

STATEMENT UNDER 37 CFR § 3.73(b)
ESTABLISHMENT OF ASSIGNEE

Applicant	:	Philip L. Northcott
App. No.	:	13/477600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
Examiner	:	Unknown
Group Art Unit	:	2827

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This document is being filed with a copy of a Power of Attorney signed by the Assignee. This Statement sets forth the chain of title of the above-identified application.

PMC-Sierra, Inc., a Delaware Corporation, is the Assignee of the entire right, title, and interest of the above-referenced application by virtue of:

The attached copy of the Assignment being forwarded to the Recordation Branch concurrently under separate cover.

The undersigned is an agent of Customer Number 20995 and is authorized to act on behalf of the Assignee. Please recognize or change the correspondence address for the above-identified application to **Customer No. 20995.**

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: June 22, 2012

By: Michael Okamoto

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

STMNT373

13502096 / 062212

REVOCATION AND GENERAL POWER OF ATTORNEY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The undersigned is an empowered representative of the Assignee and hereby appoints the registrants of Knobbe, Martens, Olson & Bear, LLP, **Customer No. 20,995**, as attorneys and agents to represent the Assignee before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned to the Assignee according to the USPTO assignment records or assignment documents supplied with an accompanying Statement Under 37 CFR § 3.73(b). This appointment is to be to the exclusion of the inventor(s) and his attorney(s) in accordance with the provisions of 37 CFR § 3.71.

Submission of this paper in connection with any matter of the below named assignee, together with a statement under 37 CFR 3.73(b), shall serve to revoke any previous powers of attorney in that matter.

A Statement Under 37 CFR § 3.73(b), signed by a registrant of Knobbe, Martens, Olson & Bear, LLP, is attached setting forth a full chain of title for the subject application owned by the Assignee named below.

Please recognize or change the correspondence address for the application identified in the attached Statement to **Customer No. 20,995.**

By: _____ Date: 2/17/2006

Name: Robert Yung Title: Vice President and Chief Technology Officer

Assignee: PMC-Sierra Inc.
Address: Mission Tower One, 3975 Freedom Circle, Santa Clara, California 95054

2348972
020206

FEB.17'2006 10:53 4084929462 PMC MISSION TOWERS 4TH FLOOR SO #7080 P.002/002

Ex.1002 / Page 297 of 744
Sandisk Technologies, Inc. et al

COPY ONLY DO NOT RECORD

Application No.: listed below
Filing Date: May 22, 2012

Docket Number: listed below

ASSIGNMENT

WHEREAS, Philip L. Northcott, a Canadian citizen, residing at Coquitlam, Canada (individual hereinafter "ASSIGNOR") invented certain new and useful improvements, technology, inventions, developments, ideas or discoveries collectively hereinafter referred to as the "Work" for which the Patent Applications identified below have been filed with the United States Patent and Trademark Office (hereinafter the "Applications");

Our Reference	Appl. No.	Title
DATUMTE.076A1	13/477,600	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
DATUMTE.076A2	13/477,633	SYSTEMS AND METHODS FOR INITIALIZING REGIONS OF A FLASH DRIVE HAVING DIVERSE ERROR CORRECTION CODING (ECC) SCHEMES
DATUMTE.076A3	13/477,629	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING AMONG DIFFERENT ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
DATUMTE.076A4	13/477,598	SYSTEMS AND METHODS FOR REDUNDANTLY STORING ERROR CORRECTION CODES IN A FLASH DRIVE
DATUMTE.076A5	13/477,595	SYSTEMS AND METHODS FOR LOW LATENCY, HIGH RELIABILITY ERROR CORRECTION IN A FLASH DRIVE
DATUMTE.076A6	13/477,568	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE
DATUMTE.076A7	13/477,599	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE
DATUMTE.076A8	13/477,845	SYSTEMS AND METHODS FOR RECLAIMING FLASH BLOCKS OF A FLASH DRIVE
DATUMTE.076A9	13/477,635	SYSTEMS AND METHODS FOR REDUNDANTLY STORING DATA IN A FLASH DRIVE SUCH THAT A FAILED PAGE CAN BE TOLERATED
DATUMTE.076A10	13/477,601	SYSTEMS AND METHODS FOR RECOVERING DATA FROM FAILED PORTIONS OF A FLASH DRIVE

AND WHEREAS, PMC-Sierra, Inc., a Delaware Corporation, with its principal place of business at 1380 Bordeaux Drive, Sunnyvale, California (hereinafter the "ASSIGNEE"), desires to acquire the entire right, title, and interest in and to the Applications and the Work:

NOW, THEREFORE, for good and valuable consideration, the receipt of which is hereby acknowledged, ASSIGNOR hereby acknowledges that ASSIGNOR has sold, assigned, transferred and set over, and by these presents does hereby sell, assign, transfer and set over, unto said ASSIGNEE, its successors, legal representatives and assigns, the entire right, title, and interest throughout the world in the Applications and the Work, including all provisional applications relating thereto (if any), and all nonprovisional applications claiming priority thereto, including, all divisions, continuations, continuations-in-part, reissues, and reexaminations thereof, and all Letters Patent of the United States which may be granted thereon and all reissues and extensions thereof, and all rights of priority under International Conventions and any related Letters Patent which may hereafter be granted or filed in any country or countries foreign to the United States, all extensions, renewals and reissues thereof; and ASSIGNOR hereby authorizes and requests the Commissioner of Patents of the United States, and any Official of any country or countries foreign to the United States, whose duty it is to issue patents on applications as aforesaid, to issue all related Letters Patent to the ASSIGNEE, its successors, legal representatives and assigns, in accordance with the terms of this instrument.

AND ASSIGNOR DOES HEREBY sell, assign, transfer, and convey to said ASSIGNEE, its successors, legal representatives, and assigns all claims for damages and all remedies arising out of any violation of the rights assigned hereby that may have accrued prior to the date of assignment to said ASSIGNEE, or may accrue hereafter, including, but not limited to, the right to sue for, collect, and retain damages for past infringements of said Letters Patent before or after issuance.

AND ASSIGNOR DOES HEREBY covenant and agree that ASSIGNOR will communicate to said ASSIGNEE, its successors, legal representatives and assigns, any facts known to ASSIGNOR respecting the Work, and testify in any legal proceeding, assist in the preparation of any other provisional or non-provisional applications relating to the Applications and the Work or any improvements made thereto, sign all lawful papers, execute and make all rightful oaths and/or declarations in connection with the Applications and the Work including any improvements made thereto, any patent applications filed therefrom, and any continuing application filed from any of the aforementioned applications, and generally do everything possible to aid the ASSIGNEE, its successors, legal representatives and assigns, to obtain and enforce proper patent protection for the Work in all countries.

This 12 day of June, 2012.

Philip L. Northcott

Witnessed by:

13380769_2
060512

Electronic Patent Application Fee Transmittal

Application Number:	13477600
Filing Date:	22-May-2012
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Filer:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Utility application filing	1011	1	380	380
Utility Search Fee	1111	1	620	620
Utility Examination Fee	1311	1	250	250
Pages:				
Claims:				
Claims in excess of 20	1202	10	60	600
Miscellaneous-Filing:				
Late filing fee for oath or declaration	1051	1	130	130

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**1980**

Electronic Acknowledgement Receipt

EFS ID:	13086772
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/ThuyQuyen Nguyen
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	22-JUN-2012
Filing Date:	22-MAY-2012
Time Stamp:	17:11:38
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$1980
RAM confirmation Number	4441
Deposit Account	111410
Authorized User	KNOBBE MARTENS OLSON AND BEAR

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Applicant Response to Pre-Exam Formalities Notice	Response_to_MParts_Notice_DATUMTE076A1.pdf	30818 9bef11ebbb442d178c4ca1775f71de71dc4f00db	no	1
Warnings:					
Information:					
2	Applicant Response to Pre-Exam Formalities Notice	Copy_of_MParts_Notice_DATUMTE076A1.PDF	76443 4ad79fd3ce3b193a0378ff19b090e1c90e3cb10d	no	2
Warnings:					
Information:					
3	Oath or Declaration filed	Declaration_DATUMTE076A1.pdf	133514 70be93e5df2e7ca7013cbe7af633f4d2cabb312f	no	1
Warnings:					
Information:					
4	Assignee showing of ownership per 37 CFR 3.73(b).	Stmnt373b_DATUMTE076A1.pdf	32966 8681db667889db4c15be0e6ef0956c92a5445af3	no	1
Warnings:					
Information:					
5	Power of Attorney	POA_DATUMTE076A1.pdf	393297 6631dab15812dac8041b0841f21ce638a57e3f0e	no	3
Warnings:					
Information:					
6	Fee Worksheet (SB06)	fee-info.pdf	38290 d3ab115859a6716af835d3f0eb1c7cbe47ff64ca	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	705328		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/477,600	05/22/2012	2827	1980	DATUMTE.076A1	30	3

CONFIRMATION NO. 2672

UPDATED FILING RECEIPT

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

OC000000055071288

Date Mailed: 07/06/2012

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Philip L. Northcott, Coquitlam, CANADA;

Assignment For Published Patent Application

PMC-Sierra, Inc., Santa Clara, CA

Power of Attorney: The patent practitioners associated with Customer Number 20995

Domestic Priority data as claimed by applicant

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.)

If Required, Foreign Filing License Granted: 06/01/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/477,600**

Projected Publication Date: Request for Non-Publication Acknowledged

Non-Publication Request: Yes

Early Publication Request: No

Title

SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE

Preliminary Class

365

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where

the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 13/477,600

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	380
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	620
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	250
TOTAL CLAIMS (37 CFR 1.16(i))	30 minus 20 =	* 10			OR	x 60 =	600
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*				x 250 =	0.00
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0.00
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	1850

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1

CONFIRMATION NO. 2672

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

POA ACCEPTANCE LETTER

OC000000055071362

Date Mailed: 07/06/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/22/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/ybedada/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477,600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	DECADY, ALBERT
SHEET 1 OF 1	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	2012/0089767	04-12-2012	Lee, et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]
	2	WO 2012047500 A1	04-12-2012	Colgrove, et al.		
	3	WO 2012075200 A2	06-07-2012	Werner, et al.		

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]

14698633
012113

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 April 2012 (12.04.2012)

PCT

(10) International Publication Number
WO 2012/047500 A1

(51) International Patent Classification:
G06F 11/10 (2006.01) *G06F 3/06* (2006.01)

(21) International Application Number:
PCT/US2011/052268

(22) International Filing Date:
20 September 2011 (20.09.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/892,535 28 September 2010 (28.09.2010) US

(71) Applicant *(for all designated States except US)*: **PURE STORAGE, INC.** [US/US]; 650 Castro Street, Suite 220, Mountain View, CA 94041 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **COLGROVE, John** [US/US]; 722 Vista Grande Ave., Los Altos, CA 94024 (US). **HAYES, John** [CA/US]; 800 High School Way, #330, Mountain View, CA 94041 (US). **HONG, Bo** [CN/US]; 1555 W. Middlefield Rd., Apt. 95, Mountain View, CA 94043 (US). **MILLER, Ethan** [US/US]; 203 Kalkar Drive, Santa Cruz, CA 95060 (US).

(74) Agent: **MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.**; RANKIN, Rory D., P.O. Box 398, Austin, TX 78767-0398 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: INTRA-DEVICE DATA PROTECTION IN A RAID ARRAY

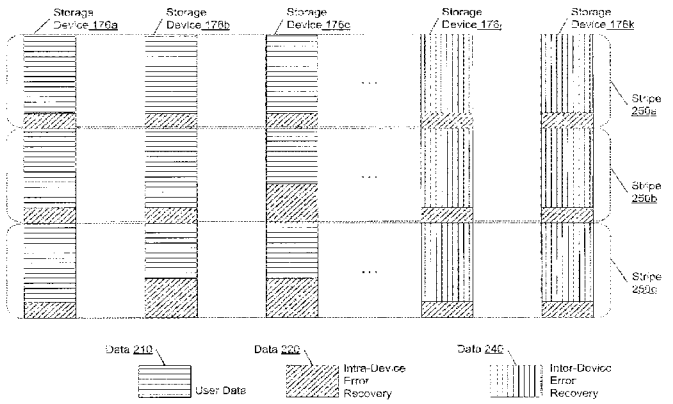


FIG. 7

(57) Abstract: A system and method for intra-device data protection in a RAID array. A computer system comprises client computers and data storage arrays coupled to one another via a network. A data storage array utilizes solid-state drives and Flash memory cells for data storage. A storage controller within a data storage array is configured to identify a unit of data stored in the data storage subsystem, wherein said unit of data is stored across at least a first storage device and a second storage device of the plurality of storage devices, each of the first storage device and the second storage device storing intra-device redundancy data corresponding to the unit of data; and change an amount of intra-device redundancy data corresponding to the unit of data on only the first storage device.

WO 2012/047500 A1

TITLE: INTRA-DEVICE DATA PROTECTION IN A RAID ARRAY

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This invention relates to computer networks and, more particularly, to efficiently distributing data among a plurality of solid-state storage devices.

Description of the Related Art

[0002] As computer memory storage and data bandwidth increase, so does the amount and complexity of data that businesses daily manage. Large-scale distributed storage systems, such as data centers, typically run many business operations. A distributed storage system may be coupled to client computers interconnected by one or more networks. If any portion of the distributed storage system has poor performance or becomes unavailable, company operations may be impaired or stopped completely. A distributed storage system therefore is expected to maintain high standards for data availability and high-performance functionality. As used herein, storage disks may be referred to as storage devices as some types of storage technologies do not include disks.

[0003] To protect against data loss, storage devices often include error detection and correction mechanisms. Often these mechanisms take the form of error correcting codes which are generated by the devices and stored within the devices themselves. In addition, distributed storage systems may also utilize decentralized algorithms to distribute data among a collection of storage devices. These algorithms generally map data objects to storage devices without relying on a central directory. Examples of such algorithms include Replication Under Scalable Hashing (RUSH), and Controlled Replication Under Scalable Hashing (CRUSH). With no central directory, multiple clients in a distributed storage system may simultaneously access data objects on multiple servers. In addition, the amount of stored metadata may be reduced. However, the difficult task remains of distributing data among multiple storage disks with varying capacities, input/output (I/O) characteristics and reliability issues. Similar to the storage devices themselves, these algorithms may also include error detection and correction algorithms such as RAID type algorithms (e.g., RAID5 and RAID6) or Reed-Solomon codes.

[0004] The technology and mechanisms associated with chosen storage devices determine the methods used to distribute data among multiple storage devices, which may be dynamically

added and removed. For example, the algorithms described above were developed for systems utilizing hard disk drives (HDDs). The HDDs comprise one or more rotating disks, each coated with a magnetic medium. These disks rotate at a rate of several thousand rotations per minute for several hours daily. In addition, a magnetic actuator is responsible for positioning magnetic read/write devices over the rotating disks. These actuators are subject to friction, wear, vibrations and mechanical misalignments, which result in reliability issues. The above-described data distribution algorithms are based upon the characteristics and behaviors of HDDs.

[0005] One example of another type of storage disk is a Solid-State Disk (SSD). A Solid-State Disk may also be referred to as a Solid-State Drive. An SSD may emulate a HDD interface, but an SSD utilizes solid-state memory to store persistent data rather than electromechanical devices as found in a HDD. For example, an SSD may comprise banks of Flash memory. Without moving parts or mechanical delays, an SSD may have a lower access time and latency than a HDD. However, SSD typically have significant write latencies. In addition to different input/output (I/O) characteristics, an SSD experiences different failure modes than a HDD. Accordingly, high performance and high reliability may not be achieved in systems comprising SSDs for storage while utilizing distributed data placement algorithms developed for HDDs.

[0006] In view of the above, systems and methods for efficiently distributing data and detecting and correcting errors among a plurality of solid-state storage devices are desired.

SUMMARY OF THE INVENTION

[0007] Various embodiments of a computer system and methods for efficiently distributing and managing data among a plurality of solid-state storage devices are disclosed.

[0008] In one embodiment, a computer system comprises a plurality of client computers configured to convey read and write requests over a network to one or more data storage arrays coupled to receive the read and write requests via the network. Contemplated is a data storage array(s) comprising a plurality of storage locations on a plurality of storage devices. In various embodiments, the storage devices are configured in a redundant array of independent drives (RAID) arrangement for data storage and protection. The data storage devices may include solid-state memory technology for data storage, such as Flash memory cells. The data storage subsystem further comprises a storage controller configured to identify a unit of data stored in the data storage subsystem, wherein said unit of data is stored across at least a first storage device and a second storage device of the plurality of storage devices, each of the first storage device

and the second storage device storing intra-device redundancy data corresponding to the unit of data; and change an amount of intra-device redundancy data corresponding to the unit of data on only the first storage device. In various embodiments, the unit of data is a RAID stripe.

[0009] Also contemplated are embodiments wherein the storage controller is configured to change the amount of intra-device redundancy data stored on the first storage device in response to detecting an indication of a change in reliability of the first storage device. Additionally, the controller may be configured to store data indicative of characteristics of the plurality of storage devices. In such embodiments, the detected change in reliability may be based at least in part on these characteristics. Such characteristics may include one or more of an error rate, an access rate, an age, an occupancy level, and an age of stored data, or otherwise.

[0010] These and other embodiments will become apparent upon consideration of the following description and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is a generalized block diagram illustrating one embodiment of network architecture.

[0012] FIG. 2 is a generalized block diagram of one embodiment of a dynamic intra-device redundancy scheme.

[0013] FIG. 3 is a generalized flow diagram illustrating one embodiment of a method for adjusting intra-device protection in a data storage subsystem.

[0014] FIG. 4 is a generalized block diagram of one embodiment of a storage subsystem.

[0015] FIG. 5 is a generalized block diagram of one embodiment of a device unit.

[0016] FIG. 6 is a generalized block diagram illustrating one embodiment of a state table.

[0017] FIG. 7 is a generalized block diagram illustrating one embodiment of a flexible RAID data layout architecture.

[0018] FIG. 8 is a generalized block diagram illustrating another embodiment of a flexible RAID data layout architecture.

[0019] FIG. 9 is a generalized flow diagram illustrating one embodiment of a method for dynamically determining a layout in a data storage subsystem.

[0020] FIG. 10 is a generalized block diagram illustrating yet another embodiment of a flexible RAID data layout architecture.

[0021] FIG. 11A illustrates one embodiment of a device layout.

[0022] FIG. 11B illustrates one embodiment of a segment.

[0023] FIG. 11C is a generalized block diagram illustrating one embodiment of data storage arrangements within different page types.

[0024] FIG. 12 is a generalized block diagram illustrating one embodiment of a hybrid RAID data layout.

[0025] FIG. 13 is a generalized flow diagram illustrating one embodiment of a method for selecting alternate RAID geometries in a data storage subsystem.

[0026] While the invention is susceptible to various modifications and alternative forms, specific embodiments are shown by way of example in the drawings and are herein described in detail. It should be understood, however, that drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION

[0027] In the following description, numerous specific details are set forth to provide a thorough understanding of the present invention. However, one having ordinary skill in the art should recognize that the invention might be practiced without these specific details. In some instances, well-known circuits, structures, signals, computer program instruction, and techniques have not been shown in detail to avoid obscuring the present invention.

[0028] Referring to FIG. 1, a generalized block diagram of one embodiment of network architecture 100 is shown. As described further below, one embodiment of network architecture 100 includes client computer systems 110a-110b interconnected to one another through a network 180 and to data storage arrays 120a-120b. Network 180 may be coupled to a second network 190 through a switch 140. Client computer system 110c is coupled to client computer systems 110a-110b and data storage arrays 120a-120b via network 190. In addition, network 190 may be coupled to the Internet 160 or other outside network through switch 150.

[0029] It is noted that in alternative embodiments, the number and type of client computers and servers, switches, networks, data storage arrays, and data storage devices is not limited to those shown in FIG. 1. At various times one or more clients may operate offline. In addition, during operation, individual client computer connection types may change as users connect, disconnect, and reconnect to network architecture 100. A further description of each of the components shown in FIG. 1 is provided shortly. First, an overview of some of the features provided by the data storage arrays 120a-120b is described.

[0030] In the network architecture 100, each of the data storage arrays 120a-120b may be used for the sharing of data among different servers and computers, such as client computer systems 110a-110c. In addition, the data storage arrays 120a-120b may be used for disk mirroring, backup and restore, archival and retrieval of archived data, and data migration from one storage device to another. In an alternate embodiment, one or more client computer systems 110a-110c may be linked to one another through fast local area networks (LANs) in order to form a cluster. One or more nodes linked to one another form a cluster, which may share a storage resource, such as a cluster shared volume residing within one of data storage arrays 120a-120b.

[0031] Each of the data storage arrays 120a-120b includes a storage subsystem 170 for data storage. Storage subsystem 170 may comprise a plurality of storage devices 176a-176m. These storage devices 176a-176m may provide data storage services to client computer systems 110a-110c. Each of the storage devices 176a-176m may be configured to receive read and write requests and comprise a plurality of data storage locations, each data storage location being addressable as rows and columns in an array. In one embodiment, the data storage locations within the storage devices 176a-176m may be arranged into logical, redundant storage containers or RAID arrays (redundant arrays of inexpensive/independent disks). However, the storage devices 176a-176m may not comprise a disk. In one embodiment, each of the storage devices 176a-176m may utilize technology for data storage that is different from a conventional hard disk drive (HDD). For example, one or more of the storage devices 176a-176m may include or be further coupled to storage consisting of solid-state memory to store persistent data. In other embodiments, one or more of the storage devices 176a-176m may include or be further coupled to storage utilizing spin torque transfer technique, magnetoresistive random access memory (MRAM) technique, or other storage techniques. These different storage techniques may lead to differing reliability characteristics between storage devices.

[0032] The type of technology and mechanism used within each of the storage devices 176a-176m may determine the algorithms used for data object mapping and error detection and correction. The logic used in these algorithms may be included within one or more of a base operating system (OS) 116, a file system 140, one or more global RAID engines 178 within a storage subsystem controller 174, and control logic within each of the storage devices 176a-176m.

[0033] In one embodiment, the included solid-state memory comprises solid-state drive (SSD) technology. Typically, SSD technology utilizes Flash memory cells. As is well known in the art, a Flash memory cell holds a binary value based on a range of electrons trapped and stored in a

floating gate. A fully erased Flash memory cell stores no or a minimal number of electrons in the floating gate. A particular binary value, such as binary 1 for single-level cell (SLC) Flash, is associated with an erased Flash memory cell. A multi-level cell (MLC) Flash has a binary value 11 associated with an erased Flash memory cell. After applying a voltage higher than a given threshold voltage to a controlling gate within a Flash memory cell, the Flash memory cell traps a given range of electrons in the floating gate. Accordingly, another particular binary value, such as binary 0 for SLC Flash, is associated with the programmed (written) Flash memory cell. A MLC Flash cell may have one of multiple binary values associated with the programmed memory cell depending on the applied voltage to the control gate.

[0034] Generally speaking, SSD technologies provide lower read access latency times than HDD technologies. However, the write performance of SSDs is significantly impacted by the availability of free, programmable blocks within the SSD. As the write performance of SSDs is significantly slower compared to the read performance of SSDs, problems may occur with certain functions or operations expecting similar latencies. In addition, the differences in technology and mechanisms between HDD technology and SDD technology lead to differences in reliability characteristics of the data storage devices 176a-176m.

[0035] In various embodiments, a Flash cell within an SSD must generally be erased before it is written with new data. Additionally, an erase operation in various flash technologies must also be performed on a block-wise basis. Consequently, all of the Flash memory cells within a block (an erase segment or erase block) are erased together. A Flash erase block may comprise multiple pages. For example, a page may be 4 kilobytes (KB) in size and a block may include 64 pages, or 256KB. Compared to read operations in a Flash device, an erase operation may have a relatively high latency - which may in turn increase the latency of a corresponding write operation. Programming or reading of Flash technologies may be performed at a lower level of granularity than the erase block size. For example, Flash cells may be programmed or read at a byte, word, or other size.

[0036] A Flash cell experiences wear after repetitive erase-and-program operations. The wear in this case is due to electric charges that are injected and trapped in the dielectric oxide layer between the substrate and the floating gate of the MLC Flash cell. In one example, a MLC Flash cell may have a limit of a number of times it experiences an erase-and-program operation, such as a range from 10,000 to 100,000 cycles. In addition, SSDs may experience program disturb errors that cause a neighboring or nearby Flash cell to experience an accidental state change while another Flash cell is being erased or programmed. Further, SSDs include read disturb

errors, wherein the accidental state change of a nearby Flash cell occurs when another Flash cell is being read.

[0037] Knowing the characteristics of each of the one or more storage devices 176a-176m may lead to more efficient data object mapping and error detection and correction. In one embodiment, the global RAID engine 178 within the storage controller 174 may detect for the storage devices 176a-176m at least one or more of the following: inconsistent response times for I/O requests, incorrect data for corresponding accesses, error rates and access rates. In response to at least these characteristics, the global RAID engine 178 may determine which RAID data layout architecture to utilize for a corresponding group of storage devices within storage devices 176a-176m. In addition, the global RAID engine 178 may dynamically change both an intra-device redundancy scheme and an inter-device RAID data layout based on the characteristics of the storage devices 176a-176m.

[0038] FIG. 1 illustrates an example of a system capable of the described features according to one embodiment. Further details are provided below. Referring to FIG. 1, a further description of the components of network architecture 100 is provided below.

[0039] Components of a Network Architecture

[0040] Again, as shown, network architecture 100 includes client computer systems 110a-110c interconnected through networks 180 and 190 to one another and to data storage arrays 120a-120b. Networks 180 and 190 may include a variety of techniques including wireless connection, direct local area network (LAN) connections, storage area networks (SANs), wide area network (WAN) connections such as the Internet, a router, and others. Networks 180 and 190 may comprise one or more LANs that may also be wireless. Networks 180 and 190 may further include remote direct memory access (RDMA) hardware and/or software, transmission control protocol/internet protocol (TCP/IP) hardware and/or software, router, repeaters, switches, grids, and/or others. Protocols such as Ethernet, Fibre Channel, Fibre Channel over Ethernet (FCoE), iSCSI, and so forth may be used in networks 180 and 190. Switch 140 may utilize a protocol associated with both networks 180 and 190. The network 190 may interface with a set of communications protocols used for the Internet 160 such as the Transmission Control Protocol (TCP) and the Internet Protocol (IP), or TCP/IP. Switch 150 may be a TCP/IP switch.

[0041] Client computer systems 110a-110c are representative of any number of stationary or mobile computers such as desktop personal computers (PCs), workstations, laptops, handheld computers, servers, server farms, personal digital assistants (PDAs), smart phones, and so forth.

Generally speaking, client computer systems 110a-110c include one or more processors comprising one or more processor cores. Each processor core includes circuitry for executing instructions according to a predefined general-purpose instruction set. For example, the x86 instruction set architecture may be selected. Alternatively, the Alpha®, PowerPC®, SPARC®, or any other general-purpose instruction set architecture may be selected. The processor cores may access cache memory subsystems for data and computer program instructions. The cache subsystems may be coupled to a memory hierarchy comprising random access memory (RAM) and a storage device.

[0042] Each processor core and memory hierarchy within a client computer system may be in turn connected to a network interface. In addition to hardware components, each of the client computer systems 110a-110c may include a base operating system (OS) stored within the memory hierarchy. The base OS may be representative of any of a variety of specific operating systems, such as, for example, MS-DOS®, MS-WINDOWS®, OS/2®, UNIX®, Linux®, Solaris® or another known operating system. As such, the base OS may be operable to provide various services to the end-user and provide a software framework operable to support the execution of various programs. Additionally, each of the client computer systems 110a-110c may include a hypervisor used to support higher-level virtual machines (VMs). As is well known to those skilled in the art, virtualization may be used in desktops and servers to fully or partially decouple software, such as an OS, from a system's hardware. Virtualization may provide an end-user with an illusion of multiple OSes running on a same machine each having its own resources, such logical storage entities (e.g., logical unit numbers, LUNs) corresponding to the storage devices 176a-176m within each of the data storage arrays 120a-120b.

[0043] Each of the data storage arrays 120a-120b may be used for the sharing of data among different servers, such as the client computer systems 110a-110c. Each of the data storage arrays 120a-120b includes a storage subsystem 170 for data storage. Storage subsystem 170 may comprise a plurality of storage devices 176a-176m. Each of these storage devices 176a-176m may be a SSD. A controller 174 may comprise logic for handling received read/write requests. For example, the algorithms briefly described above may be executed in at least controller 174. A random-access memory (RAM) 172 may be used to batch operations, such as received write requests.

[0044] The base OS 132, the file system 134, any OS drivers (not shown) and other software stored in memory medium 130 may provide functionality enabling access to files and LUNs, and the management of these functionalities. The base OS 134 and the OS drivers may comprise

program instructions stored on the memory medium 130 and executable by processor 122 to perform one or more memory access operations in storage subsystem 170 that correspond to received requests.

[0045] Each of the data storage arrays 120a-120b may use a network interface 124 to connect to network 180. Similar to client computer systems 110a-110c, in one embodiment, the functionality of network interface 124 may be included on a network adapter card. The functionality of network interface 124 may be implemented using both hardware and software. Both a random-access memory (RAM) and a read-only memory (ROM) may be included on a network card implementation of network interface 124. One or more application specific integrated circuits (ASICs) may be used to provide the functionality of network interface 124.

[0046] In one embodiment, a data storage model may be developed which seeks to optimize data layouts for both user data and corresponding error correction code (ECC) information. In one embodiment, the model is based at least in part on characteristics of the storage devices within a storage system. For example, in a storage system, which utilizes solid-state storage technologies, characteristics of the particular devices may be used to develop a model for the storage system and may also serve to inform corresponding data storage arrangement algorithms. For example, if particular storage devices being used exhibit a change in reliability over time, such a characteristic may be accounted for in dynamically changing a data storage arrangement.

[0047] Generally speaking, any model which is developed for a computing system is incomplete. Often, there are simply too many variables to account for in a real world system to completely model a given system. In some cases, it may be possible to develop models which are not complete but which are nevertheless valuable. As discussed more fully below, embodiments are described wherein a storage system is modeled based upon characteristics of the underlying devices. In various embodiments, selecting a data storage arrangement is performed based on certain predictions as to how the system may behave. Based upon an understanding of the characteristics of the devices, certain device behaviors are more predictable than others. However, device behaviors may change over time, and in response, a selected data layout may also be changed. As used herein, characteristics of a device may refer to characteristics of the device as a whole, characteristics of a sub-portion of a device such as a chip or other component, characteristics of an erase block, or any other characteristics related to the device.

[0048] Intra-Device Redundancy

[0049] Turning now to FIG. 2, a generalized block diagram illustrating one embodiment of a dynamic intra-device redundancy scheme is shown. As is well known to those skilled in the art, one of several intra-device redundancy schemes may be chosen to reduce the effects of latent sector errors in a storage device. The term "sector" typically refers to a basic unit of storage on a HDD, such as a segment within a given track on the disk. Here, the term "sector" may also refer to a basic unit of allocation on a SSD.

[0050] An allocation unit within an SSD may include one or more erase blocks within an SSD. Referring to FIG. 2, the user data 210 may refer to both stored data to be modified and accessed by end-users and inter-device error-correction code (ECC) data. The inter-device ECC data may be parity information generated from one or more pages on other storage devices holding user data. For example, the inter-device ECC data may be parity information used in a RAID data layout architecture. The user data 210 may be stored within one or more pages included within one or more of the storage devices 176a-176k. In one embodiment, each of the storage devices 176a-176k is an SSD.

[0051] An erase block within an SSD may comprise several pages. As described earlier, in one embodiment, a page may include 4KB of data storage space. An erase block may include 64 pages, or 256KB. In other embodiments, an erase block may be as large as 1 megabyte (MB), and include 256 pages. An allocation unit size may be chosen in a manner to provide both sufficiently large sized units and a relatively low number of units to reduce overhead tracking of the allocation units. In one embodiment, one or more state tables may maintain a state of an allocation unit (allocated, free, erased, error), a wear level, and a count of a number of errors (correctable and/or uncorrectable) that have occurred within the allocation unit. In various embodiments, the size of an allocation unit may be selected to balance the number of allocation units available for a give device against the overhead of maintaining the allocation units. For example, in one embodiment the size of an allocation unit may be selected to be approximately 1/100th of one percent of the total storage capacity of an SSD. Other amounts of data storage space for pages, erase blocks and other unit arrangements are possible and contemplated.

[0052] Latent sector errors (LSEs) occur when a given sector or other storage unit within a storage device is inaccessible. A read or write operation may not be able to complete for the given sector. In addition, there may be an uncorrectable error-correction code (ECC) error. An LSE is an error that is undetected until the given sector is accessed. Therefore, any data previously stored in the given sector may be lost. A single LSE may lead to data loss when encountered during RAID reconstruction after a storage device failure. For an SSD, an increase

in the probability of an occurrence of another LSE may result from at least one of the following statistics: device age, device size, access rates, storage compactness and the occurrence of previous correctable and uncorrectable errors. To protect against LSEs and data loss within a given storage device, one of a multiple of intra-device redundancy schemes may be used within the given storage device.

[0053] An intra-device redundancy scheme utilizes ECC information, such as parity information, within the given storage device. This intra-device redundancy scheme and its ECC information corresponds to a given device and may be maintained within a given device, but is distinct from ECC that may be internally generated and maintained by the device itself. Generally speaking, the internally generated and maintained ECC of the device is invisible to the system within which the device is included. The intra-device ECC information included within the given storage device may be used to increase data storage reliability within the given storage device. This intra-device ECC information is in addition to other ECC information that may be included within another storage device such as parity information utilized in a RAID data layout architecture.

[0054] A highly effective intra-device redundancy scheme may sufficiently enhance a reliability of a given RAID data layout to cause a reduction in a number of devices used to hold parity information. For example, a double parity RAID layout may be replaced with a single parity RAID layout if there is additional intra-device redundancy to protect the data on each device. For a fixed degree of storage efficiency, increasing the redundancy in an intra-device redundancy scheme increases the reliability of the given storage device. However, increasing the redundancy in such a manner may also increase a penalty on the input/output (I/O) performance of the given storage device.

[0055] In one embodiment, an intra-device redundancy scheme divides a device into groups of locations for storage of user data. For example, a division may be a group of locations within a device that correspond to a stripe within a RAID layout as shown by stripes 250a-250c. User data or inter-device RAID redundancy information may be stored in one or more pages within each of the storage devices 176a-176k as shown by data 210. Within each storage device, intra-device error recovery data 220 may be stored in one or more pages. As used herein, the intra-device error recovery data 220 may be referred to as intra-device redundancy data 220. As is well known by those skilled in the art, the intra-device redundancy data 220 may be obtained by performing a function on chosen bits of information within the data 210. An XOR-based operation may be used to derive parity information to store in the intra-device redundancy data

220. Other examples of intra-device redundancy schemes include single parity check (SPC), maximum distance separable (MDS) erasure codes, interleaved parity check codes (IPC), hybrid SPC and MDS code (MDS+SPC), and column diagonal parity (CDP). The schemes vary in terms of delivered reliability and overhead depending on the manner the data 220 is computed. In addition to the above described redundancy information, the system may be configured to calculate a checksum value for a region on the device. For example, a checksum may be calculated when information is written to the device. This checksum is stored by the system. When the information is read back from the device, the system may calculate the checksum again and compare it to the value that was stored originally. If the two checksums differ, the information was not read properly, and the system may use other schemes to recover the data. Examples of checksum functions include cyclical redundancy check (CRC), MD5, and SHA-1.

[0056] As shown in stripes 250a-250c, the width, or number of pages, used to store the data 210 within a given stripe may be the same in each of the storage devices 176a-176k. However, as shown in stripes 250b-250c, the width, or number of pages, used to store the intra-device redundancy data 220 within a given stripe may not be the same in each of the storage devices 176a-176k. In one embodiment, changing characteristics or behaviors of a given storage device may determine, at least in part, the width used to store corresponding intra-device redundancy data 220. For example, as described above, Flash cells experience program disturb errors and read disturb errors, wherein programming or reading a page may disturb nearby pages and cause errors within these nearby pages. When a storage device is aging and producing more errors, the amount of corresponding intra-device redundancy data 220 may increase. For example, prior to a write operation for stripe 250b, characteristics of each of the storage devices 176a-176k may be monitored and used to predict an increasing error rate. A predicted increase in errors for storage devices 176c and 176j may be detected. In response, the amount of intra-device redundancy data 220 may be increased for storage devices 176c and 176j. In the example of stripes 250a and 250b of FIG. 2, an increase in the amount of protection data stored can be seen for storage devices 176c and 176j for stripes 250a and 250b. For example, now, rather than protecting storage devices 176c and 176j with single parity, these devices may be protected with double parity or triple parity. It is noted that increasing the amount of intra-device protection for devices 176c and 176j does not necessitate a corresponding increase in other devices of the same stripe. Rather, data for the stripe may have differing levels of protection in each device as desired.

[0057] In various embodiments, increases or decreases in a given level of data protection may occur on a selective basis. For example, in one embodiment, an increase in protection may occur

only for storage devices that are detected to generate more errors, such as storage devices 176c and 176j in the above example. In another embodiment, an increase in protection may occur for each of the storage devices 176a-176k when storage devices 176c and 176j are detected to generate more errors. In one embodiment, increasing the amount of intra-device protection on a parity device such as device 176k may require a reduction in the amount of data protected within the stripe. For example, increasing the amount of intra-device data stored on a parity device for a given stripe will necessarily reduce an amount of parity data stored by that device for data within the stripe. If this amount of parity data is reduced to an amount that is less than that needed to protect all of the data in the stripe, then data within the stripe must be reduced if continued parity protection is desired. As an alternative to reducing an amount of data stored within the stripe, a different device could be selected for storing the parity data. Various options are possible and are contemplated. It is also noted that while FIG. 2 and other figures described herein may depict a distinct parity device (e.g., 176k), in various embodiments the parity may be distributed across multiple devices rather than stored in a single device. Accordingly, the depiction of a separate parity device in the figures may generally be considered a logical depiction for ease of discussion.

[0058] Referring now to FIG. 3, one embodiment of a method 300 for adjusting intra-device protection in a data storage subsystem is shown. The components embodied in network architecture 100 and data storage arrays 120a-120b described above may generally operate in accordance with method 300. The steps in this embodiment are shown in sequential order. However, some steps may occur in a different order than shown, some steps may be performed concurrently, some steps may be combined with other steps, and some steps may be absent in another embodiment.

[0059] In block 302, a first amount of space for storing user data in a storage device is determined. This user data may be data used in end-user applications or inter-device parity information used in a RAID architecture as described earlier regarding data 210. This first amount of space may comprise one or more pages within a storage device as described earlier. In one embodiment, a global RAID engine 178 within the storage controller 174 receives behavioral statistics from each one of the storage devices 176a-176m. For a given device group comprising two or more of the storage devices 176a-176m, the global RAID engine 178 may determine both a RAID data layout and an initial amount of intra-device redundancy to maintain within each of the two or more storage devices. In block 304, the RAID engine 178 may determine a second amount of space for storing corresponding intra-device protection data in a

storage device. This second amount of space may comprise one or more pages within a storage device. The intra-device protection data may correspond to the to intra-device redundancy data 220 described earlier.

[0060] In block 306, data is written in the first amount of space within each storage device included within a given device group. In one embodiment, both user data and inter-device parity information is written as a single RAID stripe across multiple storage devices included within the given device group. Referring again to FIG. 2, the width for the corresponding data being written is the same within each storage device. In block 308, the intra-device protection data is generated by an ECC algorithm, an XOR-based algorithm, or any other suitable algorithm. In addition, the system may generate a checksum to help identify data that has not been retrieved properly. In block 310, the generated intra-device protection data is written in the second amount of space in the storage devices.

[0061] In block 312, the RAID engine 178 may monitor behavior of the one or more storage devices. In one embodiment, the RAID engine 178 may include a model of a corresponding storage device and receive behavioral statistics from the storage device to input to the model. The model may predict behavior of the storage device by utilizing known characteristics of the storage device. For example, the model may predict an upcoming increasing error rate for a given storage device. If the RAID engine 178 detects characteristics of a given storage device which affect reliability (conditional block 314), then in block 316, the RAID engine may adjust the first amount and the second amount of space for storing data and corresponding intra-device redundancy data. For example, the RAID engine may be monitoring the statistics described earlier such as at least device age, access rate and error rate. Referring again to FIG. 2, the RAID engine 178 may detect storage devices 176c and 176j have an increase in a number of errors. Alternatively, the RAID engine may predict an increase in a number of errors for storage devices 176c and 176j. Accordingly, prior to writing the second stripe 250b, the RAID engine 178 may adjust a number of pages used to store data 210 and data 220 in each of the storage devices 176a-176k. Similarly, the RAID engine 178 may detect storage device 176b has decreased reliability. Therefore, prior to writing the third stripe 250c, the RAID engine 178 may again adjust a number of pages used to store data 210 and data 220 in each of the storage devices 176a-176k.

[0062] Monitoring Storage Device Characteristics

[0063] Turning now to FIG. 4, a generalized block diagram of one embodiment of a storage subsystem is shown. Each of the one or more storage devices 176a-176m may be partitioned in

one of one or more device groups 173a-173m. Other device groups with other devices may be present as well. One or more corresponding operation queues and status tables for each storage device may be included in one of the device units 400a-400w. These device units may be stored in RAM 172. A corresponding RAID engine 178a-178m may be included for each one of the device groups 173a-173m. Each RAID engine 178 may include a monitor 410 that tracks statistics for each of the storage devices included within a corresponding device group. Data layout logic 420 may determine an amount of space to allocate within a corresponding storage device for user data, inter-device redundancy data and intra-device redundancy data. The storage controller 174 may comprise other control logic 430 to perform at least one of the following tasks: wear leveling, garbage collection, I/O scheduling, deduplication and protocol conversion for incoming and outgoing packets.

[0064] Turning now to FIG. 5, a generalized block diagram of one embodiment of a device unit is shown. A device unit may comprise a device queue 510 and tables 520. Device queue 510 may include a read queue 512, a write queue 514 and one or more other queues such as other operation queue 516. Each queue may comprise a plurality of entries for storing one or more corresponding requests 530a-530d. For example, a device unit for a corresponding SSD may include queues to store at least read requests, write requests, trim requests, erase requests and so forth. Tables 520 may comprise one or more state tables 522a-522b, each comprising a plurality of entries for storing state data, or statistics, 530. It is also noted that while the queues and tables are shown to include a particular number of entries in this and other figures, the entries themselves do not necessarily correspond to one another. Additionally, the number of queues, tables, and entries may vary from that shown in the figure and may differ from one another.

[0065] Referring now to FIG. 6, a generalized block diagram illustrating one embodiment of a state table corresponding to a given device is shown. In one embodiment, such a table may include data corresponding to state, error and wear level information for a given storage device, such as an SSD. A corresponding RAID engine may have access to this information, which may allow the RAID engine to dynamically change space allocated for data storage and schemes used for both inter-device protection and intra-device protection. In one embodiment, the information may include at least one or more of a device age 602, an error rate 604, a total number of errors detected on the device 606, a number of recoverable errors 608, a number of unrecoverable errors 610, an access rate of the device 612, an age of the data stored 614 and one or more allocation states for allocation spaces 616a-616n. The allocation states may include filled, empty, error and so forth.

[0066] Flexible RAID Layout

[0067] Turning now to FIG. 7, a generalized block diagram illustrating one embodiment of a flexible RAID data layout architecture is shown. A RAID engine may determine a level of protection to use for storage devices 176a-176k. For example, a RAID engine may determine to utilize RAID double parity for the storage devices 176a-176k. The inter-device redundancy data 240 may represent the RAID double parity values generated from corresponding user data. In one embodiment, storage devices 176j and 176k may store the double parity information. It is understood other levels of RAID parity protection are possible and contemplated. In addition, in other embodiments, the storage of the double parity information may rotate between the storage devices rather than be stored within storage devices 176j and 176k for each RAID stripe. The storage of the double parity information is shown to be stored in storage devices 176j and 176k for ease of illustration and description.

[0068] Referring now to FIG. 8, a generalized block diagram illustrating another embodiment of a flexible RAID data layout architecture is shown. Similar to the example shown in FIG. 7, double parity may be used for the storage devices 176a-176k. Although a RAID double parity is described in this example, any amount of redundancy in a RAID data layout architecture may be chosen.

[0069] During operation, the RAID engine 178 may monitor characteristics of the storage devices 176a-176k and determine the devices are exhibiting a reliability level higher than an initial or other given reliability level. In response, the RAID engine 178 may change the RAID protection from a RAID double parity to a RAID single parity. In other RAID data layout architectures, another reduction in the amount of supported redundancy may be used. In other embodiments, the monitoring of storage devices 176a-176k and changing a protection level may be performed by other logic within storage controller 174.

[0070] Continuing with the above example, only single parity information may be generated and stored for subsequent write operations executing on a given RAID stripe. For example, storage device 176k may not be used in subsequent RAID stripes for write operations after the change in the amount of supported redundancy. In addition, data stored in storage device 176k may be invalidated, thereby freeing the storage. Pages corresponding to freed data in storage device 176k may then be reallocated for other uses. The process of reducing an amount of parity protection and freeing space formerly used for storing parity protection data may be referred to as "parity shredding". In addition, in an embodiment wherein storage device 176k is an SSD, one or

more erase operations may occur within storage device 176k prior to rewriting the pages within stripe 250a.

[0071] Continuing with the above example of parity shredding, the data stored in the reallocated pages of storage device 176k within stripe 250a after parity shredding may hold user data or corresponding RAID single parity information for other RAID stripes that do not correspond to stripe 250a. For example, the data stored in storage devices 176a-176j within stripe 250a may correspond to one or more write operations executed prior to parity shredding. The data stored in storage device 176k within stripe 250a may correspond to one or more write operations executed after parity shredding. Similarly, the data stored in storage devices 176a-176j within stripe 250b may correspond to one or more write operations executed prior to parity shredding. The pages in storage device 176k within stripe 250b may be freed, later erased, and later rewritten with data corresponding to one or more write operations executed after the change in the amount of supported redundancy. It is noted that this scheme may be even more effective when redundancy information is rotated across storage devices. In such an embodiment, space that is freed by shredding will likewise be distributed across the storage devices.

[0072] Referring again to FIG. 8, the deallocated pages shown in storage device 176k within stripe 250c represent storage locations that may have previously stored RAID double parity information prior to parity shredding. However, now these pages are invalid and have not yet been reallocated. Particular characteristics of an SSD determine the manner and the timing of both freeing and reallocating pages within storage device 176k in the above example. Examples of these characteristics include at least erasing an entire erase block prior to reprogramming (rewriting) one or more pages. As can be seen from FIG. 8, when parity is shredded, it is not necessary to shred an entire device. Rather, parity may be shredded for individual stripes as desired. Similarly, parity protection for a stripe may be increased may adding protection data stored on an additional device to a stripe.

[0073] Referring now to FIG. 9, one embodiment of a method for dynamically determining a RAID layout is shown. The components embodied in network architecture 100 and data storage arrays 120a-120b described above may generally operate in accordance with method 900. In FIG. 9, two processes 910 and 920 are shown. Each of the processes may operate concurrently, or in a given order. Further, the steps in this embodiment are shown in sequential order. However, some steps may occur in a different order than shown, some steps may be performed concurrently, some steps may be combined with other steps, and some steps may be absent in another embodiment. Block 910 illustrates a process whereby a storage control system monitors the

characteristics and behaviors of storage devices in the system (block 912). For example, characteristics such as those described in FIG. 6 may be observed and/or recorded. If a particular condition is detected, such as a change in reliability (decision block 914), then a change in the amount of protection used for stored data may be made (block 916). For example, when given devices are relatively young in age, the reliability of the devices may not be known (e.g., the devices may suffer "infant mortality" and fail at a relatively young age). Therefore, one or more extra storage devices per RAID stripe may be used to store parity information. At a later time, this extra protection may be removed when the devices prove over time that they are reliable. In various embodiments, characteristics regarding error rates may be maintained for devices. For example, characteristics concerning correctable and/or uncorrectable errors may be maintained and used to make decisions regarding the reliability of a given device. Based upon this information, the storage controller may dynamically alter various levels of protection for a device or stripe.

[0074] Block 920 of FIG. 9 generally illustrates a process whereby at the time a stripe or other portion of storage is to be allocated (decision block 922), a determination regarding the layout and protection level to use for the data may be made (block 924). It is noted that the process of block 910 could be performed at this time. Alternatively, levels of protection may have been determined by process 910 and stored. The determination of block 924 could then be based upon that stored data. In one embodiment, once a given layout has been determined, the particular devices to be used for the layout may be selected from a group of devices (block 925). For example, in one embodiment a group of 20 devices may be available for use. If a layout of 5+2 is determined, then any seven devices may be selected for use from the group of 20. Additionally, it is noted that a subsequent write with a selected 5+2 layout need not use the same 7 devices. Subsequent to determining the layout, protection level, and devices for the stripe, the stripe may be written (block 926).

[0075] In various embodiments, the RUSH algorithm may be utilized to determine which devices on which the data and redundancy information for a given stripe will reside . For example, the RUSH algorithm may be used to select the particular devices to utilize for an 8+2 RAID layout for a given stripe in storage devices 176a-176k. Generally speaking, as used herein, an M+N layout may generally describe a layout which includes M data devices and N parity devices for a given data stripe. Additionally, as discussed above, parity may be distributed across the devices rather than fully located within particular devices. Accordingly, an 8+2 layout may include data and parity striped across 10 devices – with 8 of the devices storing data and two

of the devices storing parity. On a subsequent occasion, a layout of 12+2 may be selected. In this manner, the desired layout and protection characteristics may be determined dynamically at the time a write (e.g., a stripe) is to be written. In one embodiment, storage devices 176a-176k may include more than 10 storage devices, such as 30, 50 or more storage devices. However, for a stripe with an 8+2 layout, only 10 of the storage devices are utilized . It is noted that any 10 of the devices may be selected and any suitable algorithm may be used for selecting the 10 devices for use in storing the stripe. For example, the CRUSH algorithm could be used to select which 10 of the storage devices 176a-176k to utilize for a given 8+2 RAID layout.

[0076] In one example of a chosen 8+2 RAID layout for storage devices 176a-176k, 2 of the storage devices may be used to store error correcting code (ECC) information, such as parity information. This information may be used to perform reconstruct read requests. Referring again to FIG. 8, the storage devices 176j and 176k may be selected to store RAID double parity information in this example. Again, the parity information may be stored in a rotated fashion between each of the storage devices 176a-176k included within the RAID array, rather than consistently stored in the same storage devices. For ease of illustration and description, the storage devices 176j and 176k are described as storing RAID double parity.

[0077] In block 926, during execution of a write operation, metadata, user data, intra-device parity information and inter-device parity information may be written as a RAID stripe across multiple storage devices included within the RAID array. In block 912, the RAID engine 178 may monitor behavior of the one or more storage devices within the RAID array. In one embodiment, the RAID engine 178 may include a monitor 410 and data layout logic 420 as shown in FIG. 4. The RAID engine 178 may monitor at least an age of a given storage device, a number and a type of errors, detected configuration changes since a last allocation of data, an age of given data, a current usage of storage space in the RAID array, and so forth.

[0078] The data, which is monitored by the RAID engine 178, may be stored in RAM 172, such as in one of the device units 400a-400w shown in FIG. 4. Tables may be used to store this data, such as the examples shown in FIG. 5 and FIG. 6. The logic included within a corresponding RAID engine may both detect and predict behavior of storage devices by monitoring updated statistics of the storage devices. For example, the model may predict an upcoming increasing error rate for a given storage device.

[0079] If increased reliability of the storage device(s) is detected (conditional block 908), then in block 910, the RAID engine may decrease the level of data protection within the system. For example, in one embodiment the amount of parity information stored in the storage subsystem

may be reduced. Regarding the above example, the RAID engine may decrease the RAID double parity to RAID single parity for the corresponding 8+2 RAID array, converting it to an 8+1 RAID array. In other examples a given RAID array may be utilizing an N-level amount of redundancy, or parity, in a RAID architecture prior to block 916. In block 916, the RAID engine may determine to utilize an (N-m)-level amount of redundancy, wherein $N > 1$ and $1 \leq m < N$. Therefore, during subsequent write operations for a given RAID stripe, there will be m fewer storage devices written to within the given RAID stripe.

[0080] In order to reduce the level of data protection within the system, the RAID engine (or another component) may perform parity shredding as described earlier. Subsequently, the storage controller 174 may reallocate those pages which were freed as a result of the shredding operation to be used in subsequent write operations.

[0081] As each of the storage devices 176a-176k both age and fill up with data, extra parity information may be removed from the RAID array as described above. The metadata, the user data, corresponding intra-device redundancy information and some of the inter-device redundancy information remains. Regarding the above example with an 8+2 RAID array, the information stored in storage devices 176a-176j remains. However, extra inter-device redundancy information, or extra parity information, may be removed from the RAID array. For example, extra parity information stored in storage device 176k may be removed from the RAID stripes.

[0082] The information that remains, such as the information stored in storage devices 176a-176j in the above example, may remain in place. The storage space storing the extra parity information, such as the corresponding pages in storage device 176k in the above example, may be reused and reallocated for subsequent write operations. In one embodiment, each new allocation receives a new virtual address. Each new allocation may have any given size, any given alignment or geometry, and may fit in any given storage space (either virtual or physical). In one embodiment, each one of the storage devices 176a-176k and each allocated page within a storage device have a header comprising identification information. This identification information may allow the reuse of storage space for freed extra parity information without changing a given configuration.

[0083] In an embodiment wherein one or more of the storage devices 176a-176k is an SSD, an erase block is erased prior to reprogramming one or more pages within the erase block. Therefore, in an embodiment wherein storage device 176k is an SSD, corresponding erase blocks are erased prior to reprogramming freed pages in storage device 176k. Regarding the above

example with an original 8+2 RAID array, one or more erase blocks are erased in storage device 176k within stripes 250a-250b prior to reprogramming pages with data 210. The original 8+2 RAID array is now an 8+1 RAID array with storage device 176j providing the single parity information for RAID stripes written prior to the parity shredding.

[0084] As is well known to those skilled in the art, during a read or write failure for a given storage device, data may be reconstructed from the supported inter-device parity information within a corresponding RAID stripe. The reconstructed data may be written to the storage device. However, if the reconstructed data fails to be written to the storage device, then all the data stored on the storage device may be rebuilt from corresponding parity information. The rebuilt data may be relocated to another location. With Flash memory, a Flash Translation Layer (FTL) remaps the storage locations of the data. In addition, with Flash memory, relocation of data includes erasing an entire erase block prior to reprogramming corresponding pages within the erase block. Maintaining mapping tables at a granularity of erase blocks versus pages allows the remapping tables to be more compact. Further, during relocation, extra pages that were freed during parity shredding may be used.

[0085] Offset Parity

[0086] Turning now to FIG. 10, a generalized block diagram illustrating yet another embodiment of a flexible RAID data layout architecture is shown. Similar to the generalized block diagram shown in FIG. 8, a flexible RAID data layout architecture may be used. The storage devices 176a-176k comprise multiple RAID stripes laid out across multiple storage devices. Although each of the storage devices 176a-176k comprises multiple pages, only page 1010 and page 1020 are labeled for ease of illustration. In the example shown, a double parity RAID data layout is chosen, wherein storage devices 176j and 176k store double parity information.

[0087] Each of the pages in the storage devices 176a-176k stores a particular type of data. Some pages store user data 210 and corresponding generated inter-device parity information 240. Other pages store corresponding generated intra-device parity information 220. Yet other pages store metadata 242. The metadata 242 may include page header information, RAID stripe identification information, log data for one or more RAID stripes, and so forth. In addition to inter-device parity protection and intra-device parity protection, each of the pages in storage devices 176a-176k may comprise additional protection such as a checksum stored within each given page. In various embodiments, the single metadata page at the beginning of each stripe may be rebuilt from the other stripe headers. Alternatively, this page could be at a different offset

in the parity shard so the data can be protected by the inter-device parity. A "shard" represents a portion of a device. Accordingly, a parity shard refers to a portion of a device storing parity data.

Physical Layer

[0088] In various embodiments, the systems described herein may include a physical layer through which other elements of the system communicate with the storage devices. For example, scheduling logic, RAID logic, and other logic may communicate with the storage devices via a physical layer comprising any suitable combination of software and/or hardware. In general, the physical layer performs a variety of functions including providing access to persistent storage, and performing functions related to integrity of data storage.

[0089] FIG. 11A illustrates one embodiment of a hypothetical device layout for a 500GB device. In various embodiments, the storage devices described herein may be formatted with a partition table 1101 at the beginning of the device, and a copy of the partition table at the end of the device. Additionally, a device header 1103 may be stored in the first and last blocks. For example, in a flash based storage device, a device header may be stored in the first and last erase blocks. As previously discussed, an erase block is a flash construct that is typically in the range of 256KB-1MB. Additional unused space in the first erase block may be reserved (padding 1105). The second erase block in each device may be reserved for writing logging and diagnostic information 1107. The rest of the erase blocks in between are divided into Allocation Units (AUs) 1109 of a multiple erase blocks. The AU size may be chosen so there are a reasonable number of AUs per device for good allocation granularity. In one embodiment, there may be something in the range of 10,000 AUs on a device to permit allocation in large enough units to avoid overhead, but not too many units for easy tracking. Tracking of the state of an AU (allocated/free/erased/bad) may be maintained an AU State Table. The wear level of an AU may be maintained in a Wear Level Table, and a count of errors may be maintained in an AU Error Table.

[0090] In various embodiments, the physical layer allocates space in segments which include one segment shard in each device across a set of devices (which could be on different nodes). FIG. 11B depicts one embodiment of a segment and various identifiable portions of that segment in one possible segment layout. In the embodiment shown, a single segment is shown stored in multiple devices. Illustrated are data devices Data 0 – Data N, and parity devices Parity P and Parity Q. In one embodiment, each segment shard includes one or more allocation units on a

device such that the size of the shard is equal on each device. Segment shard 1123 is called out to illustrate a segment shard. Also illustrated if FIG. 11B, is an I/O read size 1127 which in one embodiment corresponds to a page. Also shown is an I/O parity chunk 1129 which may include one or more pages of page parity for the I/O shard.

[0091] In one embodiment, each segment will have its own geometry which may include one or more of the following parameters:

(1)RAID level - The RAID level used for cross device protection in the segment. This may determine mirroring, parity, or ECC RAID and how many segment shards contain parity.

(2) Device Layout I/O shard size - This represents the size used to stripe across each device during a write. This will typically be in the range of 256KB to 1MB and probably be a multiple of the erase block size on each device. FIG. 11B calls out I/O shard size 1125 for purposes of illustration.

(3)I/O read size - This is a logical read size. Each I/O shard may be formatted as a series of logical pages. Each page may in turn include a header and a checksum for the data in the page. When a read is issued it will be for one or more logical pages and the data in each page may be validated with the checksum.

(4)I/O shard RAID level - The I/O shard has intra-shard parity to handle latent errors found during a rebuild. This parameter determines what type of parity is used for intra-shard protection and thus how many copies of the intra-shard parity will be maintained.

(5)I/O parity chunk – In various embodiments, the storage devices may do ECC on a page basis. Consequently, if an error is seen it is likely to indicate failure of an entire physical page. The I/O parity chunk is the least common multiple of the physical page size on each device in the segment and the intra-shard parity is calculated by striping down the I/O shard in the larger of the I/O parity chunks or the I/O read size. Included may be one or more pages of page parity. In various embodiments, this parity may be used to rebuild data in the event of a failed checksum validation.

[0092] In various embodiments, as each new segment is written a RAID geometry for the segment will be selected. Selection of the RAID geometry may be based on factors such as the current set of active nodes and devices, and the type of data in the segment. For example if 10 nodes or devices are available then an (8+2) RAID 6 geometry may be chosen and the segment striped across the nodes to withstand two device or node failures. If a node then fails, the next

segment may switch to a (7+2) RAID 6 geometry. Within the segment some of the segment shards will contain data and some will contain ECC (e.g., parity).

[0093] In one embodiment, there are five types of segments. Three of these segments correspond to the AU State Table, the AU Error Table, and the Wear Level Table. In some embodiments, these three segments may be mirrored for additional protection. In addition to these three segments, there are metadata segments which may also be additionally protected through mirroring. Finally there are Data segments which hold client blocks and log information. The log information contains update information associated with the client blocks in the segment. The data segments will likely be protected by RAID 6 as illustrated in FIG. 11B with Parity P and Parity Q shards. In addition to the above, a segment table is maintained as an in memory data structure that is populated at startup with information from the headers of all the segment shards. In some embodiments, the table may be cached completely on all nodes so any node can translate a storage access to a physical address. However, in other embodiments an object storage model may be used where each node may have a segment table that can take a logical reference and identify the segment layout node where the data is stored. Then the request would be passed to the node to identify the exact storage location on the node. FIG. 11B also depicts segment tail data which identifies any (volume, snapshot) combinations that take up a significant amount of space in the segment. When snapshots are removed, a data scrubber may help identify segments for garbage collection based on this data.

[0094] In one embodiment, the basic unit of writing is the segio which is one I/O shard on each of the devices in the segment. Each logical page in the segio is formatted with a page header that contains a checksum (which may be referred to as a "media" checksum) of the page so the actual page size for data is slightly smaller than one page. For pages in the parity shards of a segment the page header is smaller so that the page checksums in the data page are protected by the parity page. The last page of each I/O shard is a parity page that again has a smaller header and protects all the checksums and page data in the erase block against a page failure. The page size referred to here is the I/O read size which may be one or more physical flash pages. For some segments, a read size smaller than a physical page may be used. This may occur for metadata where reads to lookup information may be index driven and smaller portion of data may be read while still obtaining the desired data. In such a case, reading half a physical page would mean tying up the I/O bus (and network) with less data and validating (e.g., checksumming) less data. To support a read size smaller than a physical page, an embodiment may include multiple parity pages at the end of the erase block such that the total size of all the parity pages is equal to the flash page size.

[0095] As the wear level of an erase block increases, the likelihood of an error increases. In addition to tracking wear levels, data may be maintained regarding observed how often errors are seen on an erase block and blocks with a higher probability of error identified. For some erase blocks, it may be decided to keep double or triple error correcting parity at the end of the erase block instead of the single RAID 5 parity. In this case, the data payload of the segio may be reduced accordingly. It may only be necessary to reduce the poor erase block within the segio, rather than all the erase blocks. The page headers in the erase block may be used to identify which pages are parity and which are data.

[0096] Whenever a page is read from storage, the contents may be validated using the page checksum. If the validation fails, a rebuild of the data using the erase block parity may be attempted. If that fails, then cross device ECC for the segment may be used to reconstruct the data.

[0097] In data segments the payload area may be divided into two areas. There will be pages formatted as log data which may include updates related to stored client blocks. The remainder of the payload area may contain pages formatted as client blocks. The client block data may be stored in a compressed form. Numerous compression algorithms are possible and are contemplated. Additionally, in various embodiments Intel® Advanced Encryption Standard instructions may be used for generating checksums. Additionally, there may be a header for the client block that resides in the same page as the data and contains information needed to read the client block, including an identification of the algorithm used to compress the data. Garbage collection may utilize both the client block header and the log entries in the segio. In addition, the client block may have a data hash which may be a checksum of the uncompressed data used for deduplication and to check the correctness of the decompressed data.

[0098] In some embodiments, segments and segios may have a monotonically increasing ID number used to order them. As part of writing a segio, a logical layer can record dependencies on prior flushes. At startup, the physical layer may build an ordered list of segments and segios and if a segio is dependent on another uncompleted segio it may be rolled back and not considered to have been written.

Wear Level Table

[0099] The Wear Level Table (WLT) for each device may be stored in a segment local to each device. The information may also be stored in the header of each segment shard. In one embodiment, the wear information is an integer that represents the number of times the allocation

unit has been erased and reused. As the wear information may not be accurate, a flush of the table to the device may be performed when there has been a certain amount of activity or when the system has been idle for a reasonable period. The WLT may also be responsible for cleaning up old WLT segments as it allocates new ones. To add an extra layer of protection, old copies may be maintained before freeing them. For example, a table manager may ensure that it keeps the previous erase block and the current erase block of WLT entries at all times. when it allocates a new segment it won't free the old segment until it has written into the second erase block of the new segment.

AU State Table

[00100] The AU State Table (AST) tracks the state of each AU. The states include Free, Allocated, Erased and Bad. The AST may be stored in a segment on the device. Changing a state to Allocated or Free may be a synchronous update, while changing a state to Bad or Erased may be an asynchronous update. This table may generally be small enough and have enough updates that updates may be logged in NVRAM. The AST may be responsible for cleaning up old AST segments as it allocates new ones. Since the AST can be completely recovered by scanning the first block of each AU on the drive, there is no need to keep old copies of the AST.

AU Error Table

[00101] The AU Error Table (AET) may be used to track the number of recoverable errors and unrecoverable errors within each AU. The AET is stored in a segment on the device and each field may be a two byte integer. With four bytes per AU the entire table may be relatively small.

[00102] Referring now to FIG. 11C, a generalized block diagram illustrating one embodiment of data storage arrangements within different page types is shown. In the embodiment shown, three page types are shown although other types are possible and contemplated. The shown page types include page 1110 comprising metadata 1150, page 1120 comprising user data 1160, and page 1130 comprising parity information 1170 (inter-device or intra-device). Each of the pages 1110-1130 comprises metadata 1140, which may include header and identification information. In addition, each of the pages 1110-1130 may comprise intra-page error recovery data 1142, such as a corresponding checksum or other error detecting and/or correcting code. This checksum value may provide added protection for data stored in storage devices 176a-176k in a given device group.

[00103] Further, page 1130 may comprise inter-page error recovery data 1144. The data 1144 may be ECC information derived from the intra-page data 1142 stored in other storage devices. For example, referring again to FIG. 10, each page within storage device 176j, which stores inter-device parity information 240, may also store inter-page error recovery data 1144. The data 1144 may be a parity, a checksum, or other value generated from intra-page error recovery data 1142 stored in one or more of the storage devices 176a-176i. In one embodiment, the data 1144 is a checksum value generated from one or more other checksum values 1142 stored in other storage devices. In order to align data 1144 in a given page in storage device 176j with data 1142 in a corresponding page in one or more of the storage devices 176a-176i, padding 1146 may be added to the corresponding pages.

[00104] In one embodiment, end-user applications perform I/O operations on a sector-boundary, wherein a sector is 512 bytes for HDDs. In order to add extra protection, an 8-byte checksum may be added to form a 520-byte sector. In various embodiments, compression and remapping may be used in a flash memory based system to allow user data to be arranged on a byte boundary rather than a sector boundary. In addition, a checksum (8 byte, 4 byte, or otherwise) may be placed inside a page after a header and before the user data, which may be compressed. This placement is shown in each of pages 1110-1130.

[00105] When an end-user application reads a 512-byte sector, a corresponding page, which may be 2KB-8KB in size in one embodiment, has extra protection with an 8-byte checksum at the beginning of the page. In various embodiments, the page may not be formatted for a non-power of 2 sector size. As shown in pages 1110-1120, the checksum may be offset a few bytes into the page. This offset allows a parity page, such as page 1130, to store both a checksum that covers the parity page and ECC to protect checksums of the other pages.

[00106] For yet another level of protection, data location information may be included when calculating a checksum value. The data 1142 in each of pages 1110-1130 may include this information. This information may include both a logical address and a physical address. Sector numbers, data chunk and offset numbers, track numbers, plane numbers, and so forth may be included in this information as well.

[00107] Alternate Geometries

[00108] Turning now to FIG. 12, a generalized block diagram illustrating one embodiment of a hybrid RAID data layout 1200 is shown. Three partitions are shown although any number of partitions may be chosen. Each partition may correspond to a separate device group, such as

device groups 713a-173b shown in FIG. 1. Each partition comprises multiple storage devices. In one embodiment, an algorithm such as the CRUSH algorithm may be utilized to select which devices to use in a RAID data layout architecture to use for data storage.

[00109] In the example shown, an L+1 RAID array, M+1 RAID array, and N+1 RAID array are shown. In various embodiments, L, M, and N may all be different, the same, or a combination thereof. For example, RAID array 1210 is shown in partition 1. The other storage devices 1212 are candidates for other RAID arrays within partition 1. Similarly, RAID array 1220 illustrates a given RAID array in partition 2. The other storage devices 1222 are candidates for other RAID arrays within partition 2. RAID array 1230 illustrates a given RAID array in partition 3. The other storage devices 1232 are candidates for other RAID arrays within partition 3.

[00110] Within each of the RAID arrays 1210, 1220 and 1230, a storage device P1 provides RAID single parity protection within a respective RAID array. Storage devices D1-DN store user data within a respective RAID array. Again, the storage of both the user data and the RAID single parity information may rotate between the storage devices D1-DN and P1. However, the storage of user data is described as being stored in devices D1-DN. Similarly, the storage of RAID single parity information is described as being stored in device P1 for ease of illustration and description.

[00111] One or more storage devices among each of the three partitions may be chosen to provide an additional amount of supported redundancy for one or more given RAID arrays. For example, storage device Q1 in partition 3 may be combined with each of the RAID arrays 1210, 1220 and 1230. The storage device Q1 may provide RAID double parity information for each of the RAID arrays 1210, 1220 and 1230. This additional parity information is generated and stored when a stripe is written to one of the arrays 1210, 1220, or 1230. Further this additional parity information may cover stripes in each of the arrays 1210, 1220, and 1230. Therefore, the ratio of a number of storage devices storing RAID parity information to a total number of storage devices is lower. For example, if each of the partitions used N+2 RAID arrays, then the ratio of a number of storage devices storing RAID parity information to a total number of storage devices is 3(2)/ (3(N+2)), or 2/(N+2). In contrast, the ratio for the hybrid RAID layout 1200 is (3+1)/(3(N+1)), or 4/(3(N+1)).

[00112] It is possible to reduce the above ratio by increasing a number of storage devices used to store user data. For example, rather than utilize storage device Q1, each of the partitions may utilize a 3N+2 RAID array. In such a case, the ratio of a number of storage devices storing RAID parity information to a total number of storage devices is 2/(3N+2). However, during a

reconstruct read operation, (3N + 1) storage devices receive a reconstruct read request for a single device failure. In contrast, for the hybrid RAID layout 1200, only N storage devices receive a reconstruct read request for a single device failure.

[00113] It is noted each of the three partitions may utilize a different RAID data layout architecture. A selection of a given RAID data layout architecture may be based on a given ratio number of storage devices storing RAID parity information to a total number of storage devices. In addition, the selection may be based on a given number of storage devices, which may receive a reconstruct read request during reconstruction. For example, the RAID arrays 1210, 1220 and 1230 may include geometries such as L+a, M+b and N+c, respectively.

[00114] In addition, one or more storage devices, such as storage device Q1, may be chosen based on the above conditions to provide an additional amount of supported redundancy for one or more of the RAID arrays within the partitions. In an example with three partitions comprising the above RAID arrays and a number Q of storage devices providing extra protection for each of the RAID arrays, a ratio of a number of storage devices storing RAID parity information to a total number of storage devices is (a+b+c+Q)/(L+a+M+b+N+c+Q). For a single device failure, a number of storage devices to receive a reconstruct read request is L, M and N, respectively, for partitions 1 to 3 in the above example. It is noted that the above discussion generally describes 3 distinct partitions in FIG. 12. In such an embodiment, this type of "hard" partitioning where a given layout is limited to a particular group of devices may guarantee that reconstruct reads in one partition will not collide with those in another partition. However, in other embodiments the partitions may not be hard as described above. Rather, given a pool of devices, layouts may be selected from any of the devices. For example, treating the devices as on big pool it is possible to configure layouts such as (L+1, M+1, N+1) + 1. Consequently, there is a chance that geometries overlap and reconstruct reads could collide. If L, M, and N are small relative to the size of the pool then the percentage of reconstruct reads relative to normal reads may be kept low.

[00115] Referring now to FIG. 13, one embodiment of a method 1300 for selecting alternate RAID geometries in a data storage subsystem is shown. The components embodied in network architecture 100 and data storage arrays 120a-120b described above may generally operate in accordance with method 1300. The steps in this embodiment are shown in sequential order. However, some steps may occur in a different order than shown, some steps may be performed concurrently, some steps may be combined with other steps, and some steps may be absent in another embodiment.

[00116] In block 1302, a RAID engine 178 or other logic within a storage controller 174 determines to use a given number of devices to store user data in a RAID array within each partition of a storage subsystem. A RUSH or other algorithm may then be used to select which devices are to be used. In one embodiment, each partition utilizes a same number of storage devices. In other embodiments, each partition may utilize a different, unique number of storage devices to store user data. In block 1304, the storage controller 174 may determine to support a number of storage devices to store corresponding Inter-Device Error Recovery (parity) data within each partition of the subsystem. Again, each partition may utilize a same number or a different, unique number of storage devices for storing RAID parity information.

[00117] In block 1306, the storage controller may determine to support a number Q of storage devices to store extra Inter-Device Error Recovery (parity) data across the partitions of the subsystem. In block 1308, both user data and corresponding RAID parity data may be written in selected storage devices. Referring again to FIG. 12, when a given RAID array is written, such as RAID array 1210 in partition 1, one or more bits of parity information may be generated and stored in storage device Q1 in partition 3.

[00118] If the storage controller 174 detects a condition for performing read reconstruction in a given partition (conditional block 1310), and if the given partition has a sufficient number of storage devices holding RAID parity information to handle a number of unavailable storage devices (conditional block 1312), then in block 1314, the reconstruct read operation(s) is performed with one or more corresponding storage devices within the given partition. The condition may include a storage device within a given RAID array is unavailable due to a device failure or the device operates below a given performance level. The given RAID array is able to handle a maximum number of unavailable storage devices with the number of storage devices storing RAID parity information within the given partition. For example, if RAID array 1210 in partition 1 in the above example is an L+a RAID array, then RAID array 1210 is able to perform read reconstruction utilizing only storage devices within partition 1 when k storage devices are unavailable, where $1 <= k <= a$.

[00119] If the given partition does not have a sufficient number of storage devices holding RAID parity information to handle a number of unavailable storage devices (conditional block 1312), and if there is a sufficient number of Q storage devices to handle the number of unavailable storage devices (conditional block 1316), then in block 1318, the reconstruct read operation(s) is performed with one or more corresponding Q storage devices. One or more storage devices in other partitions, which are storing user data, may be accessed during the read

reconstruction. A selection of these storage devices may be based on a manner of a derivation of the parity information stored in the one or more Q storage devices. For example, referring again to FIG. 12, storage device D2 in partition 2 may be accessed during the read reconstruction, since this storage device may have been used to generate corresponding RAID parity information stored in storage device Q1. If there are not a sufficient number of Q storage devices to handle the number of unavailable storage devices (conditional block 1316), then in block 1320, the corresponding user data may be read from another source or be considered lost.

[00120] It is noted that the above-described embodiments may comprise software. In such an embodiment, the program instructions that implement the methods and/or mechanisms may be conveyed or stored on a computer readable medium. Numerous types of media which are configured to store program instructions are available and include hard disks, floppy disks, CD-ROM, DVD, flash memory, Programmable ROMs (PROM), random access memory (RAM), and various other forms of volatile or non-volatile storage.

[00121] In various embodiments, one or more portions of the methods and mechanisms described herein may form part of a cloud-computing environment. In such embodiments, resources may be provided over the Internet as services according to one or more various models. Such models may include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In IaaS, computer infrastructure is delivered as a service. In such a case, the computing equipment is generally owned and operated by the service provider. In the PaaS model, software tools and underlying equipment used by developers to develop software solutions may be provided as a service and hosted by the service provider. SaaS typically includes a service provider licensing software as a service on demand. The service provider may host the software, or may deploy the software to a customer for a given period of time. Numerous combinations of the above models are possible and are contemplated.

[00122] Although the embodiments above have been described in considerable detail, numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

WHAT IS CLAIMED IS

1. A computer system comprising:

a client computer configured to send read and write requests over a network;

a data storage subsystem coupled to the network configured to receive the read and write requests, the data storage subsystem comprising a plurality of storage devices;

wherein the data storage subsystem further comprises a storage controller configured to:

identify a unit of data stored in the data storage subsystem, wherein said unit of data is stored across at least a first storage device and a second storage device of the plurality of storage devices, each of the first storage device and the second storage device storing intra-device redundancy data corresponding to the unit of data; and

change an amount of intra-device redundancy data corresponding to the unit of data on only the first storage device.

2. The computer system as recited in claim 1, wherein said unit of data is a RAID stripe.

3. The computer system as recited in claim 1, wherein the storage controller is configured to change said amount of intra-device redundancy data stored on the first storage device in response to detecting an indication of a change in reliability of the first storage device.

4. The computer system as recited in claim 3, wherein the storage controller is further configured to store data indicative of characteristics of the plurality of storage devices, and wherein said change in reliability is based at least in part on the characteristics.

5. The computer system as recited in claim 1, wherein the storage controller is configured to change said amount of intra-device redundancy data stored on the first storage device in response to detecting a characteristic of the first storage device, the characteristic being one or more of an error rate, an access rate, an age, an occupancy level, an expected life of data, and an age of stored data.

6. The computer system as recited in claim 1, wherein between two successive writes of two successive stripes among the plurality of storage devices, the storage controller is configured

to determine for at least two different storage devices of the plurality of storage devices new distinct values for respective amounts of storage space for user data and corresponding intra-device redundancy data.

7. The computer system as recited in claim 2, wherein each of the plurality of storage devices comprises solid-state memory.

8. The computer system as recited in claim 1, wherein each of the first storage device and the second storage device further include a checksum for stored user data of the unit of data, and wherein in response to a read to at least a portion of the unit of data the storage controller is configured to:

utilize the checksum to validate corresponding read data; and

utilize intra-device redundancy data to rebuild the corresponding read data in response to detecting the checksum indicates the read data is not valid.

9. A method for use in a computing system, the method comprising:

receiving read and write requests at a data storage subsystem, the data storage subsystem comprising a plurality of storage devices;

identifying a unit of data stored in the data storage subsystem, wherein said unit of data is stored across at least a first storage device and a second storage device of the plurality of storage devices, each of the first storage device and the second storage device storing intra-device redundancy data corresponding to the unit of data; and

changing an amount of intra-device redundancy data corresponding to the unit of data on only the first storage device.

10. The method as recited in claim 9, wherein said unit of data is a RAID stripe.

11. The method as recited in claim 9, further comprising changing said amount of intra-device redundancy data stored on the first storage device in response to detecting an indication of a change in reliability of the first storage device.

12. The method as recited in claim 11, further comprising storing data indicative of characteristics of the plurality of storage devices, and wherein said change in reliability is based at least in part on the characteristics.

13. The method as recited in claim 9, further comprising changing said amount of intra-device redundancy data stored on the first storage device in response to detecting a characteristic of the first storage device, the characteristic being one or more of an error rate, an access rate, an age, an occupancy level, an expected life of data, and an age of stored data.

14. The method as recited in claim 9, wherein between two successive writes of two successive stripes among the plurality of storage devices, the method comprises determining for at least two different storage devices of the plurality of storage devices new distinct values for respective amounts of storage space for user data and corresponding intra-device redundancy data.

15. The method as recited in claim 10, wherein each of the plurality of storage devices comprises solid-state memory.

16. The method as recited in claim 9, wherein each of the first storage device and the second storage device further include a checksum for stored user data of the unit of data, and wherein in response to a read to at least a portion of the unit of data the method further comprises:

utilizing the checksum to validate corresponding read data; and

utilizing intra-device redundancy data to rebuild the corresponding read data in response to detecting the checksum indicates the read data is not valid.

17. A computer readable storage medium storing program instructions, wherein said program instructions are executable to:

receive read and write requests at a data storage subsystem, the data storage subsystem comprising a plurality of storage devices;

identify a unit of data stored in the data storage subsystem, wherein said unit of data is stored across at least a first storage device and a second storage device of the

plurality of storage devices, each of the first storage device and the second storage device storing intra-device redundancy data corresponding to the unit of data; and

change an amount of intra-device redundancy data corresponding to the unit of data on only the first storage device.

18. The computer readable storage medium as recited in claim 17, wherein said unit of data is a RAID stripe.

19. The computer readable storage medium as recited in claim 17, wherein said program instructions are further executable to change said amount of intra-device redundancy data stored on the first storage device in response to detecting an indication of a change in reliability of the first storage device.

20. The computer readable storage medium as recited in claim 19, wherein said program instructions are further executable to store data indicative of characteristics of the plurality of storage devices, and wherein said change in reliability is based at least in part on the characteristics.

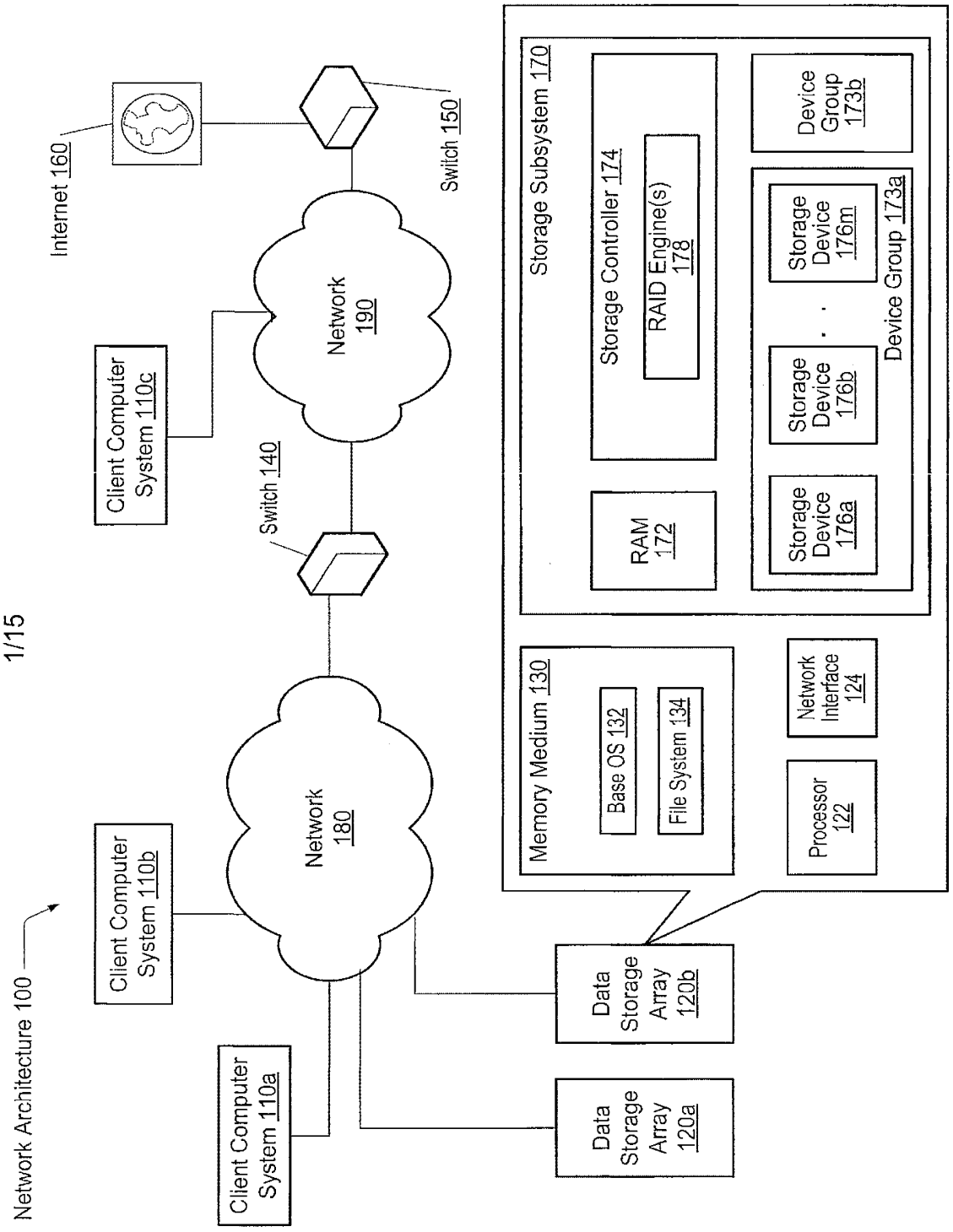

Network Architecture 100
Client Computer System 110a
Client Computer System 110b
Client Computer System 110c
Network 180
Network 190
Switch 140
Switch 150
Internet 160
Data Storage Array 120a
Data Storage Array 120b
Storage Subsystem 170
Memory Medium 130
Base OS 132
File System 134
Processor 122
Network Interface 124
RAM 172
Storage Controller 174
RAID Engine(s) 178
Device Group 173a
Storage Device 176a
Storage Device 176b
Storage Device 176m
Device Group 173b

FIG. 1

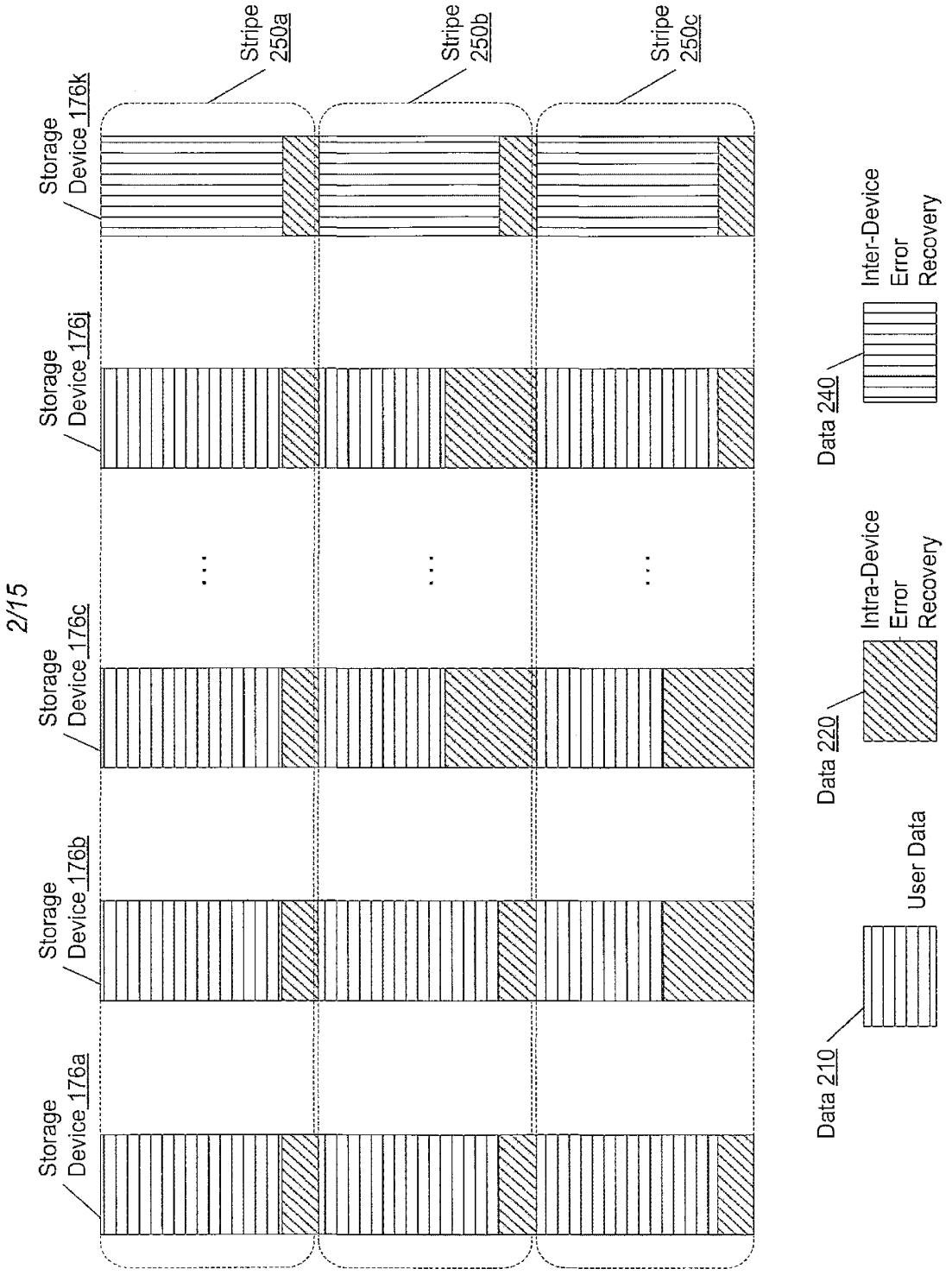

Stripe 250a
Stripe 250b
Stripe 250c
Storage Device 176k
Storage Device 176j
Storage Device 176c
Storage Device 176b
Storage Device 176a
Data 240
Inter-Device Error Recovery
Data 220
Intra-Device Error Recovery
Data 210
User Data

FIG. 2

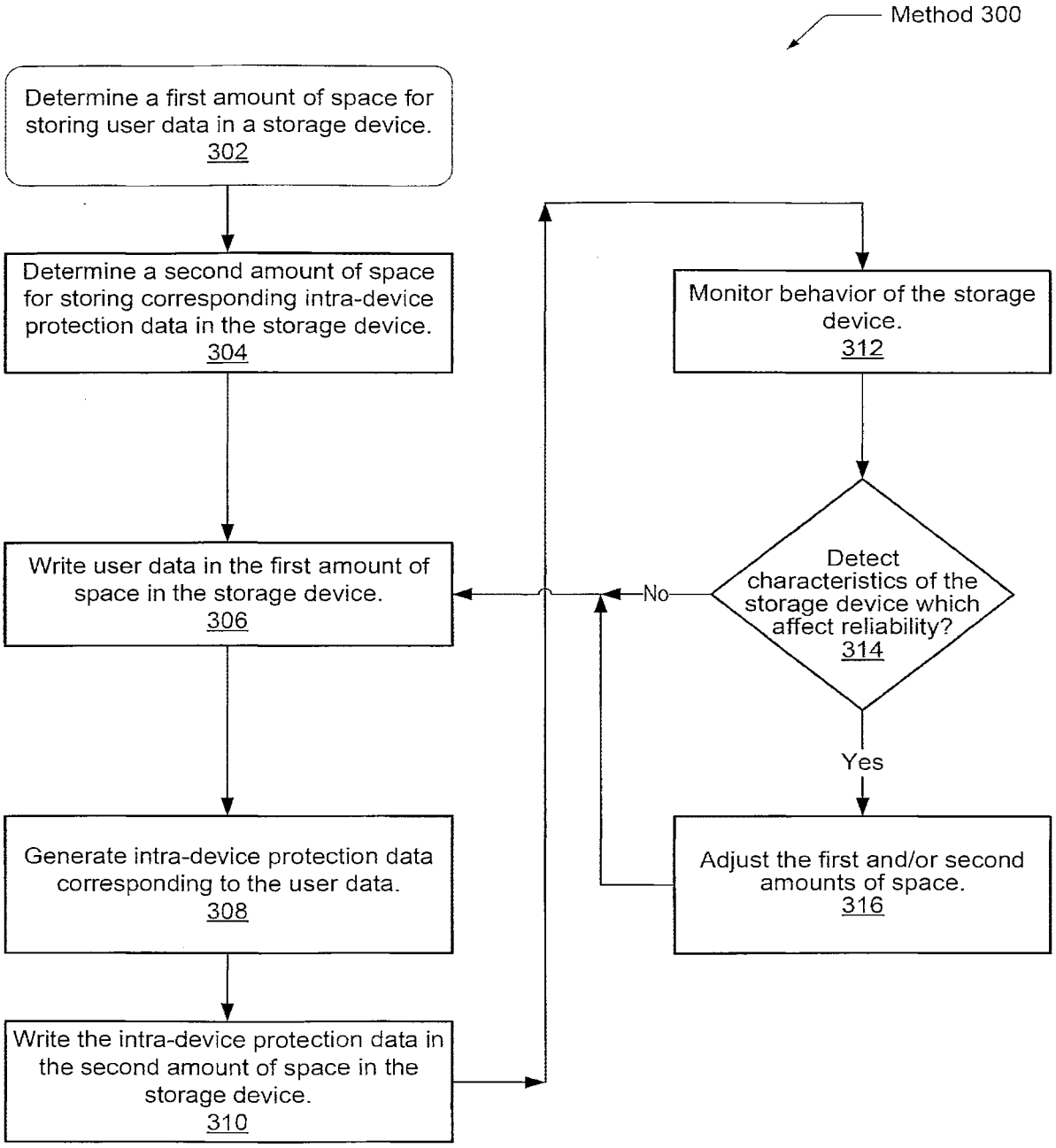

Method 300
Determine a first amount of space for storing user data in a storage device.
302
Determine a second amount of space for storing corresponding intra-device protection data in the storage device.
304
Write user data in the first amount of space in the storage device.
306
Generate intra-device protection data corresponding to the user data.
308
Write the intra-device protection data in the second amount of space in the storage device.
310
Monitor behavior of the storage device.
312
Detect characteristics of the storage device which affect reliability?
314
No
Yes
Adjust the first and/or second amounts of space.
316

FIG. 3

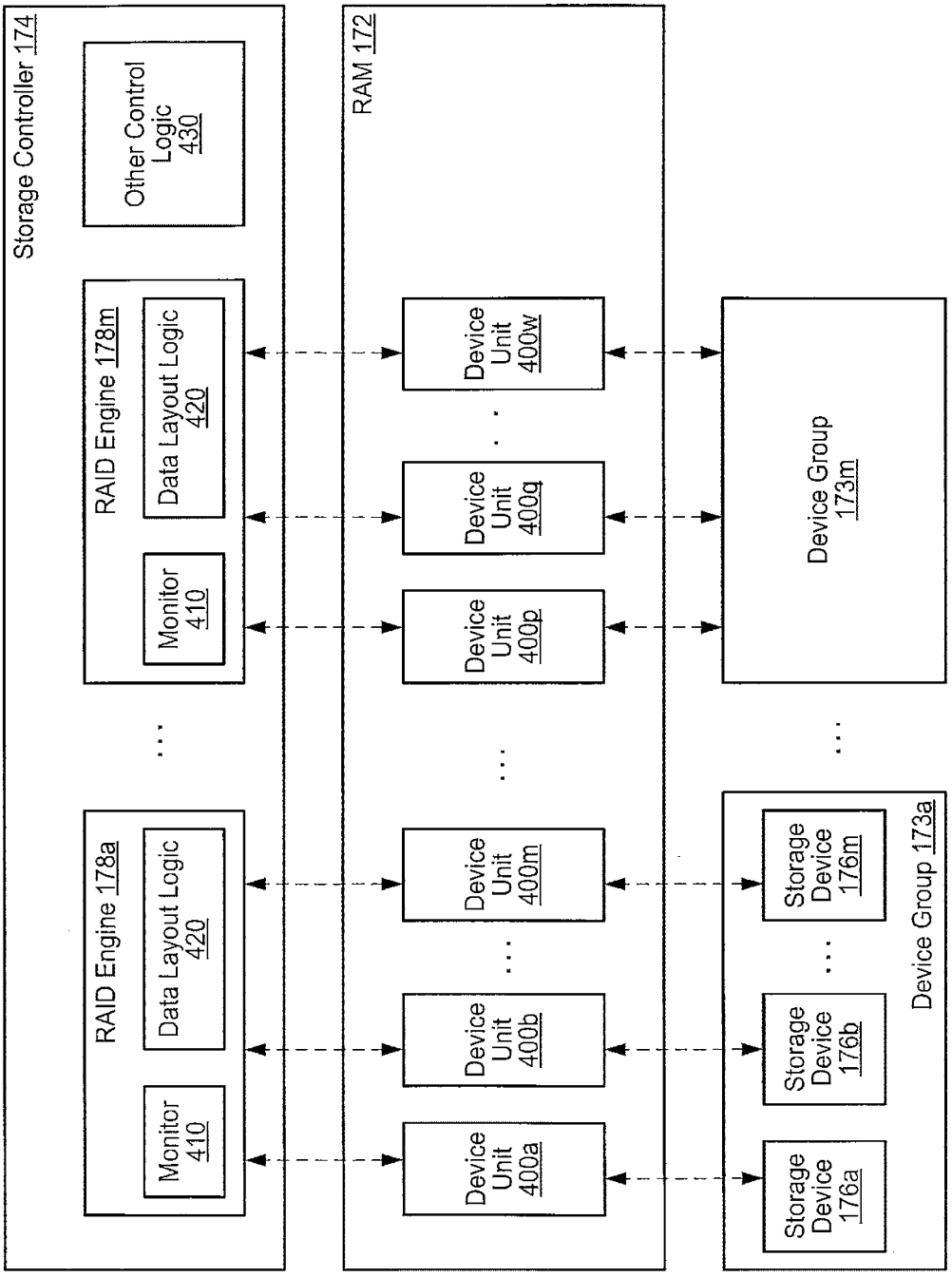

Storage Controller 174
Other Control Logic 430
RAID Engine 178m
Data Layout Logic 420
Monitor 410
RAID Engine 178a
Data Layout Logic 420
Monitor 410
RAM 172
Device Unit 400a
Device Unit 400b
Device Unit 400m
Device Unit 400p
Device Unit 400q
Device Unit 400w
Device Group 173m
Device Group 173a
Storage Device 176a
Storage Device 176b
Storage Device 176m

FIG. 4

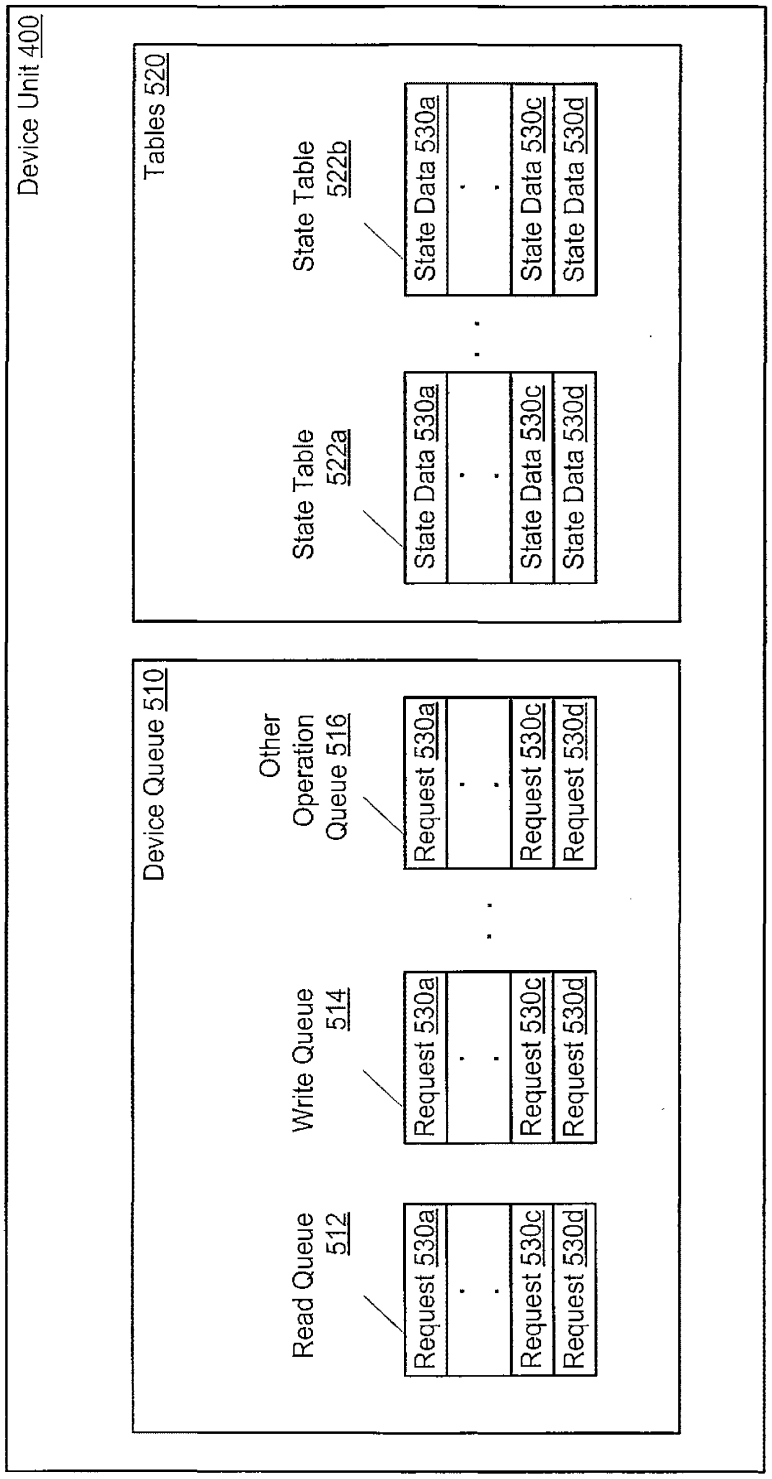

Device Unit 400
Tables 520
State Table 522a
State Table 522b
State Data 530a
State Data 530c
State Data 530d
Device Queue 510
Read Queue 512
Write Queue 514
Other Operation Queue 516
Request 530a
Request 530c
Request 530d

FIG. 5

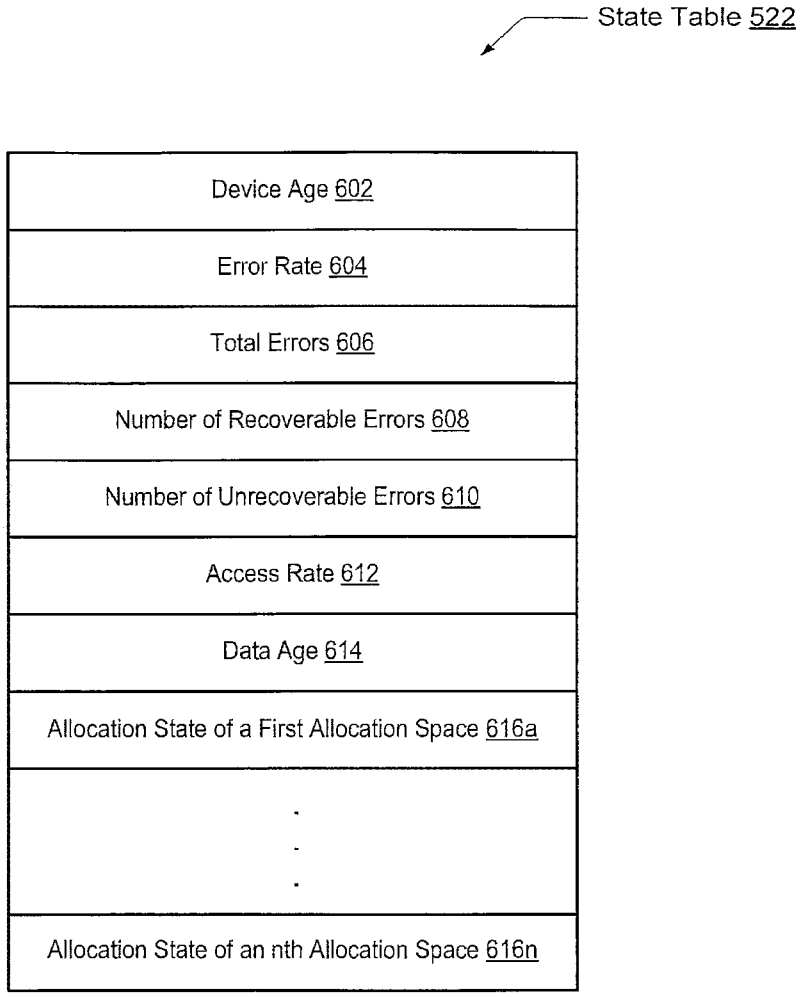

State Table 522
Device Age 602
Error Rate 604
Total Errors 606
Number of Recoverable Errors 608
Number of Unrecoverable Errors 610
Access Rate 612
Data Age 614
Allocation State of a First Allocation Space 616a
Allocation State of an nth Allocation Space 616n

FIG. 6

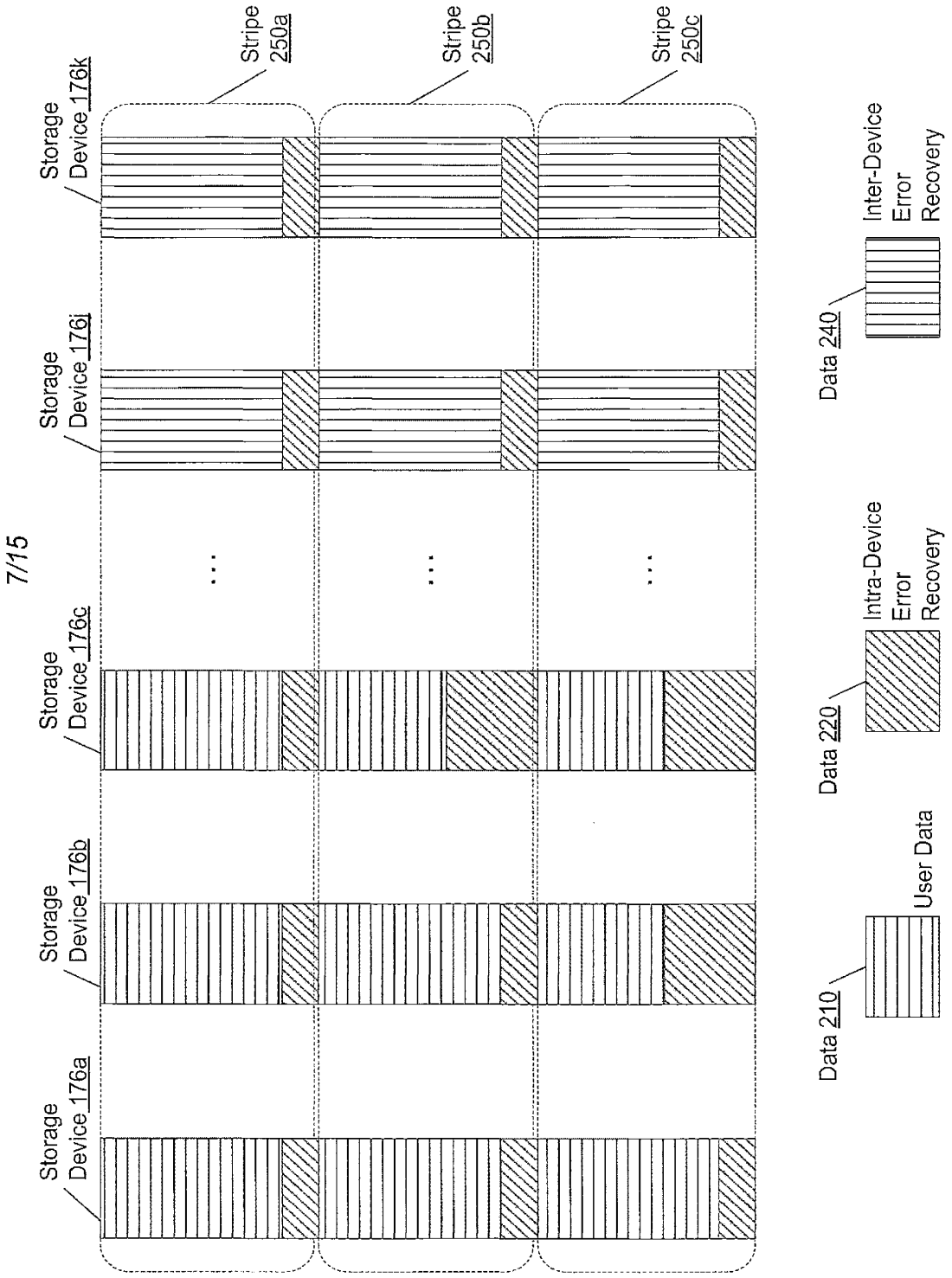
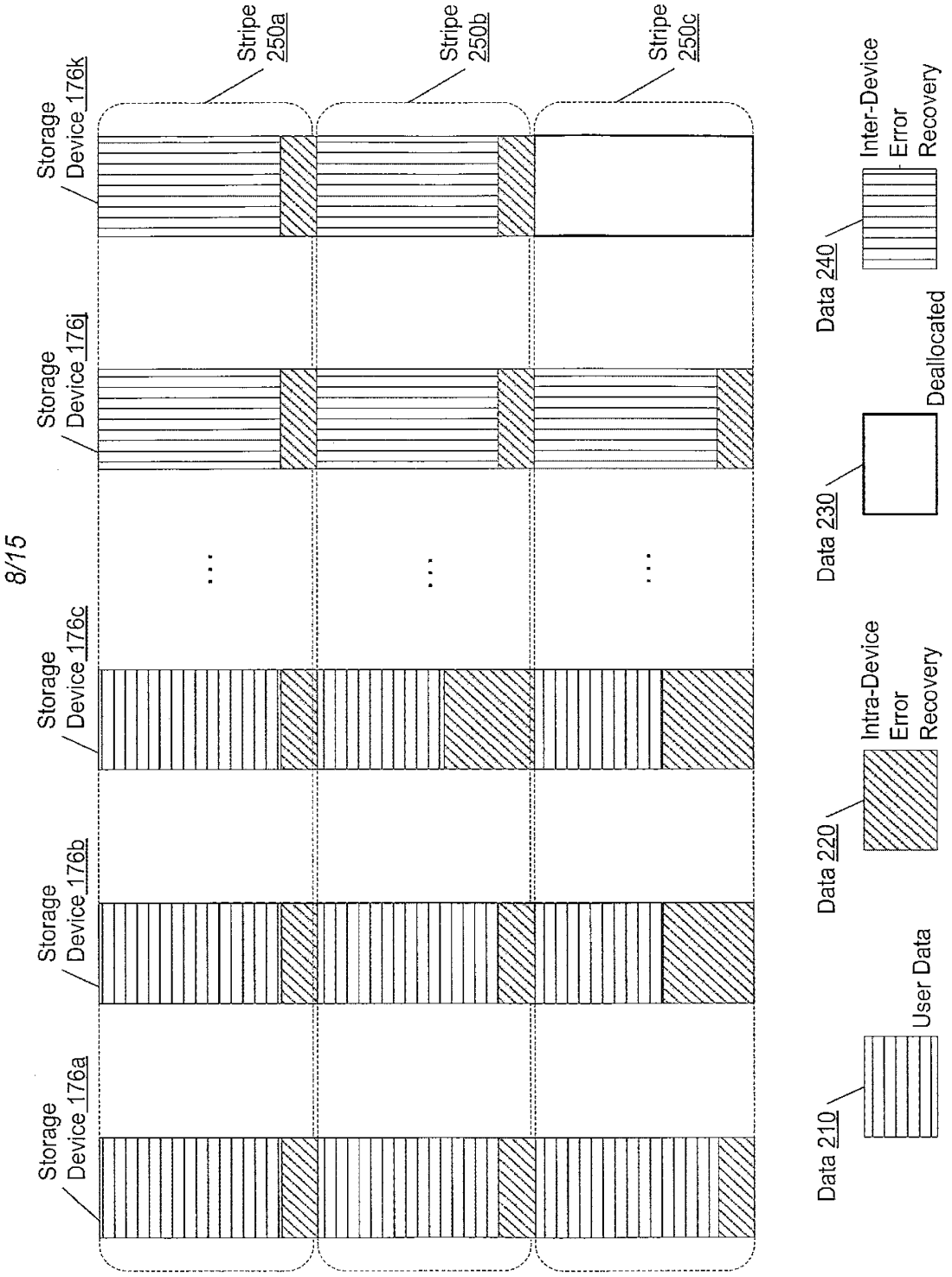
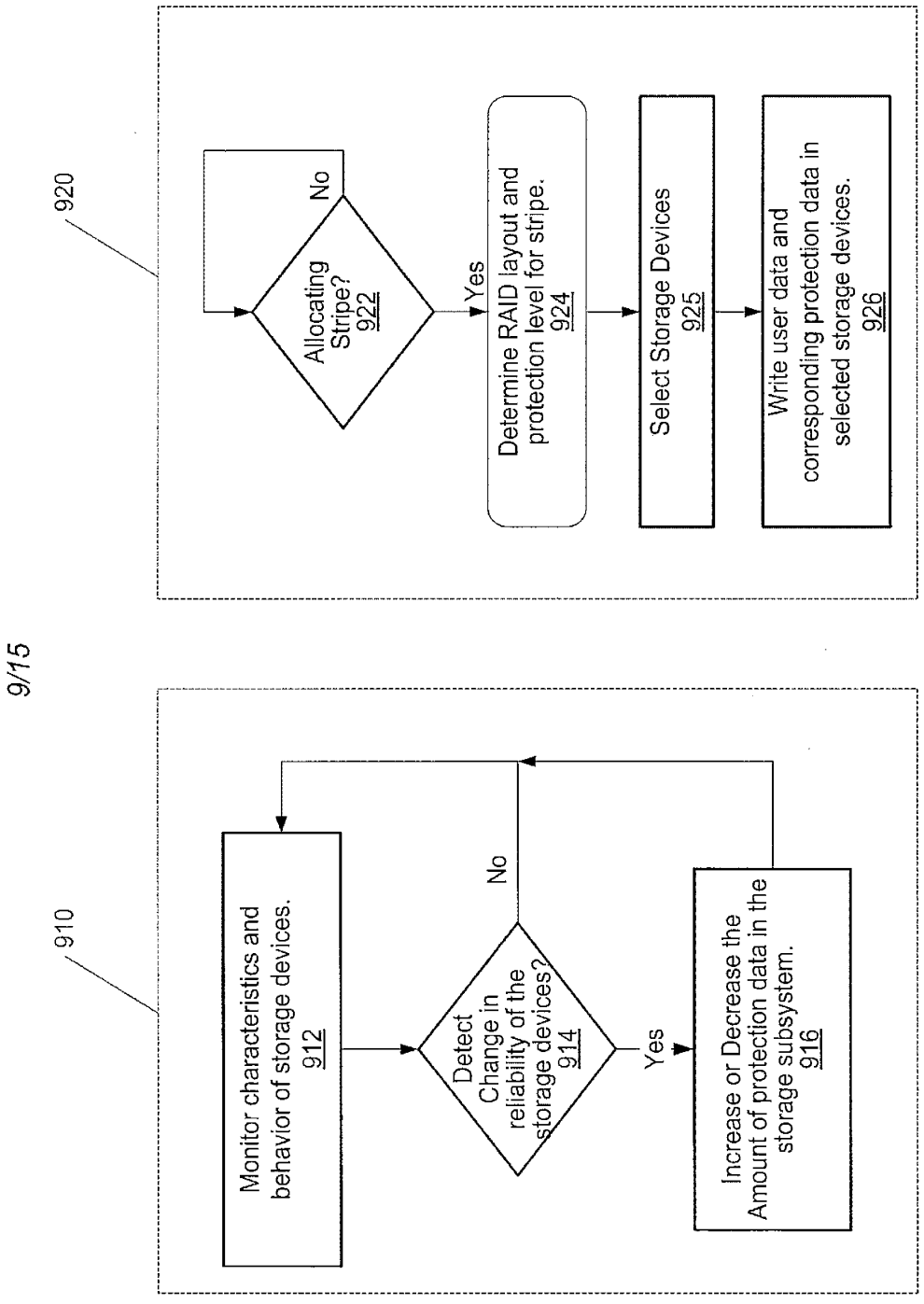

Stripe 250a
Stripe 250b
Stripe 250c
Storage Device 176k
Storage Device 176j
Storage Device 176c
Storage Device 176b
Storage Device 176a
Data 240
Inter-Device Error Recovery
Data 220
Intra-Device Error Recovery
Data 210
User Data

FIG. 7


Stripe 250a
Stripe 250b
Stripe 250c
Storage Device 176k
Storage Device 176j
Storage Device 176c
Storage Device 176b
Storage Device 176a
Data 240
Inter-Device Error Recovery
Data 230
Deallocated
Data 220
Intra-Device Error Recovery
Data 210
User Data

8/15

FIG. 8


910
Monitor characteristics and behavior of storage devices. 912
Detect Change in reliability of the storage devices? 914
No
Yes
Increase or Decrease the Amount of protection data in the storage subsystem. 916
920
Allocating Stripe? 922
No
Yes
Determine RAID layout and protection level for stripe. 924
Select Storage Devices 925
Write user data and corresponding protection data in selected storage devices. 926

9/15

FIG. 9

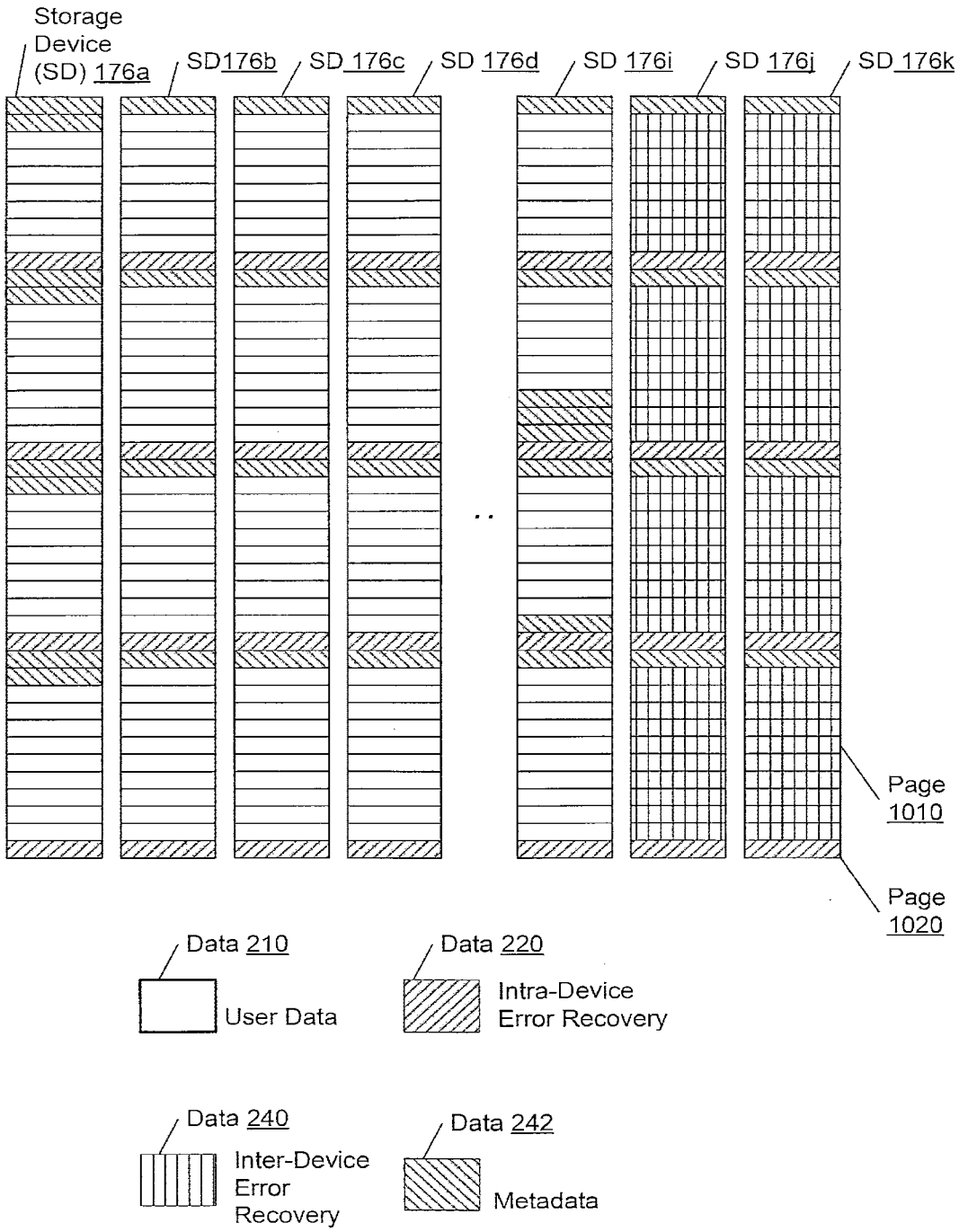

Storage Device (SD) 176a
SD176b
SD 176c
SD 176d
SD 176i
SD 176j
SD 176k
Page 1010
Page 1020
Data 210
User Data
Data 220
Intra-Device Error Recovery
Data 240
Inter-Device Error Recovery
Data 242
Metadata

FIG. 10

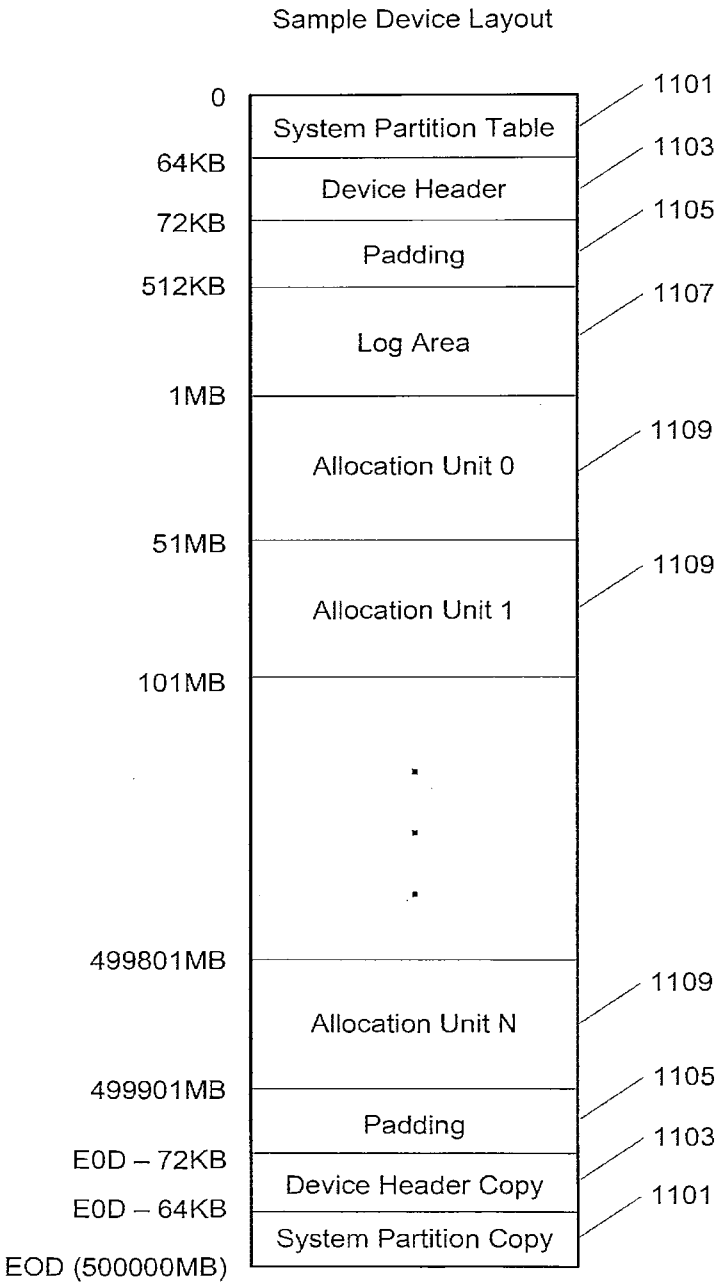
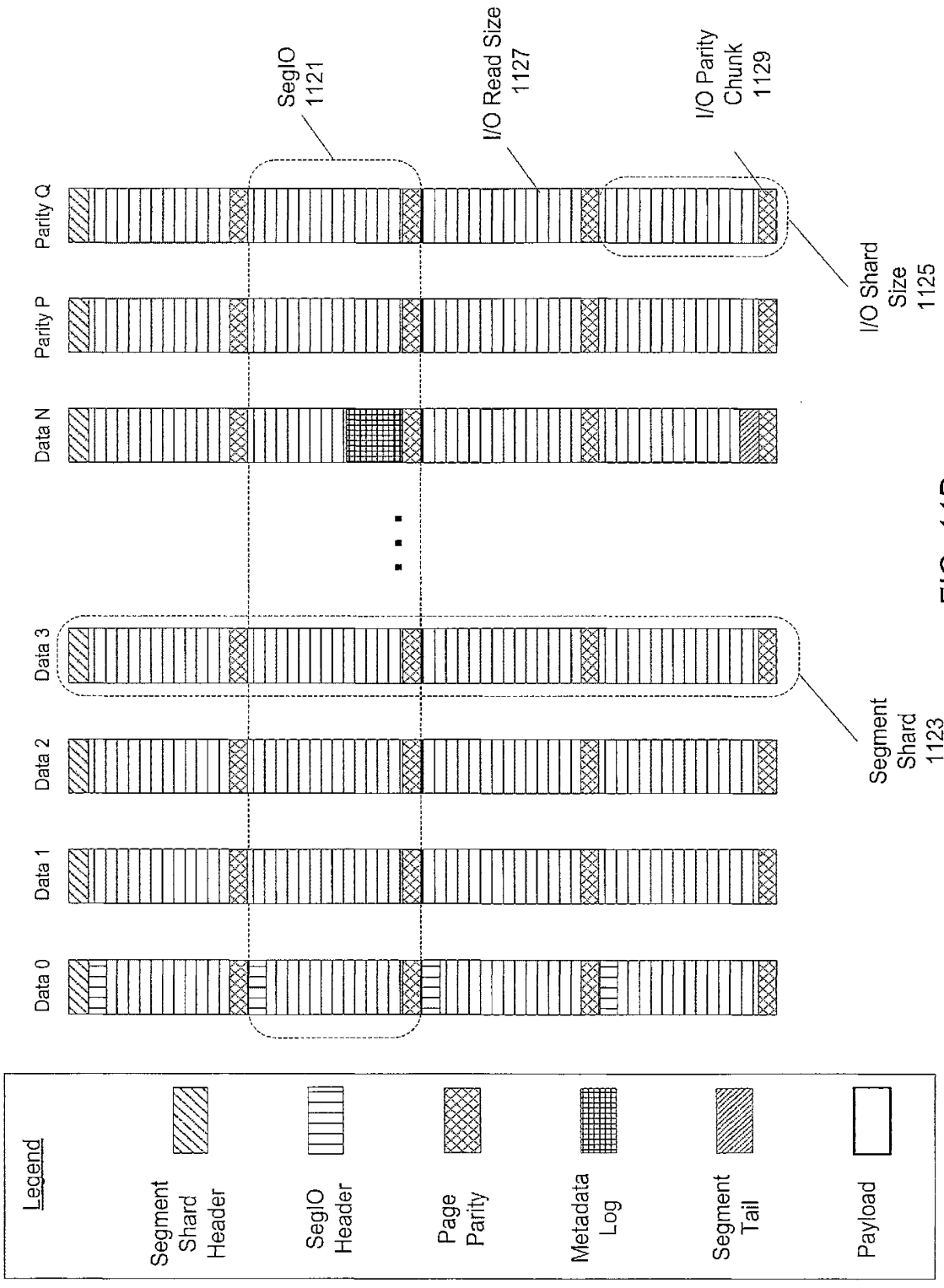

Sample Device Layout
0
64KB
72KB
512KB
1MB
51MB
101MB
499801MB
499901MB
E0D – 72KB
E0D – 64KB
EOD (500000MB)
System Partition Table
Device Header
Padding
Log Area
Allocation Unit 0
Allocation Unit 1
Allocation Unit N
Padding
Device Header Copy
System Partition Copy
1101
1103
1105
1107
1109
1109
1109
1105
1103
1101

FIG. 11A


SegIO 1121
I/O Read Size 1127
I/O Parity Chunk 1129
I/O Shard Size 1125
Segment Shard 1123
Parity Q
Parity P
Data N
Data 3
Data 2
Data 1
Data 0
Legend
Segment Shard Header
SegIO Header
Page Parity
Metadata Log
Segment Tail
Payload

12/15

FIG. 11B

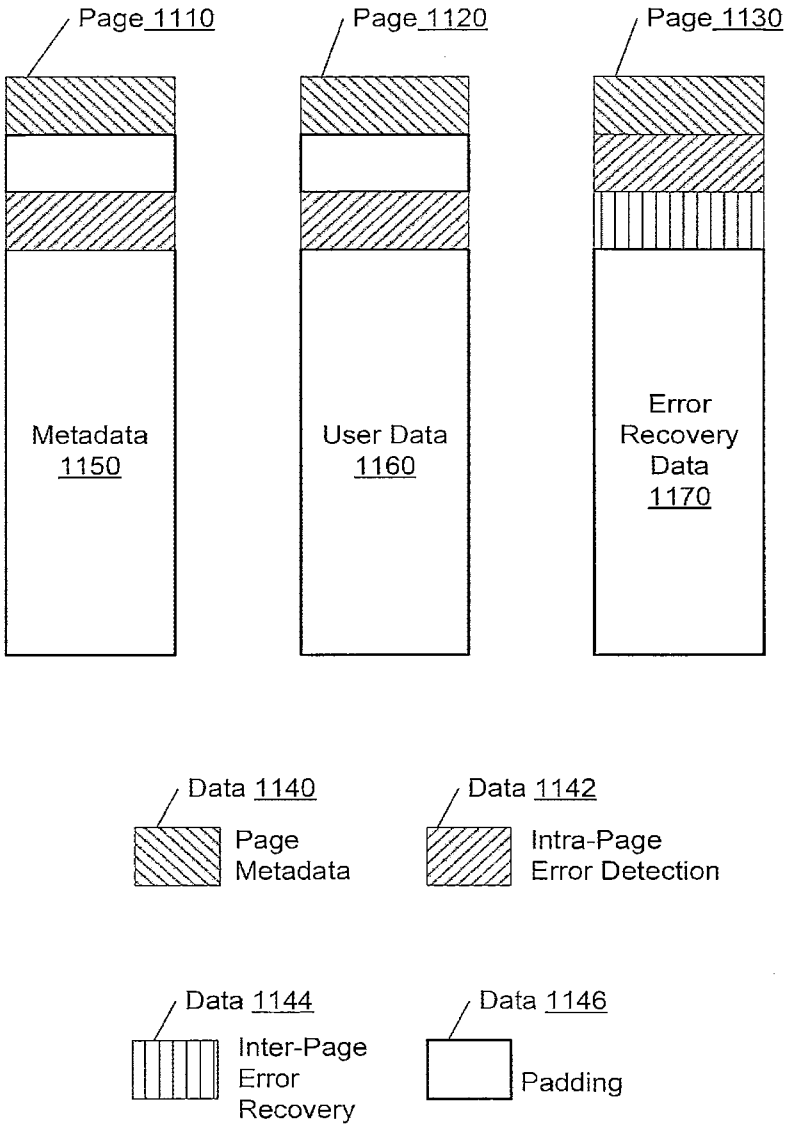

Page 1110
Page 1120
Page 1130
Metadata 1150
User Data 1160
Error Recovery Data 1170
Data 1140
Page Metadata
Data 1142
Intra-Page Error Detection
Data 1144
Inter-Page Error Recovery
Data 1146
Padding

FIG. 11C

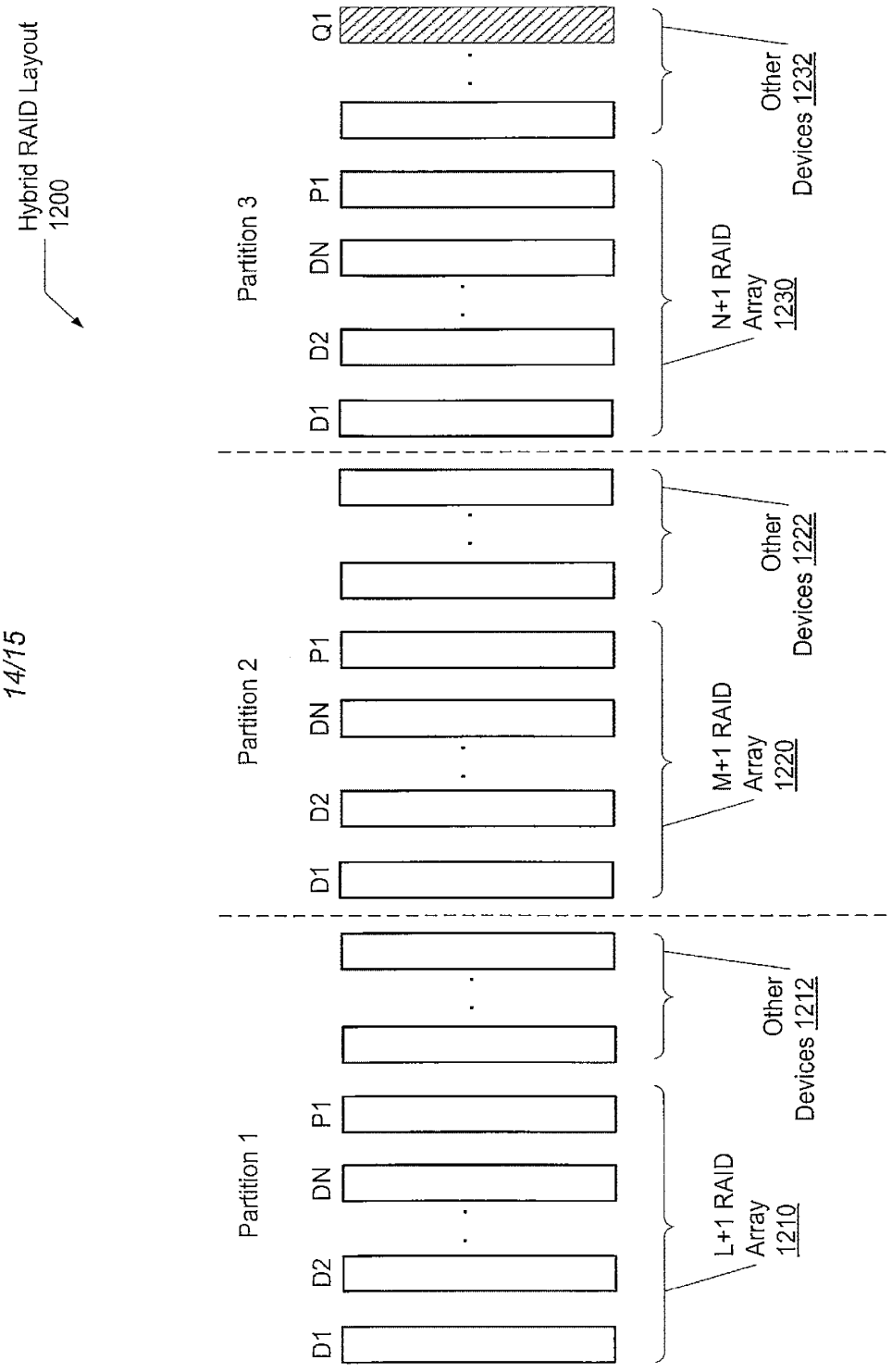

Hybrid RAID Layout
1200
Partition 1
D1
D2
DN
P1
L+1 RAID
Array
1210
Other
Devices 1212
Partition 2
D1
D2
DN
P1
M+1 RAID
Array
1220
Other
Devices 1222
Partition 3
D1
D2
DN
P1
Q1
N+1 RAID
Array
1230
Other
Devices 1232

FIG. 12

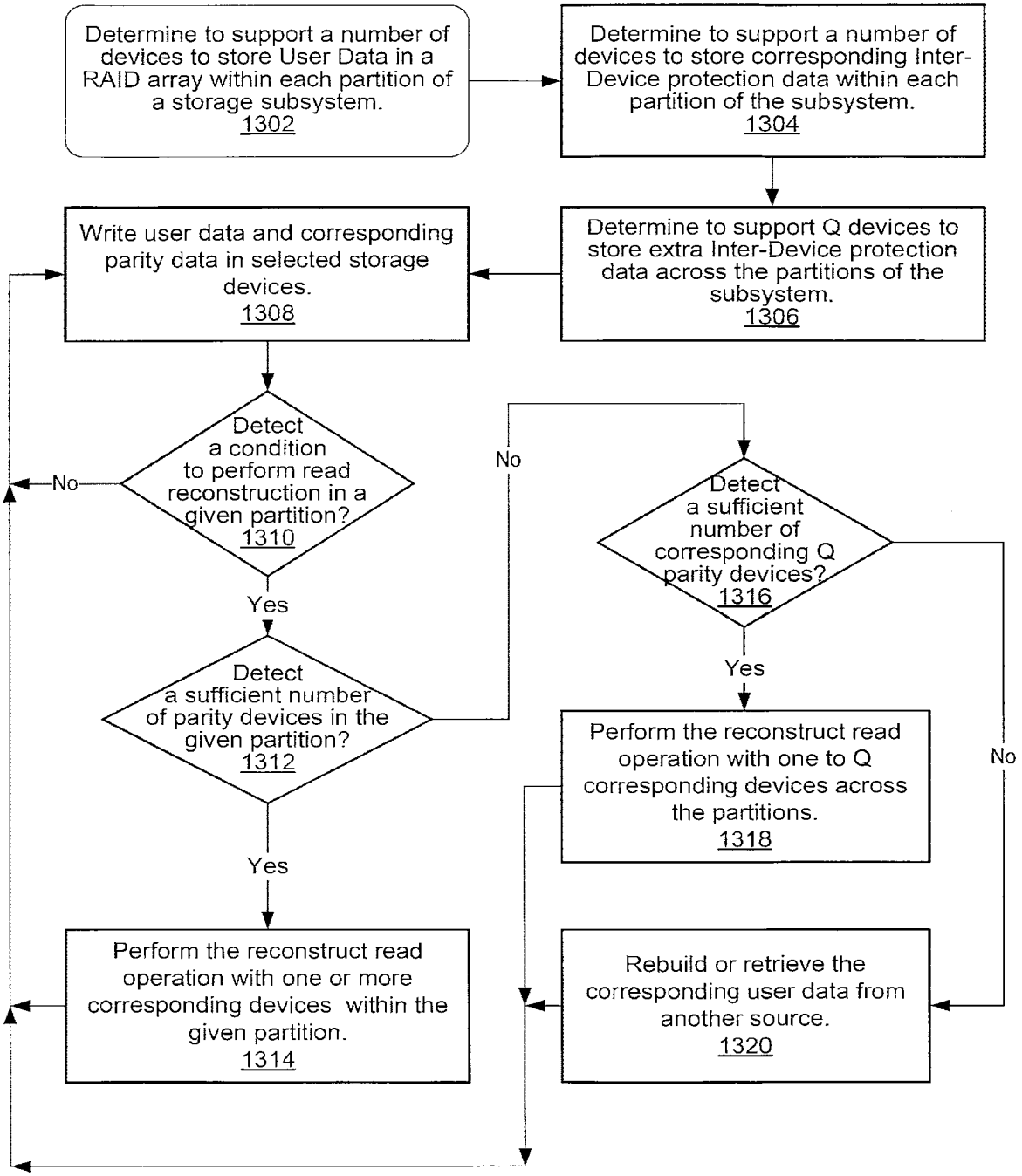

Determine to support a number of devices to store User Data in a RAID array within each partition of a storage subsystem. 1302
Determine to support a number of devices to store corresponding Inter-Device protection data within each partition of the subsystem. 1304
Determine to support Q devices to store extra Inter-Device protection data across the partitions of the subsystem. 1306
Write user data and corresponding parity data in selected storage devices. 1308
Detect a condition to perform read reconstruction in a given partition? 1310
No
Yes
Detect a sufficient number of parity devices in the given partition? 1312
No
Yes
Detect a sufficient number of corresponding Q parity devices? 1316
Yes
No
Perform the reconstruct read operation with one or more corresponding devices within the given partition. 1314
Perform the reconstruct read operation with one to Q corresponding devices across the partitions. 1318
Rebuild or retrieve the corresponding user data from another source. 1320

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/052268

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/10 G06F3/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, IBM-TDB, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2005/108594 A1 (MENON JAISHANKAR M [US] ET AL) 19 May 2005 (2005-05-19) the whole document	1-20
Y	ALINA OPREA , ARI JUELS: "A clean-slate look at disk scrubbing", PROCEEDINGS OF THE 8TH USENIX CONFERENCE ON FILE AND STORAGE TECHNOLOGIES, 23 February 2010 (2010-02-23), - 26 February 2010 (2010-02-26), XP002665466, San Jose, California the whole document	1-20

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 December 2011	30/12/2011

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Salsa, Francesco

2

Form PCT/ISA/210 (second sheet) (April 2005)

Ex.1002 / Page 362 of 744
Sandisk Technologies, Inc. et al

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	MARK W. STORER , KEVIN M. GREENAN , ETHAN L. MILLER , KALADHAR VORUGANTI: "Pergamum: replacing tape with energy efficient, reliable, disk-based archival storage", 6TH USENIX CONFERENCE ON FILE AND STORAGE TECHNOLOGIES (FAST '08) 26-29 FEB. 2008 SAN JOSE, CA, USA, 2008, pages 1-16, XP002665467, 6th USENIX Conference on File and Storage Technologies (FAST '08) USENIX Berkeley, CA, USA ISBN: 978-1-931971-56-0 paragraph [0001] paragraph [0004] - paragraph [04.2] figures 2,3	8,16
A	AVANI WILDANI ET AL: "Protecting against rare event failures in archival systems", MODELING, ANALYSIS&SIMULATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS, 2009. MASCOTS '09. IEEE INTERNATIONAL SYMPOSIUM ON, IEEE, PISCATAWAY, NJ, USA, 21 September 2009 (2009-09-21), pages 1-11, XP031591252, ISBN: 978-1-4244-4927-9 the whole document	1-20
A	ILIAS ILIADIS ET AL: "Disk scrubbing versus intra-disk redundancy for high-reliability raid storage systems", PROCEEDINGS OF THE 2007 ACM SIGMETRICS INTERNATIONAL CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS , SIGMETRICS '07, vol. 36, 1 January 2008 (2008-01-01), page 241, XP55014591, New York, New York, USA ISSN: 0163-5999, DOI: 10.1145/1375457.1375485 ISBN: 978-1-59-593639-4 abstract paragraphs [0001], [0010]	1-20

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2011/052268

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005108594 A1	19-05-2005	NONE	

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 June 2012 (07.06.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/075200 A2

(51) International Patent Classification:
G06F 12/16 (2006.01) *G06F 11/08* (2006.01)

(21) International Application Number:
PCT/US2011/062726

(22) International Filing Date:
30 November 2011 (30.11.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/418,846 1 December 2010 (01.12.2010) US
61/433,918 18 January 2011 (18.01.2011) US

(71) Applicant *(for all designated States except US)*: **SANDFORCE, INC.** [US/US]; 691 S. Milpitas Blvd., Ste 100, Milpitas, California 95035 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **WERNER, Jeremy** [US/US]; 5579 Harvard Dr, San Jose, California 95118 (US). **BARYUDIN, Leonid** [US/US]; 5067 Leigh Avenue, San Jose, California 95124 (US). **CANEPA, Timothy** [US/US]; 418 University Ave, Los Gatos, California 95032 (US). **COHEN, Earl** [US/US]; 78 Hermosa Ave, Oakland, California 94618 (US).

(74) Agent: **SMITH, Walstein**; P.O. Box 1668, Georgetown, Texas 78627-1668 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) **Title**: DYNAMIC HIGHER-LEVEL REDUNDANCY MODE MANAGEMENT WITH INDEPENDENT SILICON ELEMENTS

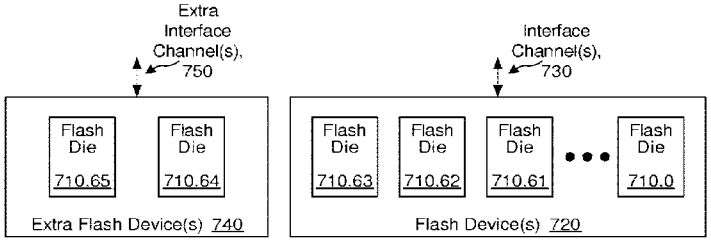


Fig. 7A

(57) **Abstract**: A Solid-State Disk (SSD) controller enables dynamic higher-level redundancy mode management with independent silicon elements to provide graceful degradation as non-volatile (e.g. flash) memory elements fail during operation of an SSD implemented by the controller. Higher-level error correction provides correction of lower-level uncorrectable errors. If a failure of one of the non-volatile memory elements is detected, then the higher-level error correction is dynamically transitioned from operating in a current mode to operating in a new mode. The transition includes one or more of reducing free space available on the SSD, rearranging data storage of the SSD, recovering/storing failed user data (if possible), and determining/storing revised higher-level error correction information. Operation then continues in the new mode. If another failure of the non-volatile memory elements is detected, then another transition is made to another new mode.

WO 2012/075200 A2

DYNAMIC HIGHER-LEVEL REDUNDANCY MODE MANAGEMENT WITH INDEPENDENT SILICON ELEMENTS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] Priority benefit claims for this application are made in the accompanying Application Data Sheet, Request, or Transmittal (as appropriate, if any). To the extent permitted by the type of the instant application, this application incorporates by reference for all purposes the following applications, all commonly owned with the instant application at the time the invention was made:

U.S. Provisional Application (Docket No. SF-10-10 and Serial No. 61/418,846), filed 12-01-2010, first named inventor Jeremy Isaac Nathaniel Werner, and entitled **Dynamic Higher-Level Redundancy Mode Management with Independent Silicon Elements**;

U.S. Provisional Application (Docket No. SF-10-14 and Serial No. 61/433,918), filed 01-18-2011, first named inventor Jeremy Isaac Nathaniel Werner, and entitled **Higher-Level Redundancy Information Computation**;

PCT Application (Docket No. SF-10-01PCTB and Serial No. PCT/US11/28244), filed 03-11-2011, first named inventor Hao Zhong, and entitled **LDPC Erasure Decoding for Flash Memories**; and

PCT Application (Docket No. SF-10-03PCT and Serial No. PCT/US11/57914), filed 10-26-2011, first named inventor Yan Li, and entitled **Adaptive ECC Techniques for Flash Memory Based Data Storage**.

BACKGROUND

[0002] Field: Advancements in flash memory storage technology are needed to provide improvements in performance, efficiency, and utility of use.

[0003] Related Art: Unless expressly identified as being publicly or well known, mention herein of techniques and concepts, including for context, definitions, or comparison purposes, should not be construed as an admission that such techniques and concepts are previously publicly known or otherwise part of the prior art. All references cited herein (if any),

including patents, patent applications, and publications, are hereby incorporated by reference in their entireties, whether specifically incorporated or not, for all purposes.

SYNOPSIS

[0004] The invention may be implemented in numerous ways, e.g. as a process, an article of manufacture, an apparatus, a system, a composition of matter, and a computer readable medium such as a computer readable storage medium (e.g., media in an optical and/or magnetic mass storage device such as a disk, an integrated circuit having non-volatile storage such as flash storage), or a computer network wherein program instructions are sent over optical or electronic communication links. The Detailed Description provides an exposition of one or more embodiments of the invention that enable improvements in cost, profitability, performance, efficiency, and utility of use in the field identified above. The Detailed Description includes an Introduction to facilitate understanding of the remainder of the Detailed Description. The Introduction includes Example Embodiments of one or more of systems, methods, articles of manufacture, and computer readable media in accordance with concepts described herein. As is discussed in more detail in the Conclusions, the invention encompasses all possible modifications and variations within the scope of the issued claims.

Brief Description of Drawings

[0005] Fig. 1A illustrates selected details of an embodiment of a Solid-State Disk (SSD) including an SSD controller using dynamic higher-level redundancy mode management with independent silicon elements for accessing and controlling Non-Volatile Memories (NVMs).

[0006] Fig. 1B illustrates selected details of various embodiments of systems including one or more instances of the SSD of Fig. 1A.

[0007] Fig. 2 illustrates selected details of an embodiment of mapping a Logical Page Number (LPN) portion of a Logical Block Address (LBA).

[0008] Fig. 3 illustrates selected details of an embodiment of accessing a Non-Volatile Memory (NVM) at a read unit address to produce read data organized as various read units, collectively having a length measured in quanta of read units.

[0009] Fig. 4A illustrates selected details of an embodiment of a read unit.

[0010] Fig. 4B illustrates selected details of another embodiment of a read unit.

[0011] Fig. 5 illustrates selected details of an embodiment of a header having a number of fields.

[0012] Fig. 6 illustrates selected details of an embodiment of blocks, pages, and read units of multiple NVM devices.

[0013] Fig. 7A illustrates selected details of various embodiments of higher-level redundancy techniques.

[0014] Fig. 7B illustrates selected details of an embodiment of R-blocks.

[0015] Fig. 8 illustrates selected details of an embodiment of dynamic higher-level redundancy mode management with a Redundant Array of Silicon Independent Elements (RASIE).

[0016] Fig. 9 illustrates an embodiment of read units having lower-level redundancy information of adaptive code rates protected by higher-level redundancy information stored in one or more of the read units.

[0017] Fig. 10 illustrates selected details of an embodiment of higher-level redundancy information result and data source correspondences.

[0018] Fig. 11 illustrates selected details of an embodiment of higher-level redundancy information computations.

[0019] Fig. 12 illustrates selected details of an embodiment of recovery from one (lower-level) failure (during a single operation).

[0020] Figs. 13A-13D illustrate selected details of an embodiment of recovery from two (lower-level) failures (during a single operation).

[0021] Figs. 14A and 14B illustrate selected details of an embodiment of computing higher-level redundancy information with respect to pages received from NVMs.

[0022] Figs. 15A-15C illustrate selected details of an embodiment of backing out of a computation of higher-level redundancy information with respect to a write provided to NVMs.

List of Reference Symbols in Drawings

[0023]

Ref. Symbol	Element Name
100	SSD Controller
101	SSD
102	Host
103	(optional) Switch / Fabric / Intermediate Controller
104	Intermediate Interfaces
105	OS
106	FirmWare (FW)
107	Driver
107D	dotted-arrow (Host Software ←→ I/O Device Communication)
109	Application
109D	dotted-arrow (Application ←→ I/O Device Communication via driver)
109V	dotted-arrow (Application ←→ I/O Device Communication via VF)
110	External Interfaces
111	Host Interfaces
112C	(optional) Card Memory
113	Tag Tracking
114	Multi-Device Management Software
115	Host Software
116	I/O Card
117	I/O & Storage Devices/Resources
118	Servers
119	LAN/WAN
121	Data Processing
123	Engines
125	Control RAM
126	Scaled Data Structures
131	Buffer
133	DMA
135	ECC-X
137	Memory

Ref. Symbol	Element Name
141	Map
143	Table
151	Recycler
161	ECC
171	CPU
172	CPU Core
173	Command Management
175	Buffer Management
177	Translation Management
179	Coherency Management
180	Memory Interface
181	Device Management
182	Identity Management
190	Device Interfaces
191	Device Interface Logic
192	Flash Device
193	Scheduling
194	Flash Die
199	NVM
211	LBA
213	LPN
215	Logical Offset
221	Map Info for LPN
223	Read Unit Address
225	Length in Read Units
311	Read Data
313	First Read Unit
315	Last Read Unit
401A	Read Unit
401B	Read Unit
410B	Header Marker (HM)
411A	Header 1
411B	Header 1
412B	Header 2

Ref. Symbol	Element Name
419A	Header N
419B	Header N
421A	Data Bytes
421B	Data Bytes
422B	Data Bytes
429B	Data Bytes
431A	Optional Padding Bytes
431B	Optional Padding Bytes
501	Header
511	Type
513	Last Indicator
515	Flags
517	LPN
519	Length
521	Offset
600	Striping Direction
601, 603, 609	Devices
611, 613, 619, 641, 643, 649, 671, 673, 679	Read Units
621, 623, 629, 651, 653, 659, 681, 683, 689	Pages
631, 633, 639, 661, 663, 669, 691, 693, 699	Blocks
710.0, 710.0B0, 710.0B1, 710.0BB	Blocks
710.0P0, 710.0P1, 710.0PP	Pages
710.61,	Flash Dic

Ref. Symbol	Element Name
710.62, 710.63	
710.63B0, 710.63B1, 710.63BB	Blocks
710.63P0, 710.63P1, 710.63PP	Pages
710.64	Flash Die
710.64B0, 710.64B1, 710.64BB	Blocks
710.64P0, 710.64P1, 710.64PP	Pages
710.65	Flash Die
710.65B0, 710.65B1, 710.65BB	Blocks
710.65P0, 710.65P1, 710.65PP	Pages
720	Flash Device(s)
730	Interface Channel(s)
740	Extra Flash Device(s)
750	Extra Interface Channel(s)
760.0, 760.1, 760.R	R-blocks
801	Start
802	Operate in First Higher-Level Redundancy Mode
803	Failure?
804	Reduce Free Space
805	Rearrange Data Storage
806	Recover/Store Failed User Data

Ref. Symbol	Element Name
807	Determine/Store Revised Higher-Level Redundancy Information
808	Operate in Second Higher-Level Redundancy Mode
899	End
911, 931, 951, 971	Read Unit
911.E, 931.E, 951.E, 971.E	Lower-Level ECC
911.U, 931.U, 951.U, 971.U	User Data
1001	Result
1002	Result
1003	Result
1010	R0
1011	R1
1019	Data
1401A	Start
1401B	Start
1402A	Issue Read Operation
1402B	Page Ready?
1403A	All Issued?
1403B	Perform Page-Based Computations
1404B	Pages Finished?
1499A	End
1499B	End
1501A	Start
1501B	Start
1502A	Issue Write Operation
1502B	Write OK?
1503A	All Issued?
1503B	Backout Write from Redundancy
1504B	Writes Finished?
1599A	End
1599B	End

Ex.1002 / Page 374 of 744
Sandisk Technologies, Inc. et al

DETAILED DESCRIPTION

[0024] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures illustrating selected details of the invention. The invention is described in connection with the embodiments. The embodiments herein are understood to be merely exemplary, the invention is expressly not limited to or by any or all of the embodiments herein, and the invention encompasses numerous alternatives, modifications, and equivalents. To avoid monotony in the exposition, a variety of word labels (including but not limited to: first, last, certain, various, further, other, particular, select, some, and notable) may be applied to separate sets of embodiments; as used herein such labels are expressly not meant to convey quality, or any form of preference or prejudice, but merely to conveniently distinguish among the separate sets. The order of some operations of disclosed processes is alterable within the scope of the invention. Wherever multiple embodiments serve to describe variations in process, method, and/or program instruction features, other embodiments are contemplated that in accordance with a predetermined or a dynamically determined criterion perform static and/or dynamic selection of one of a plurality of modes of operation corresponding respectively to a plurality of the multiple embodiments. Numerous specific details are set forth in the following description to provide a thorough understanding of the invention. The details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of the details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

INTRODUCTION

[0025] This introduction is included only to facilitate the more rapid understanding of the Detailed Description; the invention is not limited to the concepts presented in the introduction (including explicit examples, if any), as the paragraphs of any introduction are necessarily an abridged view of the entire subject and are not meant to be an exhaustive or restrictive description. For example, the introduction that follows provides overview information limited by space and organization to only certain embodiments. There are many other embodiments, including those to which claims will ultimately be drawn, discussed throughout the balance of the specification.

Acronyms

[0026] At least some of the various shorthand abbreviations (e.g. acronyms) defined here refer to certain elements used herein.

Acronym	Description
AHCI	Advanced Host Controller Interface
API	Application Program Interface
ATA	Advanced Technology Attachment (AT Attachment)
BCH	Bose Chaudhuri Hocquenghem
CD	Compact Disk
CF	Compact Flash
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DAS	Direct Attached Storage
DDR	Double-Data-Rate
DMA	Direct Memory Access
DNA	Direct NAND Access
DRAM	Dynamic Random Access Memory
DVD	Digital Versatile/Video Disk
DVR	Digital Video Recorder
ECC	Error-Correcting Code
eMMC	Embedded MultiMediaCard
eSATA	external Serial Advanced Technology Attachment
GPS	Global Positioning System
HDD	Hard Disk Drive
I/O	Input / Output
IC	Integrated Circuit
IDE	Integrated Drive Electronics
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LBA	Logical Block Address
LDPC	Low-Density Parity-Check
LPN	Logical Page Number

Acronym	Description
MLC	Multi-Level Cell
MMC	MultiMediaCard
MPEG	Moving Picture Experts Group
NAS	Network Attached Storage
NCQ	Native Command Queuing
NVM	Non-Volatile Memory
ONA	Optimized NAND Access
ONFI	Open NAND Flash Interface
OS	Operating System
PC	Personal Computer
PCIe	Peripheral Component Interconnect express (PCI express)
PDA	Personal Digital Assistant
PHY	PHYsical interface
POS	Point Of Sale
RAID	Redundant Array of Inexpensive/Independent Disks
RASIE	Redundant Array of Silicon Independent Elements
RS	Reed-Solomon
SAN	Storage Attached Network
SAS	Serial Attached Small Computer System Interface (Serial SCSI)
SATA	Serial Advanced Technology Attachment (Serial ATA)
SCSI	Small Computer System Interface
SD	Secure Digital
SDR	Single-Data-Rate
SLC	Single-Level Cell
SMART	Self-Monitoring Analysis and Reporting Technology
SRAM	Static Random Access Memory
SSD	Solid-State Disk/Drive
UFS	Unified Flash Storage
USB	Universal Serial Bus
VF	Virtual Function
WAN	Wide Area Network

1
2

[0027] NAND flash memory uses an array of floating gate transistors to store information. In SLC technology, each bit cell (e.g. floating gate transistor) is enabled to store one bit of information. In MLC technology, each bit cell is enabled to store multiple bits of information. As manufacturing technology (e.g. CMOS technology) scales down, each floating gate stores fewer electrons. Further, as storage capacity and density increase, each bit cell stores more bits. Therefore, values stored in the bit cells are represented by smaller voltage ranges. Uncertainties in sensing and/or changes in amount of stored electrons over time increase a probability for data to be stored or read incorrectly. Use of one or more redundancy and/or ECC techniques (e.g. at a lower-level) enables correct retrieval of otherwise corrupted data from NAND flash memory, overcoming, in some usage scenarios, some of the aforementioned difficulties.

[0028] Some types of SSDs use flash memory to provide non-volatile storage (e.g., the flash memory retains information without application of power). Use of one or more ECC and/or redundancy techniques (e.g. at a higher-level) enables correct retrieval of otherwise corrupted data from flash memory, and/or enables proper system-level operation of an SSD even when one or more flash memory elements fail intermittently or permanently.

[0029] For example, an SSD controller enables dynamic higher-level redundancy mode management with independent silicon elements to provide graceful degradation as one or more NVM (e.g. flash) elements fail during operation of an SSD implemented in part by the controller. A portion of the NVM is read. If an error occurs that is not correctable using lower-level redundancy and/or error correction (such as in accordance with one or more ECC techniques), then higher-level error correction (such as in accordance with one or more RASIE techniques and/or dynamic higher-level redundancy mode management techniques) is used to attempt to correct the error. If a failure of one of the NVM elements is detected by the lower-level and/or the higher-level error correction, and/or by other techniques (such as a failing status reported by one or more of the NVM elements), then the higher-level error correction is dynamically transitioned from operating in a current mode to operating in a new mode. The transition includes one or more of reducing free space available on the SSD, rearranging data storage of the SSD, recovering/storing failed user data (if possible), and determining/storing revised higher-level error correction information. Operation then continues in the new mode. If another failure of another one of the NVM elements is detected with the higher-level error correction now operating in the new mode, then another transition is made to another higher-level error correction mode. Writing the NVM is in accordance with a higher-level error

correction operating mode, including determining/storing higher-level error correction information in accordance with the higher-level error correction operating mode and write data.

[0030] Determining/storing higher-level error correction information is in accordance, in some embodiments and/or usage scenarios, with computation of higher-level redundancy information. Higher-level redundancy information computation enables an SSD controller to provide higher-level redundancy capabilities to maintain reliable operation in a context of failures of non-volatile (e.g. flash) memory elements during operation of an SSD implemented in part by the controller. A first portion of higher-level redundancy information is computed using parity coding via an XOR of all pages in a portion of data (e.g. a stripe) to be protected by the higher-level redundancy information. A second portion of the higher-level redundancy information is computed using a weighted-sum technique, each page in the portion being assigned a unique non-zero "index" as a weight when computing the weighted-sum. Arithmetic is performed over a finite field (such as a Galois Field, or such as the integers mod p where p is a prime).

[0031] The portions of the higher-level redundancy information are computable in any order, such as an order determined by an order of completion of one or more read operations performed on NVM elements, or such as an order based on order of data returned and/or available from NVM elements, enabling, in various embodiments, reduced or eliminated buffering. The any order computability enables, in various embodiments, computing recovery data values and/or backing out of a write using relatively little temporary and/or intermediate buffering and/or state. The portions of the higher-level redundancy information are computable with any degree of parallelism, such as determined by available dedicated hardware elements, enabling, in various embodiments, reduced latency processing and/or reduced memory (e.g. NVM) bandwidth usage.

[0032] In some embodiments, an I/O device, such as an SSD, includes an SSD controller. The SSD controller acts as a bridge between the host interface and NVM of the SSD, and executes commands of a host protocol sent from a computing host via a host interface of the SSD. At least some of the commands direct the SSD to write and read the NVM with data sent from and to the computing host, respectively. In further embodiments, the SSD controller is enabled to use a map to translate between LBAs of the host protocol and physical storage addresses in the NVM. In further embodiments, at least a portion of the map is used for private storage (not visible to the computing host) of the I/O device. For example, a portion of the

LBAs not accessible by the computing host is used by the I/O device to manage access to logs, statistics, or other private data.

[0033] In some embodiments, accessing compressed data of varying-sized quanta in NVM provides improved storage efficiency in some usage scenarios. For example, an SSD controller receives (uncompressed) data from a computing host (e.g., relating to a disk write command), compresses the data, and stores the compressed data into flash memory. In response to a subsequent request from the computing host (e.g., relating to a disk read command), the SSD controller reads the compressed data from the flash memory, uncompresses the compressed data, and provides the uncompressed data to the computing host. The compressed data is stored in the flash memory according to varying-sized quanta, the quanta size varying due to, e.g., compression algorithm, operating mode, and compression effectiveness on various data. The SSD controller uncompresses the data in part by consulting an included map table to determine where header(s) are stored in the flash memory. The SSD controller parses the header(s) obtained from the flash memory to determine where appropriate (compressed) data is stored in the flash memory. The SSD controller uncompresses the appropriate data from the flash memory to produce the uncompressed data to provide to the computing host. In the instant application, uncompress (and variants thereof) is synonymous with decompress (and variants thereof).

[0034] In various embodiments, an SSD controller includes a host interface for interfacing with a computing host, an interface for interfacing with NVM such as flash memory, and circuitry for controlling the interfaces and performing (and/or controlling various aspects of the performing) compressing and uncompressing, as well as lower-level error correction, higher-level error correction, and dynamic higher-level redundancy mode management with independent silicon elements.

[0035] According to various embodiments, some host interfaces are compatible with one or more of a USB interface standard, a CF interface standard, an MMC interface standard, an eMMC interface standard, a Thunderbolt interface standard, a UFS interface standard, an SD interface standard, a Memory Stick interface standard, an xD-picture card interface standard, an IDE interface standard, a SATA interface standard, a SCSI interface standard, a SAS interface standard, and a PCIe interface standard. According to various embodiments, the computing host is all or any portions of a computer, a workstation computer, a server computer, a storage server, a SAN, a NAS device, a DAS device, a storage appliance, a PC, a laptop computer, a notebook computer, a netbook computer, a tablet device or computer, an ultrabook computer, an electronic

reading device (such as an e-reader), a PDA, a navigation system, a (handheld) GPS device, an automotive control system, an automotive media control system or computer, a printer, copier or fax machine or all-in-one device, a POS device, a cash-register, a media player, a television, a media recorder, a DVR, a digital camera, a cellular handset, a cordless telephone handset, and an electronic game. In some embodiments, an interfacing host (such as an SAS/SATA bridge) operates as a computing host and/or as a bridge to a computing host.

[0036] In various embodiments, the SSD controller includes one or more processors. The processors execute firmware to control and/or perform operation of the SSD controller. The SSD controller communicates with the computing host to send and receive commands and/or status as well as data. The computing host executes one or more of an operating system, a driver, and an application. Communication by the computing host with the SSD controller is optionally and/or selectively via the driver and/or via the application. In a first example, all communication to the SSD controller is via the driver, and the application provides higher-level commands to the driver that the driver translates into specific commands for the SSD controller. In a second example, the driver implements a bypass mode and the application is enabled to send specific commands to the SSD controller via the driver. In a third example, a PCIe SSD controller supports one or more Virtual Functions (VFs), enabling an application, once configured, to communicate directly with the SSD controller, bypassing the driver.

[0037] According to various embodiments, some SSDs are compatible with form-factors, electrical interfaces, and/or protocols used by magnetic and/or optical non-volatile storage, such as HDDs, CD drives, and DVD drives. In various embodiments, SSDs use various combinations of zero or more parity codes, zero or more RS codes, zero or more BCH codes, zero or more Viterbi or other trellis codes, and zero or more LDPC codes.

EXAMPLE EMBODIMENTS

[0038] In concluding the introduction to the detailed description, what follows is a collection of example embodiments, including at least some explicitly enumerated as "ECs" (Example Combinations), providing additional description of a variety of embodiment types in accordance with the concepts described herein; these examples are not meant to be mutually exclusive, exhaustive, or restrictive; and the invention is not limited to these example embodiments but rather encompasses all possible modifications and variations within the scope of the issued claims.

[0039] EC1) A system comprising:

means for computing k redundant data portions to protect n-k useful data portions;

means for storing each of the k redundant data portions and each of the n-k useful data portions in separate corresponding areas, each of the corresponding areas in a respective one of n physical devices;

means for, subsequent to a failure such that one of the corresponding areas is no longer usable, computing j redundant data portions to protect n-1-j useful data portions; and

wherein j <= k.

[0040] EC2) The system of EC1, wherein the means for computing k redundant data portions to protect n-k useful data portions is operable in a first data protection mode, the means for computing j redundant data portions to protect n-1-j useful data portions is operable in a second data protection mode, and further comprising means for, in response to the failure, switching from operating in the first data protection mode to operating in the second data protection mode.

[0041] EC3) The system of EC1, wherein when j == k, a total amount of the useful data portions is decreased and a degree of protection is at least preserved.

[0042] EC4) The system of EC3, wherein the degree of protection is protection against failures of up to j of the corresponding areas.

[0043] EC5) The system of EC1, wherein when j == k-1 and j >=1, a total amount of the useful data portions is preserved, a degree of protection is decreased, and the useful data portions are protected.

[0044] EC6) The system of EC5, wherein the decreased degree of protection is protection against failures of up to j-1 of the corresponding areas.

[0045] EC7) The system of EC1,

wherein each of the n physical devices comprises a plurality of the corresponding areas;

further comprising means for storing, in each of a plurality of sets of the corresponding areas, a respective set of data comprising a useful data subset and a redundant data subset protecting the useful data subset, the redundant data subset stored in distinct ones of the n physical devices from the useful data subset; and

wherein each of the sets of the corresponding areas comprises no more than one corresponding area of each one of the n physical devices, and the means for storing is in accordance with each of the respective sets of data optionally being stored in a different set of n or less of the n physical devices.

[0046] EC8) The system of EC7, wherein prior to the failure, the useful data subset of a particular one of the sets of data comprises the n-k useful data portions, and the redundant data subset of the particular set of data comprises the k redundant data portions.

[0047] EC9) The system of EC7, wherein the means for storing is in accordance with a first one of the corresponding areas in a particular one of the n physical devices being used to store a portion of the useful data subset of a first one of the sets of data, and is further in accordance with a second one of the corresponding areas in the particular physical device being used to store a portion or all of the redundant data subset of a second one of the sets of data.

[0048] EC10) The system of EC7, wherein an amount of storage in a first one of the corresponding areas differs from an amount of storage in a second one of the corresponding areas.

[0049] EC11) The system of EC10, wherein the first corresponding area and the second corresponding area are in a same one of the n physical devices.

[0050] EC12) A system comprising:

means for operating in a first higher-level redundancy mode comprising means for storing n information portions in n respective non-overlapping areas of memory;

means for operating in a second higher-level redundancy mode comprising means for storing n-1 information portions in n-1 respective non-overlapping areas of the memory;

means for detecting a failure of a failing one of the n respective non-overlapping areas;

means for, in response to the failure, dynamically transitioning from operating via the means for operating in the first higher-level redundancy mode to operating via the means for operating in the second higher-level redundancy mode, the n-1 of the n respective non-overlapping areas being the n respective non-overlapping areas excluding the failing one of the n respective non-overlapping areas;

wherein each of the n information portions comprises respective n-k data information portions and respective k redundant information portions computed to protect the respective n-k data information portions;

wherein each of the n-1 information portions comprises respective n-1-j data information portions and respective j redundant information portions computed to protect the respective n-1-j data information portions;

wherein j <= k; and

wherein each of the n respective non-overlapping areas is in a respective one of n physical devices of the memory.

[0051] EC13) The system of EC12, wherein when j == k, a total amount of the data information portions is decreased and a degree of protection is at least preserved.

[0052] EC14) The system of EC13, wherein the degree of protection is protection against failures of up to j of the n respective non-overlapping areas.

[0053] EC15) The system of EC12, wherein when j == k-1 and j >=1, a total amount of the data information portions is preserved, a degree of protection is decreased, and the data information portions are protected.

[0054] EC16) The system of EC15, wherein the decreased degree of protection is protection against failures of up to j-1 of the n respective non-overlapping areas.

[0055] EC17) The system of EC1 or EC12, wherein at least one of the means for storing is responsive to write requests from a computing host.

[0056] EC18) The system of EC1 or EC12, wherein at least one of the means for storing comprises means for operating a flash memory interface.

[0057] EC19) The system of EC1 or EC12, wherein at least one of the n physical devices comprises one or more flash memories.

[0058] EC20) A system comprising:

means for managing one or more flash memories in accordance with a current higher-level redundancy mode in response to read and/or write requests from a computing host; and

means for dynamically transitioning the current higher-level redundancy mode in response to detecting a failure of one or more portions of one or more of the flash memories.

[0059] EC21) The system of EC20, wherein the means for dynamically transitioning comprises means for reducing free space.

[0060] EC22) The system of EC20, wherein the means for dynamically transitioning comprises means for rearranging data storage in accordance with a target higher-level redundancy mode.

[0061] EC23) The system of EC20, wherein the means for dynamically transitioning comprises means for determining and storing revised higher-level redundancy information in accordance with a target higher-level redundancy mode.

[0062] EC24) The system of EC20, wherein the current higher-level redundancy mode is a first higher-level redundancy mode and the means for dynamically transitioning the current higher-level redundancy mode comprises means for transitioning the current higher-level redundancy mode to a second higher-level redundancy mode in response to a first failure and means for transitioning the current higher-level redundancy mode to a third higher-level redundancy mode in response to a second failure.

[0063] EC25) The system of EC24, wherein the second failure is subsequent to the first failure and less free space is available in the third higher-level redundancy mode than in the second higher-level redundancy mode.

[0064] EC26) The system of EC17 or EC20, further comprising means for interfacing the requests with the computing host.

[0065] EC27) The system of EC26, wherein the means for interfacing the requests with the computing host is compatible with a storage interface standard.

[0066] EC28) The system of EC20, further comprising means for interfacing the means for managing with the flash memories.

[0067] EC29) The system of EC28, wherein the means for interfacing the means for managing with the flash memories comprises means for operating a flash memory interface.

[0068] EC30) The system of EC20, further comprising:

means for interfacing the requests with the computing host; and

means for interfacing the managing with the flash memories.

[0069] EC31) The system of EC27, wherein the system is implemented in a single Integrated Circuit (IC).

[0070] EC32) The system of EC27, wherein the system is implemented in a Solid-State Disk (SSD).

[0071] EC33) The system of EC17 or EC20, further comprising means for operating the computing host to generate at least some of the read and/or write requests.

[0072] EC34) The system of EC19 or EC20, further comprising means for operating at least one of the flash memories.

[0073] EC35) A method comprising:

computing k redundant data portions to protect n-k useful data portions;

storing each of the k redundant data portions and each of the n-k useful data portions in separate corresponding areas, each of the corresponding areas in a respective one of n physical devices;

subsequent to a failure such that one of the corresponding areas is no longer usable, computing j redundant data portions to protect n-1-j useful data portions; and

wherein j <= k.

[0074] EC36) The method of EC35, wherein the computing k redundant data portions to protect n-k useful data portions is a first data protection mode, the computing j redundant data portions to protect n-1-j useful data portions is a second data protection mode, and further comprising in response to the failure, switching from the first data protection mode to the second data protection mode.

[0075] EC37) The method of EC35, wherein when j == k, a total amount of the useful data portions is decreased and a degree of protection is at least preserved.

[0076] EC38) The method of EC37, wherein the degree of protection is protection against failures of up to j of the corresponding areas.

[0077] EC39) The method of EC35, wherein when j == k-1 and j >=1, a total amount of the useful data portions is preserved, a degree of protection is decreased, and the useful data portions are protected.

[0078] EC40) The method of EC39, wherein the decreased degree of protection is protection against failures of up to j-1 of the corresponding areas.

[0079] EC41) The method of EC35,

wherein each of the n physical devices comprises a plurality of the corresponding areas;

further comprising storing, in each of a plurality of sets of the corresponding areas, a respective set of data comprising a useful data subset and a redundant data subset protecting the useful data subset, the redundant data subset stored in distinct ones of the n physical devices from the useful data subset; and

wherein each of the sets of the corresponding areas comprises no more than one corresponding area of each one of the n physical devices, and the storing is in accordance with each of the respective sets of data optionally being stored in a different set of n or less of the n physical devices.

[0080] EC42) The method of EC41, wherein prior to the failure, the useful data subset of a particular one of the sets of data comprises the n-k useful data portions, and the redundant data subset of the particular set of data comprises the k redundant data portions.

[0081] EC43) The method of EC41, wherein the storing is in accordance with a first one of the corresponding areas in a particular one of the n physical devices being used to store a portion of the useful data subset of a first one of the sets of data, and is further in accordance with a second one of the corresponding areas in the particular physical device being used to store a portion or all of the redundant data subset of a second one of the sets of data.

[0082] EC44) The method of EC41, wherein an amount of storage in a first one of the corresponding areas differs from an amount of storage in a second one of the corresponding areas.

[0083] EC45) The method of EC44, wherein the first corresponding area and the second corresponding area are in a same one of the n physical devices.

[0084] EC46) A method comprising:

operating in a first higher-level redundancy mode comprising storing n information portions in n respective non-overlapping areas of memory;

detecting a failure of a failing one of the n respective non-overlapping areas;

in response to the failure, dynamically transitioning from operating in the first higher-level redundancy mode to operating in a second higher-level redundancy mode comprising storing n-1 information portions in n-1 of the n respective non-overlapping areas, the n-1 of the n respective non-overlapping areas being the n respective non-overlapping areas excluding the failing one of the n respective non-overlapping areas;

wherein each of the n information portions comprises respective n-k data information portions and respective k redundant information portions computed to protect the respective n-k data information portions;

wherein each of the n-1 information portions comprises respective n-1-j data information portions and respective j redundant information portions computed to protect the respective n-1-j data information portions;

wherein j <= k; and

wherein each of the n respective non-overlapping areas is in a respective one of n physical devices of the memory.

[0085] EC47) The method of EC46, wherein when j == k, a total amount of the data information portions is decreased and a degree of protection is at least preserved.

[0086] EC48) The method of EC47, wherein the degree of protection is protection against failures of up to j of the n respective non-overlapping areas.

[0087] EC49) The method of EC46, wherein when j == k-1 and j >=1, a total amount of the data information portions is preserved, a degree of protection is decreased, and the data information portions are protected.

[0088] EC50) The method of EC49, wherein the decreased degree of protection is protection against failures of up to j-1 of the n respective non-overlapping areas.

[0089] EC51) The method of EC35 or EC46, wherein at least a portion of at least one of the acts of storing is in response to write requests from a computing host.

[0090] EC52) The method of EC35 or EC46, wherein at least one of the acts of storing comprises operating a flash memory interface.

[0091] EC53) The method of EC35 or EC46, wherein at least one of the n physical devices comprises one or more flash memories.

[0092] EC54) A method comprising:

managing one or more flash memories in accordance with a current higher-level redundancy mode in response to read and/or write requests from a computing host; and

dynamically transitioning the current higher-level redundancy mode in response to detecting a failure of one or more portions of one or more of the flash memories.

[0093] EC55) The method of EC54, wherein the dynamically transitioning comprises reducing free space.

[0094] EC56) The method of EC54, wherein the dynamically transitioning comprises rearranging data storage in accordance with a target higher-level redundancy mode.

[0095] EC57) The method of EC54, wherein the dynamically transitioning comprises determining and storing revised higher-level redundancy information in accordance with a target higher-level redundancy mode.

[0096] EC58) The method of EC54, wherein the current higher-level redundancy mode is a first higher-level redundancy mode and the dynamically transitioning the current higher-level redundancy mode transitions the current higher-level redundancy mode to a second higher-level redundancy mode in response to a first failure and transitions the current higher-level redundancy mode to a third higher-level redundancy mode in response to a second failure.

[0097] EC59) The method of EC58, wherein the second failure is subsequent to the first failure and less free space is available in the third higher-level redundancy mode than in the second higher-level redundancy mode.

[0098] EC60) The method of EC51 or EC54, further comprising interfacing the requests with the computing host.

[0099] EC61) The method of EC60, wherein the interfacing the requests with the computing host is compatible with a storage interface standard.

[0100] EC62) The method of EC54, further comprising interfacing the managing with the flash memories.

[0101] EC63) The method of EC62, wherein the interfacing the managing with the flash memories comprises operating a flash memory interface.

[0102] EC64) The method of EC54, further comprising:

interfacing the requests with the computing host; and

interfacing the managing with the flash memories.

[0103] EC65) The method of EC61, wherein the method is implemented in a single Integrated Circuit (IC).

[0104] EC66) The method of EC61, wherein the method is implemented in a Solid-State Disk (SSD).

[0105] EC67) The method of EC51 or EC54, further comprising operating the computing host to generate at least some of the read and/or write requests.

[0106] EC68) The method of EC53 or EC54, further comprising operating at least one of the flash memories.

[0107] EC69) A system comprising:

a host interface enabled to interface to a computing host;

a memory interface enabled to interface to at least n physical devices;

a controller enabled to

compute k redundant data portions to protect n-k useful data portions,

in response to one or more write commands received from the computing host via the host interface, store each of the k redundant data portions and each of the n-k useful data portions in separate corresponding areas, each of the corresponding areas in a respective one of the n physical devices, and

subsequent to a failure such that one of the corresponding areas is no longer usable, compute j redundant data portions to protect n-1-j useful data portions; and

wherein j <= k.

[0108] EC70) The system of EC69, further comprising a hardware logic circuit enabled to compute all or any portions of any one or more of the k redundant data portions.

[0109] EC71) The system of EC69, wherein the computing k redundant data portions to protect n-k useful data portions is in accordance with the controller operating in a first data protection mode, the computing j redundant data portions to protect n-1-j useful data portions is in accordance with the controller operating in a second data protection mode, and the controller is further enabled, in response to the failure, to switch from operating in the first data protection mode to operating in the second data protection mode.

[0110] EC72) The system of EC69, wherein when j == k, a total amount of the useful data portions is decreased and a degree of protection is at least preserved.

[0111] EC73) The system of EC72, wherein the degree of protection is protection against failures of up to j of the corresponding areas.

[0112] EC74) The system of EC69, wherein when j == k-1 and j >=1, a total amount of the useful data portions is preserved, a degree of protection is decreased, and the useful data portions are protected.

[0113] EC75) The system of EC74, wherein the decreased degree of protection is protection against failures of up to j-1 of the corresponding areas.

[0114] EC76) The system of EC69,

wherein each of the n physical devices comprises a plurality of the corresponding areas;

wherein the controller is further enabled to store, in each of a plurality of sets of the corresponding areas, a respective set of data comprising a useful data subset and a redundant data subset protecting the useful data subset, the redundant data subset stored in distinct ones of the n physical devices from the useful data subset; and

wherein each of the sets of the corresponding areas comprises no more than one corresponding area of each one of the n physical devices, and the storing is in accordance with each of the respective sets of data optionally being stored in a different set of n or less of the n physical devices.

[0115] EC77) The system of EC76, wherein prior to the failure, the useful data subset of a particular one of the sets of data comprises the n-k useful data portions, and the redundant data subset of the particular set of data comprises the k redundant data portions.

[0116] EC78) The system of EC76, wherein the storing is in accordance with a first one of the corresponding areas in a particular one of the n physical devices being used to store a portion of the useful data subset of a first one of the sets of data, and is further in accordance with a second one of the corresponding areas in the particular physical device being used to store a portion or all of the redundant data subset of a second one of the sets of data.

[0117] EC79) The system of EC76, wherein an amount of storage in a first one of the corresponding areas differs from an amount of storage in a second one of the corresponding areas.

[0118] EC80) The system of EC79, wherein the first corresponding area and the second corresponding area are in a same one of the n physical devices.

[0119] EC81) A system comprising:

a host interface enabled to interface to a computing host;

a memory interface enabled to interface to at least n physical devices;

a controller enabled to

operate in a first higher-level redundancy mode in accordance with storing n information portions in n respective non-overlapping areas of memory;

detect a failure of a failing one of the n respective non-overlapping areas;

in response to the failure, dynamically transitioning from operating in the first higher-level redundancy mode to operating in a second higher-level redundancy mode, the second higher-level redundancy mode being in accordance with storing n-1 information portions in n-1 of the n respective non-overlapping areas, the n-1 of the n respective non-overlapping areas being the n respective non-overlapping areas excluding the failing one of the n respective non-overlapping areas;

wherein each of the n information portions comprises respective n-k data information portions and respective k redundant information portions computed to protect the respective n-k data information portions;

wherein each of the n-1 information portions comprises respective n-1-j data information portions and respective j redundant information portions computed to protect the respective n-1-j data information portions;

wherein j <= k; and

wherein each of the n respective non-overlapping areas is in a respective one of n physical devices of the memory.

[0120] EC82) The system of EC81, further comprising a hardware logic circuit enabled to compute all or any portions of any one or more of the k redundant information portions.

[0121] EC83) The system of EC81, wherein when j == k, a total amount of the data information portions is decreased and a degree of protection is at least preserved.

[0122] EC84) The system of EC83, wherein the degree of protection is protection against failures of up to j of the n respective non-overlapping areas.

[0123] EC85) The system of EC81, wherein when j == k-1 and j >=1, a total amount of the data information portions is preserved, a degree of protection is decreased, and the data information portions are protected.

[0124] EC86) The system of EC85, wherein the decreased degree of protection is protection against failures of up to j-1 of the n respective non-overlapping areas.

[0125] EC87) The system of EC69 or EC81, wherein at least a portion of at least one of the acts of storing is in response to write requests from the computing host.

[0126] EC88) The system of EC69 or EC81, wherein the memory interface comprises a flash memory interface.

[0127] EC89) The system of EC69 or EC81, wherein at least one of the n physical devices comprises one or more flash memories.

[0128] EC90) A system comprising:

a host interface enabled to interface to a computing host;

a controller enabled to

manage one or more flash memories in accordance with a current higher-level redundancy mode in response to read and/or write requests from the computing host; and

dynamically transition the current higher-level redundancy mode in response to detecting a failure of one or more portions of one or more of the flash memories.

[0129] EC91) The system of EC90, further comprising a hardware logic circuit enabled to compute redundancy information usable in the current higher-level redundancy mode.

[0130] EC92) The system of EC90, wherein the controller is further enabled to reduce free space in accordance with the dynamic transition.

[0131] EC93) The system of EC90, wherein the controller is further enabled to rearrange data storage in accordance with a target higher-level redundancy mode and in accordance with the dynamic transition.

[0132] EC94) The system of EC90, wherein the controller is further enabled to determine and store revised higher-level redundancy information in accordance with a target higher-level redundancy mode and in accordance with the dynamic transition

[0133] EC95) The system of EC90, wherein the current higher-level redundancy mode is a first higher-level redundancy mode and the controller is further enabled to transition the current higher-level redundancy mode to a second higher-level redundancy mode in response to a first failure and to transition the current higher-level redundancy mode to a third higher-level redundancy mode in response to a second failure.

[0134] EC96) The system of EC95, wherein the second failure is subsequent to the first failure and less free space is available in the third higher-level redundancy mode than in the second higher-level redundancy mode.

[0135] EC97) The system of EC90, wherein the host interface is compatible with a storage interface standard.

[0136] EC98) The system of EC90, further comprising a flash memory interface enabled to interface with the flash memories.

[0137] EC99) The system of EC88 or EC97, wherein the system is implemented in a single Integrated Circuit (IC).

[0138] EC100) The system of EC88 or EC97, wherein the system is implemented in a Solid-State Disk (SSD).

[0139] EC101) The system of EC87 or EC90, further comprising the computing host.

[0140] EC102) The system of EC89 or EC90, further comprising at least one of the flash memories.

[0141] EC103) A tangible computer readable medium having a set of instructions stored therein that when executed by a processing element cause the processing element to perform operations comprising:

computing k redundant data portions to protect n-k useful data portions;

storing each of the k redundant data portions and each of the n-k useful data portions in separate corresponding areas, each of the corresponding areas in a respective one of n physical devices;

subsequent to a failure such that one of the corresponding areas is no longer usable, computing j redundant data portions to protect n-1-j useful data portions; and

wherein j <= k.

[0142] EC104) The tangible computer readable medium of EC103, wherein the computing k redundant data portions to protect n-k useful data portions is in accordance with a first data protection mode, the computing j redundant data portions to protect n-1-j useful data portions is in accordance with a second data protection mode, and further comprising in response to the failure, switching from the first data protection mode to the second data protection mode.

[0143] EC105) The tangible computer readable medium of EC103, wherein when j == k, a total amount of the useful data portions is decreased and a degree of protection is at least preserved.

[0144] EC106) The tangible computer readable medium of EC105, wherein the degree of protection is protection against failures of up to j of the corresponding areas.

[0145] EC107) The tangible computer readable medium of EC103, wherein when j == k-1 and j >=1, a total amount of the useful data portions is preserved, a degree of protection is decreased, and the useful data portions are protected.

[0146] EC108) The tangible computer readable medium of EC107, wherein the decreased degree of protection is protection against failures of up to j-1 of the corresponding areas.

[0147] EC109) The tangible computer readable medium of EC103,

wherein each of the n physical devices comprises a plurality of the corresponding areas;

further comprising storing, in each of a plurality of sets of the corresponding areas, a respective set of data comprising a useful data subset and a redundant data subset protecting the useful data subset, the redundant data subset stored in distinct ones of the n physical devices from the useful data subset; and

wherein each of the sets of the corresponding areas comprises no more than one corresponding area of each one of the n physical devices, and the storing is in accordance with each of the respective sets of data optionally being stored in a different set of n or less of the n physical devices.

[0148] EC110) The tangible computer readable medium of EC109, wherein prior to the failure, the useful data subset of a particular one of the sets of data comprises the n-k useful data portions, and the redundant data subset of the particular set of data comprises the k redundant data portions.

[0149] EC111) The tangible computer readable medium of EC109, wherein the storing is in accordance with a first one of the corresponding areas in a particular one of the n physical devices being used to store a portion of the useful data subset of a first one of the sets of data, and is further in accordance with a second one of the corresponding areas in the particular physical device being used to store a portion or all of the redundant data subset of a second one of the sets of data.

[0150] EC112) The tangible computer readable medium of EC109, wherein an amount of storage in a first one of the corresponding areas differs from an amount of storage in a second one of the corresponding areas.

[0151] EC113) The tangible computer readable medium of EC112, wherein the first corresponding area and the second corresponding area are in a same one of the n physical devices.

[0152] EC114) A tangible computer readable medium having a set of instructions stored therein that when executed by a processing element cause the processing element to perform operations comprising:

operating in a first higher-level redundancy mode comprising storing n information portions in n respective non-overlapping areas of memory;

detecting a failure of a failing one of the n respective non-overlapping areas;

in response to the failure, dynamically transitioning from operating in the first higher-level redundancy mode to operating in a second higher-level redundancy mode comprising storing n-1 information portions in n-1 of the n respective non-overlapping areas, the n-1 of the n respective non-overlapping areas being the n respective non-overlapping areas excluding the failing one of the n respective non-overlapping areas;

wherein each of the n information portions comprises respective n-k data information portions and respective k redundant information portions computed to protect the respective n-k data information portions;

wherein each of the n-1 information portions comprises respective n-1-j data information portions and respective j redundant information portions computed to protect the respective n-1-j data information portions;

wherein j <= k; and

wherein each of the n respective non-overlapping areas is in a respective one of n physical devices of the memory.

[0153] EC115) The tangible computer readable medium of EC114, wherein when j == k, a total amount of the data information portions is decreased and a degree of protection is at least preserved.

[0154] EC116) The tangible computer readable medium of EC115, wherein the degree of protection is protection against failures of up to j of the n respective non-overlapping areas.

[0155] EC117) The tangible computer readable medium of EC114, wherein when j == k-1 and j >=1, a total amount of the data information portions is preserved, a degree of protection is decreased, and the data information portions are protected.

[0156] EC118) The tangible computer readable medium of EC117, wherein the decreased degree of protection is protection against failures of up to j-1 of the n respective non-overlapping areas.

[0157] EC119) The tangible computer readable medium of EC103 or EC114, wherein at least a portion of at least one of the acts of storing is in response to write requests from a computing host.

[0158] EC120) The tangible computer readable medium of EC103 or EC114, wherein at least one of the acts of storing comprises operating a flash memory interface.

[0159] EC121) The tangible computer readable medium of EC103 or EC114, wherein at least one of the n physical devices comprises one or more flash memories.

[0160] EC122) A tangible computer readable medium having a set of instructions stored therein that when executed by a processing element cause the processing element to perform operations comprising:

managing one or more flash memories in accordance with a current higher-level redundancy mode in response to read and/or write requests from a computing host; and

dynamically transitioning the current higher-level redundancy mode in response to detecting a failure of one or more portions of one or more of the flash memories.

[0161] EC123) The tangible computer readable medium of EC122, wherein the dynamically transitioning comprises reducing free space.

[0162] EC124) The tangible computer readable medium of EC122, wherein the dynamically transitioning comprises rearranging data storage in accordance with a target higher-level redundancy mode.

[0163] EC125) The tangible computer readable medium of EC122, wherein the dynamically transitioning comprises determining and storing revised higher-level redundancy information in accordance with a target higher-level redundancy mode.

[0164] EC126) The tangible computer readable medium of EC122, wherein the current higher-level redundancy mode is a first higher-level redundancy mode and the dynamically transitioning the current higher-level redundancy mode transitions the current higher-level redundancy mode to a second higher-level redundancy mode in response to a first failure and transitions the current higher-level redundancy mode to a third higher-level redundancy mode in response to a second failure.

[0165] EC127) The tangible computer readable medium of EC126, wherein the second failure is subsequent to the first failure and less free space is available in the third higher-level redundancy mode than in the second higher-level redundancy mode.

[0166] EC128) The tangible computer readable medium of EC119 or EC122, further comprising interfacing the requests with the computing host.

[0167] EC129) The tangible computer readable medium of EC128, wherein the interfacing the requests with the computing host is compatible with a storage interface standard.

[0168] EC130) The tangible computer readable medium of EC122, further comprising interfacing the managing with the flash memories.

[0169] EC131) The tangible computer readable medium of EC130, wherein the interfacing the managing with the flash memories comprises operating a flash memory interface.

[0170] EC132) The tangible computer readable medium of EC122, further comprising:

interfacing the requests with the computing host; and

interfacing the managing with the flash memories.

[0171] EC133) The tangible computer readable medium of EC129, wherein the tangible computer readable medium is implemented in a single Integrated Circuit (IC).

[0172] EC134) The tangible computer readable medium of EC129, wherein the tangible computer readable medium is implemented in a Solid-State Disk (SSD).

[0173] EC135) The tangible computer readable medium of EC119 or EC122, further comprising operating the computing host to generate at least some of the read and/or write requests.

[0174] EC136) The tangible computer readable medium of EC121 or EC122, further comprising operating at least one of the flash memories.

[0175] EC137) Any of the foregoing ECs having a storage interface standard, wherein the storage interface standard comprises one or more of

a Universal Serial Bus (USB) interface standard,
a Compact Flash (CF) interface standard,
a MultiMediaCard (MMC) interface standard,
an embedded MMC (eMMC) interface standard,
a Thunderbolt interface standard,
a UFS interface standard,
a Secure Digital (SD) interface standard,
a Memory Stick interface standard,
an xD-picture card interface standard,
an Integrated Drive Electronics (IDE) interface standard,
a Serial Advanced Technology Attachment (SATA) interface standard,
an external SATA (eSATA) interface standard,
a Small Computer System Interface (SCSI) interface standard,
a Serial Attached Small Computer System Interface (SAS) interface standard,
a Fibre Channel interface standard,
an Ethernet interface standard, and
a Peripheral Component Interconnect express (PCIe) interface standard.

[0176] EC138) Any of the foregoing ECs having a flash memory interface, wherein the flash memory interface is compatible with one or more of

an Open NAND Flash Interface (ONFI),
a Toggle-mode interface,
a Double-Data-Rate (DDR) synchronous interface,
a DDR2 synchronous interface;
a synchronous interface, and
an asynchronous interface.

[0177] EC139) Any of the foregoing ECs having a computing host, wherein the computing host comprises one or more of

a computer,
a workstation computer,
a server computer,
a storage server,
a Storage Attached Network (SAN),
a Network Attached Storage (NAS) device,
a Direct Attached Storage (DAS) device,
a storage appliance,
a Personal Computer (PC),
a laptop computer,
a notebook computer,
a netbook computer,
a tablet device or computer,
an ultrabook computer,
an electronic reading device (an e-reader),
a Personal Digital Assistant (PDA),
a navigation system,
a (handheld) Global Positioning System (GPS) device,
an automotive control system,
an automotive media control system or computer,
a printer, copier or fax machine or all-in-one device,
a Point Of Sale POS device,
a cash-register,
a media player,
a television,
a media recorder,
a Digital Video Recorder (DVR),
a digital camera,
a cellular handset,
a cordless telephone handset, and
an electronic game.

[0178] EC140) Any of the foregoing ECs having a at least one flash memory, wherein at least a portion of the at least one flash memory comprises one or more of

NAND flash technology storage cells, and

NOR flash technology storage cells.

[0179] EC141) Any of the foregoing ECs having a at least one flash memory, wherein at least a portion of the at least one flash memory comprises one or more of

Single-Level Cell (SLC) flash technology storage cells, and

Multi-Level Cell (MLC) flash technology storage cells.

[0180] EC142) Any of the foregoing ECs having a at least one flash memory, wherein at least a portion of the at least one flash memory comprises one or more of

polysilicon technology-based charge storage cells, and

silicon nitride technology-based charge storage cells.

[0181] EC143) Any of the foregoing ECs having a at least one flash memory, wherein at least a portion of the at least one flash memory comprises one or more of

two-dimensional technology-based flash memory technology, and

three-dimensional technology-based flash memory technology.

SYSTEM

[0182] Fig. 1A illustrates selected details of an embodiment of a Solid-State Disk (SSD) including an SSD controller using dynamic higher-level redundancy mode management with independent silicon elements for accessing and controlling NVMs. The SSD controller is for managing non-volatile storage, such as implemented via NVM elements (e.g., flash memories). SSD Controller **100** is communicatively coupled via one or more External Interfaces **110** to a host (not illustrated). According to various embodiments, External Interfaces **110** are one or more of: a SATA interface; a SAS interface; a PCIe interface; a Fibre Channel interface; an Ethernet Interface (such as 10 Gigabit Ethernet); a non-standard version of any of the preceding interfaces; a custom interface; or any other type of interface used to interconnect storage and/or communications and/or computing devices. For example, in some embodiments, SSD Controller **100** includes a SATA interface and a PCIe interface.

[0183] SSD Controller **100** is further communicatively coupled via one or more Device Interfaces **190** to NVM **199** including one or more storage devices, such as one or more of Flash Device **192**. According to various embodiments, Device Interfaces **190** are one or more of: an asynchronous interface; a synchronous interface; a single-data-rate (SDR) interface; a double-data-rate (DDR) interface; a DRAM-compatible DDR or DDR2 synchronous interface; an ONFI compatible interface, such as an ONFI 2.2 or ONFI 3.0 compatible interface; a Toggle-mode compatible flash interface; a non-standard version of any of the preceding interfaces; a custom interface; or any other type of interface used to connect to storage devices.

[0184] Each Flash Device **192** has, in some embodiments, one or more individual Flash Die **194**. According to type of a particular one of Flash Device **192**, a plurality of Flash Die **194** in the particular Flash Device **192** is optionally and/or selectively accessible in parallel. Flash Device **192** is merely representative of one type of storage device enabled to communicatively couple to SSD Controller **100**. In various embodiments, any type of storage device is usable, such as an SLC NAND flash memory, MLC NAND flash memory, NOR flash memory, flash memory using polysilicon or silicon nitride technology-based charge storage cells, two- or three-dimensional technology-based flash memory, read-only memory, static random access memory, dynamic random access memory, ferromagnetic memory, phase-change memory, racetrack memory, or any other type of memory device or storage medium.

[0185] According to various embodiments, Device Interfaces **190** are organized as: one or more busses with one or more of Flash Device **192** per bus; one or more groups of busses

with one or more of Flash Device **192** per bus, where busses in a group are generally accessed in parallel; or any other organization of one or more of Flash Device **192** onto Device Interfaces **190**.

[0186] Continuing in Fig. 1A, SSD Controller **100** has one or more modules, such as Host Interfaces **111**, Data Processing **121**, Buffer **131**, Map **141**, Recycler **151**, ECC **161**, Device Interface Logic **191**, and CPU **171**. The specific modules and interconnections illustrated in Fig. 1A are merely representative of one embodiment, and many arrangements and interconnections of some or all of the modules, as well as additional modules not illustrated, are conceived. In a first example, in some embodiments, there are two or more Host Interfaces **111** to provide dual-porting. In a second example, in some embodiments, Data Processing **121** and/or ECC **161** are combined with Buffer **131**. In a third example, in some embodiments, Host Interfaces **111** is directly coupled to Buffer **131**, and Data Processing **121** optionally and/or selectively operates on data stored in Buffer **131**. In a fourth example, in some embodiments, Device Interface Logic **191** is directly coupled to Buffer **131**, and ECC **161** optionally and/or selectively operates on data stored in Buffer **131**.

[0187] Host Interfaces **111** sends and receives commands and/or data via External Interfaces **110**, and, in some embodiments, tracks progress of individual commands via Tag Tracking **113**. For example, the commands include a read command specifying an address (such as an LBA) and an amount of data (such as a number of LBA quanta, e.g., sectors) to read; in response the SSD provides read status and/or read data. For another example, the commands include a write command specifying an address (such as an LBA) and an amount of data (such as a number of LBA quanta, e.g., sectors) to write; in response the SSD provides write status and/or requests write data and optionally subsequently provides write status. For yet another example, the commands include a de-allocation command (e.g. a trim command) specifying one or more addresses (such as one or more LBAs) that no longer need be allocated; in response the SSD modifies the map accordingly and optionally provides de-allocation status. In some contexts an ATA compatible TRIM command is an exemplary de-allocation command. For yet another example, the commands include a super capacitor test command or a data hardening success query; in response, the SSD provides appropriate status. In some embodiments, Host Interfaces **111** is compatible with a SATA protocol and, using NCQ commands, is enabled to have up to 32 pending commands, each with a unique tag represented as a number from 0 to 31. In some embodiments, Tag Tracking **113** is enabled to associate an external tag for a command received via External Interfaces **110** with an internal tag used to track the command during processing by SSD Controller **100**.

[0188] According to various embodiments, one or more of: Data Processing **121** optionally and/or selectively processes some or all data sent between Buffer **131** and External Interfaces **110**; and Data Processing **121** optionally and/or selectively processes data stored in Buffer **131**. In some embodiments, Data Processing **121** uses one or more Engines **123** to perform one or more of: formatting; reformatting; transcoding; and any other data processing and/or manipulation task.

[0189] Buffer **131** stores data sent to/from External Interfaces **110** from/to Device Interfaces **190**. In some embodiments, Buffer **131** additionally stores system data, such as some or all map tables, used by SSD Controller **100** to manage one or more of Flash Device **192**. In various embodiments, Buffer **131** has one or more of: Memory **137** used for temporary storage of data; DMA **133** used to control movement of data to and/or from Buffer **131**; and ECC-X **135** used to provide higher-level error correction and/or redundancy functions; and other data movement and/or manipulation functions. An example of a higher-level redundancy function is a RAID-like capability (e.g. RASIE), where redundancy is at a flash device (e.g., multiple ones of Flash Device **192**) level and/or a flash die (e.g., Flash Die **194**) level instead of at a disk level.

[0190] According to various embodiments, one or more of: ECC **161** optionally and/or selectively processes some or all data sent between Buffer **131** and Device Interfaces **190**; and ECC **161** optionally and/or selectively processes data stored in Buffer **131**. In some embodiments, ECC **161** is used to provide lower-level error correction and/or redundancy functions, such as in accordance with one or more ECC techniques. In some embodiments, ECC **161** implements one or more of: a CRC code; a Hamming code; an RS code; a BCH code; an LDPC code; a Viterbi code; a trellis code; a hard-decision code; a soft-decision code; an erasure-based code; any error detecting and/or correcting code; and any combination of the preceding. In some embodiments, ECC **161** includes one or more decoders (such as LDPC decoders).

[0191] Device Interface Logic **191** controls instances of Flash Device **192** via Device Interfaces **190**. Device Interface Logic **191** is enabled to send data to/from the instances of Flash Device **192** according to a protocol of Flash Device **192**. Device Interface Logic **191** includes Scheduling **193** to selectively sequence control of the instances of Flash Device **192** via Device Interfaces **190**. For example, in some embodiments, Scheduling **193** is enabled to queue operations to the instances of Flash Device **192**, and to selectively send the operations to individual ones of the instances of Flash Device **192** (or Flash Die **194**) as individual ones of the instances of Flash Device **192** (or Flash Die **194**) are available.

[0192] Map **141** converts between data addressing used on External Interfaces **110** and data addressing used on Device Interfaces **190**, using Table **143** to map external data addresses to locations in NVM **199**. For example, in some embodiments, Map **141** converts LBAs used on External Interfaces **110** to block and/or page addresses targeting one or more Flash Die **194**, via mapping provided by Table **143**. For LBAs that have never been written since drive manufacture or de-allocation, the map points to a default value to return if the LBAs are read. For example, when processing a de-allocation command, the map is modified so that entries corresponding to the de-allocated LBAs point to one of the default values. In various embodiments, there are various default values, each having a corresponding pointer. The plurality of default values enables reading some de-allocated LBAs (such as in a first range) as one default value, while reading other de-allocated LBAs (such as in a second range) as another default value. The default values, in various embodiments, are defined by flash memory, hardware, firmware, command and/or primitive arguments and/or parameters, programmable registers, or various combinations thereof.

[0193] In some embodiments, Map **141** uses Table **143** to perform and/or to look up translations between addresses used on External Interfaces **110** and data addressing used on Device Interfaces **190**. According to various embodiments, Table **143** is one or more of: a one-level map; a two-level map; a multi-level map; a map cache; a compressed map; any type of mapping from one address space to another; and any combination of the foregoing. According to various embodiments, Table **143** includes one or more of: static random access memory; dynamic random access memory; NVM (such as flash memory); cache memory; on-chip memory; off-chip memory; and any combination of the foregoing.

[0194] In some embodiments, Recycler **151** performs garbage collection. For example, in some embodiments, instances of Flash Device **192** contain blocks that must be erased before the blocks are re-writeable. Recycler **151** is enabled to determine which portions of the instances of Flash Device **192** are actively in use (e.g., allocated instead of de-allocated), such as by scanning a map maintained by Map **141**, and to make unused (e.g., de-allocated) portions of the instances of Flash Device **192** available for writing by erasing them. In further embodiments, Recycler **151** is enabled to move data stored within instances of Flash Device **192** to make larger contiguous portions of the instances of Flash Device **192** available for writing.

[0195] In some embodiments, instances of Flash Device **192** are selectively and/or dynamically configured, managed, and/or used to have one or more bands for storing data of

different types and/or properties. A number, arrangement, size, and type of the bands are dynamically changeable. For example, data from a computing host is written into a hot (active) band, while data from Recycler **151** is written into a cold (less active) band. In some usage scenarios, if the computing host writes a long, sequential stream, then a size of the hot band grows, whereas if the computing host does random writes or few writes, then a size of the cold band grows.

[0196] CPU **171** controls various portions of SSD Controller **100**. CPU **171** includes CPU Core **172**. CPU Core **172** is, according to various embodiments, one or more single-core or multi-core processors. The individual processors cores in CPU Core **172** are, in some embodiments, multi-threaded. CPU Core **172** includes instruction and/or data caches and/or memories. For example, the instruction memory contains instructions to enable CPU Core **172** to execute programs (e.g. software sometimes called firmware) to control SSD Controller **100**. In some embodiments, some or all of the firmware executed by CPU Core **172** is stored on instances of Flash Device **192** (as illustrated, e.g., as Firmware **106** of NVM **199** in Fig. 1B).

[0197] In various embodiments, CPU **171** further includes: Command Management **173** to track and control commands received via External Interfaces **110** while the commands are in progress; Buffer Management **175** to control allocation and use of Buffer **131**; Translation Management **177** to control Map **141**; Coherency Management **179** to control consistency of data addressing and to avoid conflicts such as between external data accesses and recycle data accesses; Device Management **181** to control Device Interface Logic **191**; Identity Management **182** to control modification and communication of identify information, and optionally other management units. None, any, or all of the management functions performed by CPU **171** are, according to various embodiments, controlled and/or managed by hardware, by software (such as firmware executing on CPU Core **172** or on a host connected via External Interfaces **110**), or any combination thereof.

[0198] In some embodiments, CPU **171** is enabled to perform other management tasks, such as one or more of: gathering and/or reporting performance statistics; implementing SMART; controlling power sequencing, controlling and/or monitoring and/or adjusting power consumption; responding to power failures; controlling and/or monitoring and/or adjusting clock rates; and other management tasks.

[0199] Various embodiments include a computing-host flash memory controller that is similar to SSD Controller **100** and is compatible with operation with various computing hosts,

such as via adaptation of Host Interfaces **111** and/or External Interfaces **110**. The various computing hosts include one or any combination of a computer, a workstation computer, a server computer, a storage server, a SAN, a NAS device, a DAS device, a storage appliance, a PC, a laptop computer, a notebook computer, a netbook computer, a tablet device or computer, an ultrabook computer, an electronic reading device (such as an e-reader), a PDA, a navigation system, a (handheld) GPS device, an automotive control system, an automotive media control system or computer, a printer, copier or fax machine or all-in-one device, a POS device, a cash-register, a media player, a television, a media recorder, a DVR, a digital camera, a cellular handset, a cordless telephone handset, and an electronic game.

[0200] In various embodiments, all or any portions of an SSD controller (or a computing-host flash memory controller) are implemented on a single IC, a single die of a multi-die IC, a plurality of dice of a multi-die IC, or a plurality of ICs. For example, Buffer **131** is implemented on a same die as other elements of SSD Controller **100**. For another example, Buffer **131** is implemented on a different die than other elements of SSD Controller **100**.

[0201] Fig. 1B illustrates selected details of various embodiments of systems including one or more instances of the SSD of Fig. 1A. SSD **101** includes SSD Controller **100** coupled to NVM **199** via Device Interfaces **190**. The figure illustrates various classes of embodiments: a single SSD coupled directly to a host, a plurality of SSDs each respectively coupled directly to a host via respective external interfaces, and one or more SSDs coupled indirectly to a host via various interconnection elements.

[0202] As an example embodiment of a single SSD coupled directly to a host, one instance of SSD **101** is coupled directly to Host **102** via External Interfaces **110** (e.g. Switch / Fabric / Intermediate Controller **103** is omitted, bypassed, or passed-through). As an example embodiment of a plurality of SSDs each coupled directly to a host via respective external interfaces, each of a plurality of instances of SSD **101** is respectively coupled directly to Host **102** via a respective instance of External Interfaces **110** (e.g. Switch / Fabric / Intermediate Controller **103** is omitted, bypassed, or passed-through). As an example embodiment of one or more SSDs coupled indirectly to a host via various interconnection elements, each of one or more instances of SSD **101** is respectively coupled indirectly to Host **102**. Each indirect coupling is via a respective instance of External Interfaces **110** coupled to Switch / Fabric / Intermediate Controller **103**, and Intermediate Interfaces **104** coupling to Host **102**.

[0203] Some of the embodiments including Switch / Fabric / Intermediate Controller **103** also include Card Memory **112C** coupled via Memory Interface **180** and accessible by the SSDs. In various embodiments, one or more of the SSDs, the Switch / Fabric / Intermediate Controller, and/or the Card Memory are included on a physically identifiable module, card, or pluggable element (e.g. I/O Card **116**). In some embodiments, SSD **101** (or variations thereof) corresponds to a SAS drive or a SATA drive that is coupled to an initiator operating as Host **102**.

[0204] Host **102** is enabled to execute various elements of Host Software **115**, such as various combinations of OS **105**, Driver **107**, Application **109**, and Multi-Device Management Software **114**. Dotted-arrow **107D** is representative of Host Software ←→ I/O Device Communication, e.g. data sent/received to/from one or more of the instances of SSD **101** and from/to any one or more of OS **105** via Driver **107**, Driver **107**, and Application **109**, either via Driver **107**, or directly as a VF.

[0205] OS **105** includes and/or is enabled to operate with drivers (illustrated conceptually by Driver **107**) for interfacing with the SSD. Various versions of Windows (e.g. 95, 98, ME, NT, XP, 2000, Server, Vista, and 7), various versions of Linux (e.g. Red Hat, Debian, and Ubuntu), and various versions of MacOS (e.g. 8, 9 and X) are examples of OS **105**. In various embodiments, the drivers are standard and/or generic drivers (sometimes termed "shrink-wrapped" or "pre-installed") operable with a standard interface and/or protocol such as SATA, AHCI, or NVM Express, or are optionally customized and/or vendor specific to enable use of commands specific to SSD **101**. Some drives and/or drivers have pass-through modes to enable application-level programs, such as Application **109** via Optimized NAND Access (sometimes termed ONA) or Direct NAND Access (sometimes termed DNA) techniques, to communicate commands directly to SSD **101**, enabling a customized application to use commands specific to SSD **101** even with a generic driver. ONA techniques include one or more of: use of non-standard modifiers (hints); use of vendor-specific commands; communication of non-standard statistics, such as actual NVM usage according to compressibility; and other techniques. DNA techniques include one or more of: use of non-standard commands or vendor-specific providing unmapped read, write, and/or erase access to the NVM; use of non-standard or vendor-specific commands providing more direct access to the NVM, such as by bypassing formatting of data that the I/O device would otherwise do; and other techniques. Examples of the driver are a driver without ONA or DNA support, an ONA-enabled driver, a DNA-enabled driver, and an ONA/DNA-enabled driver. Further examples of the driver are a vendor-provided, vendor-developed, and/or vendor-enhanced driver, and a client-provided, client-developed, and/or client-enhanced driver.

[0206] Examples of the application-level programs are an application without ONA or DNA support, an ONA-enabled application, a DNA-enabled application, and an ONA/DNA-enabled application. Dotted-arrow **109D** is representative of Application ←→ I/O Device Communication (e.g. bypass via a driver or bypass via a VF for an application), e.g. an ONA-enabled application and an ONA-enabled driver communicating with an SSD, such as without the application using the OS as an intermediary. Dotted-arrow **109V** is representative of Application ←→ I/O Device Communication (e.g. bypass via a VF for an application), e.g. a DNA-enabled application and a DNA-enabled driver communicating with an SSD, such as without the application using the OS or the driver as intermediaries.

[0207] One or more portions of NVM **199** are used, in some embodiments, for firmware storage, e.g. Firmware **106**. The firmware storage includes one or more firmware images (or portions thereof). A firmware image has, for example, one or more images of firmware, executed, e.g., by CPU Core **172** of SSD Controller **100**. A firmware image has, for another example, one or more images of constants, parameter values, and NVM device information, referenced, e.g. by the CPU core during the firmware execution. The images of firmware correspond, e.g., to a current firmware image and zero or more previous (with respect to firmware updates) firmware images. In various embodiments, the firmware provides for generic, standard, ONA, and/or DNA operating modes. In some embodiments, one or more of the firmware operating modes are enabled (e.g. one or more APIs are "unlocked") via keys or various software techniques, optionally communicated and/or provided by a driver.

[0208] In some embodiments lacking the the Switch / Fabric / Intermediate Controller, the SSD is coupled to the host directly via External Interfaces **110**. In various embodiments, SSD Controller **100** is coupled to the host via one or more intermediate levels of other controllers, such as a RAID controller. In some embodiments, SSD **101** (or variations thereof) corresponds to a SAS drive or a SATA drive and Switch / Fabric / Intermediate Controller **103** corresponds to an expander that is in turn coupled to an initiator, or alternatively Switch / Fabric / Intermediate Controller **103** corresponds to a bridge that is indirectly coupled to an initiator via an expander. In some embodiments, Switch / Fabric / Intermediate Controller **103** includes one or more PCIe switches and/or fabrics.

[0209] In various embodiments, such as some of the embodiments where Host **102** is a computing host (e.g. a computer, a workstation computer, a server computer, a storage server, a SAN, a NAS device, a DAS device, a storage appliance, a PC, a laptop computer, a notebook

computer, and/or a netbook computer), the computing host is optionally enabled to communicate (e.g. via optional I/O & Storage Devices/Resources **117** and optional LAN/WAN **119**) with one or more local and/or remote servers (e.g. optional Servers **118**). The communication enables, for example, local and/or remote access, management, and/or usage of any one or more of SSD **101** elements. In some embodiments, the communication is wholly or partially via Ethernet. In some embodiments, the communication is wholly or partially via Fibre Channel. LAN/WAN **119** is representative, in various embodiments, of one or more Local and/or Wide Area Networks, such as any one or more of a network in a server farm, a network coupling server farms, a metro-area network, and the Internet.

[0210] In various embodiments, an SSD controller and/or a computing-host flash memory controller in combination with one or more NVMs are implemented as a non-volatile storage component, such as a USB storage component, a CF storage component, an MMC storage component, an eMMC storage component, a Thunderbolt storage component, a UFS storage component, an SD storage component, a Memory Stick storage component, and an xD-picture card storage component.

[0211] In various embodiments, all or any portions of an SSD controller (or a computing-host flash memory controller), or functions thereof, are implemented in a host that the controller is to be coupled with (e.g., Host **102** of Fig. 1B). In various embodiments, all or any portions of an SSD controller (or a computing-host flash memory controller), or functions thereof, are implemented via hardware (e.g., logic circuitry), software and/or firmware (e.g., driver software or SSD control firmware), or any combination thereof. For example, functionality of or associated with an ECC unit (such as similar to ECC **161** and/or ECC-X **135** of Fig. 1A) is implemented partially via software on a host and partially via a combination of firmware and hardware in an SSD controller. For another example, functionality of or associated with a recycler unit (such as similar to Recycler **151** of Fig. 1A) is implemented partially via software on a host and partially via hardware in a computing-host flash memory controller.

MAPPING OPERATION

[0212] Fig. 2 illustrates selected details of an embodiment of mapping an LPN portion of an LBA. In some embodiments, a read unit is a finest granularity of an NVM that is independently readable, such as a portion of a page of the NVM. In further embodiments, the

read unit corresponds to check bits (sometimes-termed redundancy) of a (lower-level) error-correcting code along with all data protected by the check bits. For example, ECC **161** of Fig. 1A implements error correction via check bits such as via an LDPC code, and a read unit corresponds to coding bits implementing the LDPC code in addition to data bits protected by the LDPC coding bits.

[0213] In some embodiments, Map **141** maps LPN **213** portion of LBA **211** to Map Info for LPN **221**, such as via Table **143** (as illustrated in Fig. 1A). Map info for an LPN (such as Map Info for LPN **221)** is sometimes termed a map entry. Map **141** is said to associate an LPN with a corresponding map entry. In various embodiments, mapping is via one or more associative look-ups, via one or more non-associative look-ups, and/or via one or more other techniques.

[0214] In some embodiments, SSD Controller **100** maintains one map entry for each LPN potentially and/or actively in use.

[0215] In some embodiments, Map Info for LPN **221** includes respective Read Unit Address **223** and Length in Read Units **225**. In some embodiments, a length and/or a span are stored encoded, such as by storing the length as an offset from the span, e.g. in all or any portions of Length in Read Units **225**. In further embodiments, a first LPN is associated with a first map entry, a second LPN (different from the first LPN, but referring to a logical page of a same size as a logical page referred to by the first LPN) is associated with a second map entry, and the respective length in read units of the first map entry is different from the respective length in read units of the second map entry.

[0216] In various embodiments, at a same point in time, a first LPN is associated with a first map entry, a second LPN (different from the first LPN) is associated with a second map entry, and the respective read unit address of the first map entry is the same as the respective read unit address of the second map entry. In further embodiments, data associated with the first LPN and data associated with the second LPN are both stored in a same physical page of a same device in NVM **199**.

[0217] According to various embodiments, Read Unit Address **223** is associated with one or more of: a starting address in the NVM; an ending address in the NVM; an offset of any of the preceding; and any other techniques for identifying a portion of the NVM associated with LPN **213**.

[0218] Fig. 3 illustrates selected details of an embodiment of accessing an NVM at a read unit address to produce read data organized as various read units, collectively having a length measured in quanta of read units. According to various embodiments, First Read Unit **313** is one or more of: a one of read units in Read Data **311** with a lowest address in an address space of the NVM; a fixed one of the read units; an arbitrary one of the read units; a variable one of the read units; and a one of the read units selected by any other technique. In various embodiments, SSD Controller **100** is enabled to access NVM **199** and produce Read Data **311** by reading no more than a number of read units specified by Length in Read Units **225**.

[0219] Fig. 4A illustrates selected details of an embodiment of a read unit (such as Read Units **313** or **315** of Fig. 3) as Read Unit **401A**. In various embodiments and/or usage scenarios, Header 1 **411A** through Header N **419A** are contiguous, and respective data regions identified (such as via respective offsets) by each of the headers are contiguous following a last one of the headers. The data regions collectively form Data Bytes **421A**. The data regions are stored in a location order that matches the location order the headers are stored. For example, consider a first header, at the beginning of a read unit, with a second header and a third header contiguously following the first header. A first data region (identified by a first offset in the first header) contiguously follows the third header. A second data region (identified by a second offset in the second header) contiguously follows the first data region. Similarly, a third data region (identified by the third header) contiguously follows the second data region.

[0220] Fig. 4B illustrates selected details of another embodiment of a read unit (such as Read Units **313** or **315** of Fig. 3) as Read Unit **401B**. In various embodiments and/or usage scenarios, Header Marker (HM) **410B** is an optional initial field (such as a one-byte field) indicating a number of following contiguous headers (Header 1 **411B**, Header 2 **412B** ... Header N **419B**). Data regions (Data Bytes **421B**, Data Bytes **422B** ... Data Bytes **429B**) are identified respectively by the headers (Header 1 **411B**, Header 2 **412B** ... Header N **419B**) and are stored in a location order that is opposite of the location order that the headers are stored. Headers start at the beginning of a read unit, while corresponding data regions start at the end of a read unit. In some embodiments, data bytes within a data region (e.g. Data Bytes **421B**, Data Bytes **422B** ... Data Bytes **429B**) are arranged in a forward order (byte order matching location order), while in other embodiments, the data bytes are arranged in a reverse order (byte order reversed with respect to location order). In some embodiments, a header marker is used in read units where headers and data bytes are stored in a same location order (e.g. as illustrated in Fig. 4A).

[0221] In some embodiments, Optional Padding Bytes **431A** (or **431B**) are according to granularity of data associated with a particular LPN. For example, in some embodiments, if Data Bytes **421A** (or collectively Data Bytes **421B**, Data Bytes **422B** ... Data Bytes **429B**) have less than a fixed amount of remaining space, such as 8 bytes, after storing data associated with all but a last one of Header 1 **411A** through Header N **419A** (or Header 1 **411B**, Header 2 **412B** ... Header N **419B**), then data for an LPN associated with the last header starts in a subsequent read unit. In further embodiments, a particular offset value (e.g. all ones) in the last header indicates that the data for the LPN associated with the last header starts in the subsequent read unit.

[0222] Fig. 5 illustrates selected details of an embodiment of a header (such as any of Header 1 **411A** through Header N **419A** of Fig. 4A or Header 1 **411B** through Header **419B** of Fig. 4B) having a number of fields. In some embodiments, headers are fixed-length (e.g. each header is a same number of bytes long). Header **501** includes fields Type **511**, Last Indicator **513**, Flags **515**, LPN **517**, Length **519**, and Offset **521**. The type field identifies a category of the data bytes. For example, the type field indicates the category of the data bytes is one of host data (e.g. logical page data) or system data (e.g. map information or checkpoint information). The last field indicates that the header is the last header before the data bytes. In some embodiments with a header marker, the last field is optionally omitted. The LPN field is the LPN that the header is associated with. The LPN field enables parsing of the headers to determine a particular one of the headers that is associated with a particular LPN by, for example, searching the headers for one with an LPN field matching the particular LPN. The length field is the length, in bytes, of the data bytes (e.g. how many bytes of data there are in Data Bytes **421A** associated with Header **501**). In some embodiments, an offset in the offset field is rounded according to a particular granularity (e.g. 8-byte granularity).

[0223] In various embodiments, some or all information associated with a particular LPN is stored in a map entry associated with the particular LPN, a header associated with the particular LPN, or both. For example, in some embodiments, some or all of Length **519** is stored in a map entry rather than in a header.

[0224] Fig. 6 illustrates selected details of an embodiment of blocks, pages, and read units of multiple NVM devices. For example, **601**, **603**... and **609** correspond to respective ones of one or more individual Flash Die **194**, of Fig. 1A.

[0225] In some embodiments, Devices **601**, **603**... and **609** are a portion less than all of NVM **199**. For example, in various embodiments, data is striped independently across multiple groups of devices, where each of the groups of devices is independently accessible.

[0226] Each device (such as any one of Devices **601**, **603**... and **609**) provides storage organized as blocks (such as Blocks **631**, **633**... and **639** of Device **601**; Blocks **661**, **663**... and **669** of Device **603**; and so forth). In some embodiments, each device includes an integral number of blocks and a block is a smallest quantum of erasing. The blocks in turn include pages (such as Pages **621**, **623**... and **629** of Block **631**; Pages **651**, **653**... and **659** of Block **661**; and so forth). In some embodiments, each block includes an integral number of pages and a page is a smallest quantum of writing. The pages in turn include read units (such as Read Units **611**, **613**... and **619** of Page **621**; Read Units **641**, **643**... and **649** of Page **651**; and so forth). According to various embodiments, one or more of: a read unit is a smallest quantum of reading and error correction; each page includes an integral number of read units; an associated group of two or more pages includes an integral number of read units; and read units optionally and/or selectively span page boundaries . Reads and/or writes of information in the devices are performed according to an order, such as a 'read unit first' order or a 'page first' order. An example of a read unit first order for read units illustrated in the figure begins with Read Unit **611** followed by **641**, ..., **671**, **613**, **643**, ..., **673**, and so forth, ending with **679**. An example of a page first order for read units illustrated in the figure begins with Read Unit **611** followed by **613**, ..., **619**, **641**, **643**, ..., **649**, **671**, **673**, and so forth, ending with **679**.

[0227] In some embodiments, various NVM management functions are performed in units of R-blocks. An example of an R-block is a logical slice or section across all die of a flash memory. For example, in a flash memory having R flash die, each flash die having N blocks, each R-block is the ith block from each of the flash die taken together, for a total of N R-blocks. For another example, in a flash memory having R flash die, each with N blocks, each R-block is the ith and $(i+1)^{th}$ block from each of the flash die, for a total of N/2 R-blocks. For yet another example, in a flash memory having a plurality of dual plane devices, each R-block is the ith even block and the ith odd block from each of the dual plane devices.

HIGHER-LEVEL REDUNDANCY TECHNIQUES

[0228] Fig. 7A illustrates selected details of various embodiments of higher-level redundancy techniques. Flash Device(s) **720** includes 64 flash die (Flash Die **710.63**, **710.62**,

710.61 ... **710.0**) and communicates via Interface Channel(s) **730**. Extra Flash Device(s) **740** includes up to two flash die (Flash Die **710.65** and **710.64**) and communicates via Extra Interface Channel(s) **750**. The flash die provide storage for higher-level redundancy information and data storage (e.g. user data and/or user free space) in a storage sub-system, such as NVM in an SSD. (Examples of 'user data' in contexts of redundancy information and data storage include all data other than the redundancy information stored on flash memory for later retrieval, such as operating system data, application data, SSD management data, and so forth.) Higher-level redundancy enables, e.g., recovering from intermittent or permanent failure of one or more portions of one or more flash die, such as a failure to provide error-corrected data (e.g. via lower-level ECC functions) for a read operation or failure to properly complete a write operation.

[0229] For example, each flash die (or alternatively each block or each page within each block) is operated in a context of a Redundant Array of Silicon Independent Elements (RASIE). If a failure is detected in a particular flash die (e.g. due to an ECC-uncorrectable read error of a portion of a block of the particular die), then in response, redundant information stored in others of the flash die is used to determine information that would have been provided by the particular die. In some embodiments and/or usage scenarios, sufficient redundant information is stored to enable recovery from one failure within one flash die (during a single operation). Operation in a mode that enables recovery from a single failure includes, in some embodiments, allocating and managing space equivalent to one flash die for higher-level redundancy information, and is termed 'RASIE-1'. Operation in a mode that enables recovery from two failures includes, in some embodiments, allocating and managing space equivalent to two flash die for higher-level redundancy information, and is termed 'RASIE-2'.

[0230] In some embodiments and/or usage scenarios, managing die-level failures is an objective, and spreading information amongst die is performed. For example, higher-level redundancy information is stored in one or more die specifically allocated solely to the higher-level redundancy information. In some embodiments and/or usage scenarios, managing block-level failures is an objective, and spreading information amongst blocks within a die is performed. For example, higher-level redundancy information is stored in one or more blocks allocated to the higher-level redundancy information, the allocation being without regard to which particular die the blocks were part of. In some embodiments and/or usage scenarios, managing particular-entity-level failures includes spreading information so that no more than N elements (e.g. 1 for RASIE-1 and 2 for RASIE-2) are in any one of the particular entities.

[0231] The higher-level redundancy information is computed and written in accordance with (user) data written to the flash die, and is thus available to provide information when a failure is detected. In various embodiments, the higher-level redundancy information is written to the flash die before, after, or in no particular time order with respect to writing of (user) data the higher-level redundancy information is associated with.

[0232] The figure illustrates various embodiments of RASIE operating modes, as summarized in the following table.

Usage Mode		Usage of Extra Flash Device(s) 740	Usage of Flash Device(s) 720
RASIE-1			
	1-0	-none-	**710.63** (1 die) – redundancy **710.62** ... **710.0** (63 die) – data storage
	1-1	**710.64** (1 die) – redundancy	**710.63** ... **710.0** (64 die) – data storage
RASIE-2			
	2-0	-none-	**710.63**, **710.62** (2 die) – redundancy **710.61** ... **710.0** (62 die) – data storage
	2-1	**710.64** (1 die) – redundancy	**710.63** (1 die) – redundancy **710.62** ... **710.0** (63 die) – data storage
	2-2	**710.65**, **710.64** (2 die) – redundancy	**710.63** ... **710.0** (64 die) – data storage

[0233] More specifically, in RASIE-1 modes, space equivalent to one die is allocated to higher-level redundancy information. In RASIE-1 mode 1-0, Extra Flash Device(s) **740** is not used, as higher-level redundancy information is stored in one die of Flash Device(s) **720** (e.g. Flash Die **710.63**), leaving 63 of the die (Flash Die **710.62** ... **710.0**) available for data storage (e.g. user data and/or user free space). In RASIE-1 mode 1-1, one die of Extra Flash Device(s) **740** is used (e.g. Flash Die **710.64**), leaving all of Flash Device(s) **720** (64 die) available for data storage.

[0234] In RASIE-2 modes, space equivalent to two die is allocated to higher-level redundancy information. In RASIE-2 mode 2-0, Extra Flash Device(s) **740** is not used, as higher-level redundancy information is stored in two die of Flash Device(s) **720** (e.g. Flash Die **710.63** and Flash Die **710.62**), leaving 62 of the die (Flash Die **710.61** ... **710.0**) available for data storage. In RASIE-2 mode 2-1, one die of Extra Flash Device(s) **740** is used (e.g., Flash Die **710.64**), as higher-level redundancy information is partially stored in one die of Flash

Device(s) **720** (e.g. Flash Die **710.63**), leaving 63 of the die (Flash Die **710.62** ... **710.0**) available for data storage. In RASIE-2 mode 2-2, two die of Extra Flash Device(s) **740** are used (e.g. Flash Die **710.65** and Flash Die **710.64**), leaving all of Flash Device(s) **720** (64 die) available for data storage.

[0235] In some embodiments, die that are unused in all usage scenarios are unpopulated. For example, in a system operable only in RASIE 2-0 mode and RASIE 1-0 mode (but not in other RASIE modes), Extra Flash Device(s) **740** is unpopulated.

[0236] In some embodiments, higher-level redundancy information is stored entirely in "dedicated" die (e.g. flash die **710.63** in RASIE-1 mode 1-0 or Flash Die **710.65** and Flash Die **710.64** in RASIE-2 mode 2-2). In other embodiments, higher-level redundancy information is stored in any of the die, so for example, in RASIE-1 mode 1-0 Flash Die **710.62** is used for higher-level redundancy information, while Flash Die **710.63** and Flash Die **710.61** ... **710.0** are used for data storage. In some embodiments and/or usage scenarios, higher-level redundancy information is stored in different die (and/or portions thereof) over time, so, for example, in a first time period a first flash die holds higher-level redundancy information while in a second time period a second flash die hold higher-level redundancy information.

[0237] In various embodiments, there are a plurality of RASIE 1-0 modes (and a plurality of RASIE 2-0 modes) depending on how many flash die are usable. For example, in a first RASIE 1-0 mode (as illustrated in the table above), Flash Die **710.63** stores higher-level redundancy information, and Flash Die **710.62** ... **710.0** are available for data storage. In a second RASIE 1-0 mode, Flash Die **710.63** is no longer usable, Flash Die **710.62** stores higher-level redundancy information, and Flash Die **710.61** ... **710.0** are available for data storage, decreasing an available amount of data storage by one die. RASIE modes where a die (or any portions thereof) previously available for data storage is no longer available for data storage due to use of the die (or the portions) for higher-level redundancy information are sometimes referred to as reduced-capacity RASIE modes.

[0238] In some embodiments, higher-level redundancy information is stored using a same and/or a similar lower-level error correction coding scheme as user data protected by the higher-level redundancy information. Using a lower-level error correction scheme to protect the higher-level redundancy information enables determining if there is an uncorrectable error in the higher-level redundancy information, in a same and/or a similar manner that an uncorrectable lower-level error in the user data is determined.

[0239] Fig. 7B illustrates selected details of an embodiment of R-blocks. Flash die are managed (e.g. for recycling and/or erasing) in logical slices or sections across all die, and the slices/sections are referred to as R-blocks. The figure illustrates an embodiment with 66 flash die (Flash Die **710.65** ... **710.0**). Blocks, pages, and management of the blocks as R-blocks are illustrated. Each flash die is illustrated as having N blocks (such as Block **710.65B0**, Block **710.65B1** ... Block **710.65BB** of Flash Die **710.65**). Each R-block is the ith block from each of the flash die, taken together (such as R-block **760.0** being Block **710.65B0** from Flash Die **710.65**, block 0 from Flash Die **710.64** (not explicitly illustrated), and so forth to Block **710.0B0** of Flash Die **710.0**. There are thus a total of N R-blocks (R-block **760.0**, R-block **760.1** ... R-block **760.R**).

[0240] In various embodiments, a writing and/or a striping order of data within an R-block is page (e.g. lowest to highest) first, across all devices (e.g. lowest to highest numbered devices, as suggested conceptually by Striping Direction **600** of Fig. 6), then the next highest page (across all devices), and so forth, continuing throughout the last page of the R-block. Specifically with respect to R-block **760.0** of Fig. 7, an example order begins with Page **710.0P0**, followed by the first page in the first block of Flash Die **710.1**, and so forth continuing to Page **710.65P0**. The example order continues with Page **710.0P1**, followed by the second page in the first block of Flash Die **710.1**, and so forth continuing to Page **710.65P1**. The example order completes with Page **710.0PP**, followed by the last page in the first block of Flash Die **710.1**, and so forth ending with Page **710.65PP**.

[0241] Another example of an R-block is the ith and $(i+1)^{th}$ block from each of the flash die, taken together (e.g. blocks 0 and 1 from Flash Die **710.65**, blocks 0 and 1 from Flash Die **710.64**, and so forth to blocks 0 and 1 from Flash Die **710.0**). There are thus N/2 R-blocks, if there are N blocks in each flash die. Yet another example of an R-block is the ith even and odd blocks from each of a plurality of dual plane devices. Other arrangements of flash die blocks for management as R-blocks are contemplated, including mapping between virtual and physical block addresses to ensure that R-blocks have one block from each die, even if some blocks are inoperable. In various embodiments, some of the N blocks in each flash die are used as spares so that the mapping between virtual and physical block addresses has spare (otherwise unused) blocks to replace defective ones of the blocks in the R-blocks.

[0242] In various embodiments that blocks are treated in pairs or other associated groups as part of forming an R-block, respective pages from each block of an associated group

of the blocks are also treated as a unit, at least for writing, forming a larger multi-block page. For example, continuing the dual plane example above, a first page of a particular one of the even blocks and a first page of an associated one of the odd blocks are treated as a unit for writing, and optionally and/or selectively as a unit for reading. Similarly, a second page of the particular even block and a second page of the associated odd block are treated as a unit. According to various embodiments, a page of NVM as used herein refers to one or more of: a single page of NVM; a multi-block page of NVM; a multi-block page of NVM for writing that is optionally and/or selectively treated as one or more individual pages for reading; and any other grouping or association of pages of NVM.

[0243] In some embodiments, higher-level redundancy information is stored in different die for different portions of data. For instance, in some embodiments where flash die are managed in R-blocks, higher-level redundancy information is stored in different flash die for different R-blocks. For example, higher-level redundancy information for an R-block including block 0 of Flash Die **710.0** is stored in Flash Die **710.0**, while higher-level redundancy information for an R-block including block 1 of Flash Die **710.0** is stored in Flash Die **710.1**, and so forth. In some embodiments, such as some embodiments where flash die are managed in R-blocks, higher-level redundancy information is written after data the higher-level redundancy information depends on is known and/or is written.

[0244] In some usage scenarios, one or more portions of an NVM element (e.g. a block of a device, such as Block **639** of Device **601** of Fig. 6) are, or become during operation, inoperable. In some embodiments, the inoperable portions are mapped out via virtual and physical block addresses (e.g. via processing performed via Map **141** and/or Table **143** of Fig. 1A.). Alternatively, the inoperable portions are skipped (rather than explicitly mapped out). In some embodiments based on R-blocks, the skipping results in some of the R-blocks having differing numbers of blocks. For example, if Block **710.0B0** is defective and unusable, then R-block **760.0** has one fewer block than R-block **760.1**. The higher-level redundancy information is written in a (per R-block) variable location that is, for example, the last block of each R-block.

[0245] In various embodiments, one or more elements of Fig. 7A correspond to one or more elements of Fig. 1A. For example, Flash Device(s) **720** and Extra Flash Device(s) **740** collectively correspond to NVM **199**, and Interface Channel(s) **730** and Extra Interface Channel(s) **750** collectively correspond to Device Interfaces **190**. For another example, Flash Die **710.65** ... **710.0** collectively correspond to the instances of Flash Die **194**. For yet another example, one or more of the flash devices of Flash Device(s) **720** and/or Extra Flash Device(s)

740 correspond to one or more of the instances of Flash Devices **192**. In various embodiments, one or more elements of Fig. 1A manage higher-level redundancy information and/or recover user data based at least in part on the higher-level redundancy information in accordance with the RASIE operating modes described herein. For example, a portion of software execution capabilities of CPU **171** is used to manage computation of higher-level redundancy information according to various RASIE operating modes. For another example, Data Processing **121** and/or ECC-X **135** includes hardware elements dedicated to and/or specialized for computation of higher-level redundancy information and/or recovery of user data according to various RASIE operating modes. For yet another example, ECC **161** detects an ECC-uncorrectable (lower-level) read error of a portion of a flash die, and ECC-X **135** detects a RASIE (higher-level) read error and/or enables correction thereof.

[0246] In various embodiments, Interface Channel(s) **730** variously has one, four, eight, or 16 channels, and Extra Interface Channel(s) **750** variously has one or two channels. In various embodiments, Flash Device(s) **720** is implemented as one, two, four, eight, or 16 devices, each having respectively 64, 32, 16, eight, and four of the flash die. In various embodiments, Extra Flash Device(s) **740** is implemented as one device having one or two die or as two devices each having one die. In some embodiments, the flash die of Extra Flash Device(s) **740** are implemented in devices that also implement the flash die of Flash Device(s) **720**. For example, one flash device implements 66 flash die (Flash Die **710.65** ... **710.0**). For another example, two flash devices each implement 33 flash die, e.g. in a first flash device (Flash Die **710.65** ... **710.33**) and in a second flash device (Flash Die **710.32** ... **710.0**). Other arrangements of flash die and flash devices are contemplated. In some embodiments having Extra Flash Device(s) **740** implemented in devices that also implement flash die of Flash Device(s) **720**, the flash die communicate via shared interface channels, or alternatively via interface channels dedicated to particular ones (or sets) of the flash die. While Flash Device(s) **720** and Extra Flash Device(s) **740** are illustrated with specific numbers of flash die (2 and 64, respectively), other embodiments are contemplated, such as Flash Device(s) **720** having 2, 4, 8, 16, 32, or 128 flash die, and/or Extra Flash Device(s) **740** having 0, 1, or 4 flash die.

[0247] Fig. 8 illustrates selected details of an embodiment of dynamic higher-level redundancy mode management with RASIE, such as dynamically switching between RASIE modes enabled by the various embodiments illustrated by Fig. 7A and/or Fig. 7B. In some embodiments and/or usage scenarios, a form of graceful degradation is provided where a storage sub-system (e.g. an SSD) is dynamically transitioned from operating in first higher-level redundancy mode to operating in a second higher-level redundancy mode. The transition is in

response to detection of a failure, such as a permanent or intermittent malfunction of an entire flash die or one or more portions thereof, or an operation (such as a read or write operation) thereto. According to various embodiments, the transition is one or more of: global for the SSD; performed on one or more subsets of the SSD; and performed on one or more R-blocks of the SSD. For example, if a particular block of one of the NVM devices storing RASIE-2 information fails during programming, then subsequent operation of the R-block containing the particular (failed) block transitions to a different higher-level redundancy mode (e.g. a RASIE-1 mode), whereas other R-blocks in the SSD are unaffected and continue to operate in the RASIE-2 mode.

[0248] With respect to Fig. 8, processing begins with higher-level redundancy information and data storage (e.g. user data and/or user free space) arranged in flash die in accordance with a first higher-level redundancy mode (Operate in First Higher-Level Redundancy Mode **802**). Flow then proceeds to determine if a failure has been detected (Failure? **803**), such as a lower-level uncorrectable read error or a write/program failure. If no failure has been detected, then flow proceeds back to continue operation in the first higher-level redundancy mode. If a failure has been detected, then flow proceeds to switch from operating in the first higher-level redundancy mode to operating in a second higher-level redundancy mode. The switch begins by (optionally) decreasing space available for data storage (Reduce Free Space **804**) to account for the failure. If the second higher-level redundancy mode uses sufficiently less higher-level redundancy information than the first higher-level redundancy mode, then the decreasing of available space is omitted. The switch continues by reorganizing data storage in accordance with the second higher-level redundancy mode (Rearrange Data Storage **805**). The reorganizing includes optionally moving all user data and/or user free space from the flash die where the failure occurred to another one of the flash die (user free space movement is accomplished, in some embodiments, by manipulation of pointers and/or other data structure elements). The switch further continues by selectively restoring (if possible), via the higher-level redundancy information of the first higher-level redundancy mode, any user data that was stored in the flash die where the failure occurred, and writing the restored user data to another one of the flash die, in accordance with the second higher-level redundancy mode (Recover/Store Failed User Data **806**). The restoring is omitted if the failure is a write/program failure. The switch further continues by optionally computing and writing to the flash die higher-level redundancy information in accordance with the second higher-level redundancy mode (Determine/Store Revised Higher-Level Redundancy Information **807**). The computing and the writing are omitted if the second higher-level redundancy mode is operable with higher-level redundancy information that was previously in place due to operating in the first higher-

level redundancy mode. Then operation begins in the second higher-level redundancy mode (Operate in Second Higher-Level Redundancy Mode **808**).

[0249] The failure detection (Failure? **803**) is via one or more of: lower-level error correction (e.g. in accordance with one or more ECC techniques), higher-level error correction (e.g. in accordance with one or more RASIE techniques), and failing status reported by one or more of the flash die or portions thereof. For example, more than a threshold number of lower-level corrections of reads within a particular portion of a particular flash die optionally and/or conditionally results in the particular flash die (or the particular portion) being treated as failed and a higher-level redundancy mode switch is performed so that the failed flash die (or portion) is no longer used. For another example, if a higher-level error correction fails, then an appropriate one of the flash die (or portion thereof) is treated as failed and a higher-level redundancy mode switch is performed so that the failed flash die (or portion) is no longer used. For yet another example, if a flash die returns a program failure status (indicating that a write operation was unsuccessful), then an appropriate block of an appropriate one of the flash die is treated as failed, and optionally and/or conditionally a higher-level redundancy mode switch is performed so that the failed flash die is no longer used.

[0250] In some embodiments, a failed block is replaced by remapping via virtual and physical block addresses (e.g. via processing performed via Map **141** and/or Table **143** of Fig. 1A). A spare block from a pool of spare blocks is mapped in place of the failed block. Any contents written in the failed block are copied to the replacement block, and writing proceeds in the spare block from where the failure occurred in the failed block.

[0251] In some embodiments, a failed block is skipped (rather than explicitly remapped), resulting in a "hole" that optionally and/or conditionally results in a higher-level redundancy mode switch when the R-block that the hole is in is next erased (in preparation for re-writing). If the hole is in a location for data storage, then no switch is made, and the hole remains. If the hole is in a location for higher-level redundancy information, then the higher-level redundancy information is stored in another location, and optionally the higher-level redundancy mode is switched.

[0252] In some embodiments and/or usage scenarios, restoration of user data stored in the flash die where the failure occurred is not possible. For example, if the failure is due to some types of failures detected via higher-level error correction and/or some types of failing status reported by one or more of the flash die or portions thereof, then some user data may be lost.

[0253] In some embodiments, processing of Fig. 8 is performed in a context of (e.g. an SSD controller) dynamically transitioning between higher-level redundancy modes in response to a plurality of failures. Specifically, the SSD controller begins operating in a first higher-level redundancy mode and dynamically transitions to a second higher-level redundancy mode in response to a first failure, and subsequently dynamically transitions from the second higher-level redundancy mode to a third higher-level redundancy mode in response to a second failure, and so forth. For instance, an SSD controller operates various flash die in accordance with a RASIE-2 mode 2-2 and dynamically transitions the operation to be in accordance with a RASIE-2 mode 2-1 in response to a first failure. Subsequently, the SSD controller dynamically transitions the operation to be in accordance with a RASIE-2 mode 2-0 in response to a second failure. Further subsequently, the SSD controller dynamically transitions the operation to be in accordance with a RASIE-1 reduced-capacity mode 1-0 in response to a third failure (the reduced-capacity mode 1-0 being similar to RASIE-1 mode 1-0 except with one flash die used for higher-level redundancy information and 62 flash die used for data storage).

[0254] As a specific example, consider an SSD controller (such as SSD Controller **100** of Fig. 1A) coupled to the elements of Fig. 7A, initially operating in RASIE-2 mode 2-2 (e.g. higher-level redundancy information in Flash Die **710.65** and Flash Die **710.64**, and data storage in Flash Die **710.63** ... **710.0**), corresponding to operating in the first higher-level redundancy mode. Then a read, or alternatively a write, of one or more of the flash die is performed. The read results in an uncorrectable (lower-level) ECC failure, or alternatively the write is unsuccessful, in a portion of a particular one of the flash die (e.g. a page of Flash Die **710.62** used for user data and/or user free space). In response, the SSD controller dynamically switches from operating in RASIE-2 mode 2-2 to RASIE-2 mode 2-1, no longer using any of Flash Die **710.62**. As operation in RASIE-2 mode 2-1 provides 63 die for data storage (versus 64 die in RASIE-2 mode 2-2), space available for data storage is decreased from 64 die to 63 die, and user data and/or user free space is moved accordingly. E.g., all user data from Flash Die **710.62** is moved to portions of Flash Die **710.63** and Flash Die **710.61** ... **710.0** in accordance with user free space. Any user data in the page having the uncorrectable ECC failure is recovered based on the higher-level redundancy information in Flash Die **710.65** and/or Flash Die **710.64**. Higher-level redundancy information based on data storage in Flash Die **710.63** and Flash Die **710.61** ... **710.0** and in accordance with RASIE-2 mode 2-1 is computed and stored in Flash Die **710.65** and/or Flash Die **710.64**. The SSD controller then operates in RASIE-2 mode 2-1 (higher-level redundancy information in Flash Die **710.65** and flash die **710.64**, and data storage in Flash Die **710.63** and Flash Die **710.61** ... **710.0**).

[0255] While the foregoing described several embodiments of dynamic higher-level redundancy mode management with independent silicon elements of a granularity of an entire flash die, other embodiments implement dynamic higher-level redundancy mode management with independent silicon elements that are portions of a die, such as one or more pages, blocks, or R-blocks.

[0256] In various embodiments, processing of or relating to one or more elements of Fig. 8 is performed entirely or in part by one or more elements (or portions thereof) of Fig. 1A. For example, a portion of software execution capabilities of CPU **171** is used to manage dynamic transitioning between higher-level redundancy modes, such as by directing decreasing space available for data storage or directing reorganizing data storage. For another example, Data Processing **121** and/or ECC-X **135** includes hardware elements dedicated to and/or specialized for computation of higher-level redundancy information in accordance with a 'target' redundancy mode. For yet another example, ECC **161** implements lower-level (e.g. ECC) error correction and detection of uncorrectable errors, while ECC-X **135** implements higher-level (e.g. RASIE) error correction and detection of uncorrectable errors and/or memory element failures. For another example, all or any portions of functionality relating to dynamic transitioning between (higher-level) redundancy modes is performed by one or more portions of ECC-X **135**.

HIGHER-LEVEL REDUNDANCY AND ADAPTIVE LOWER-LEVEL CODE RATES

[0257] In some embodiments and/or usage scenarios, lower-level error correction uses an adaptive code rate (e.g. an adaptive ECC technique using a variable code rate). For instance, a first read unit is managed with a first code rate that provides relatively more usable data bits than a second read unit that is managed with a second code rate. In some embodiments and/or usage scenarios with lower-level error correction using a variable code rate, higher-level redundancy information is stored in portions of independent silicon elements (such as portions of flash die) that are managed with lower-level error correction that provides relatively more usable data bits or the most usable data bits with respect to data storage protected by the higher-level redundancy information. The portion(s) where the higher-level redundancy information is stored varies, in various embodiments and/or usage scenarios, on a per R-block basis, on a per-die basis, dynamically over time, or any combination thereof. In various embodiments, one or more die, R-blocks, blocks, and/or pages are selected for storage of higher-level redundancy data based on having the most usable data with respect to lower-level error correction code rate.

[0258] For example, higher-level redundancy information is stored at varying locations (such as block locations) on a per R-block basis, the locations (such as blocks) being those having the most usable data based on lower-level error correction code rate. In an arbitrary example, consider a RASIE-2 operating mode scenario in the context of Fig. 7A, wherein the collection of 66 flash die is treated as a logical "stack" of up to **M** R-blocks of one-block height each, where **M** is the number of blocks per flash die. (In the most conceptually straightforward case, each R-block in the stack is made up of the same physical block number from each die, but to accommodate failed blocks this constraint is lifted in at least some embodiments. In yet other embodiments, the constraint is maintained, but "holes" are accommodated corresponding to the failed blocks.) Thus, each R-block has up to 66 blocks, corresponding to one block from each of Flash Die **710.0** through Flash Die **710.65**. (In Fig. 7A, while only some flash die in the range **710.0** through **710.65** are explicitly enumerated, it is understood by the use of ellipsis that all flash die in this range are implicitly illustrated.) The higher-level redundancy information is written into whichever blocks of each R-block have the most useable data based on lower-level correction code rate. If for example in a first R-block, blocks corresponding to Flash Die **710.15** and **710.49** happen to have the most usable data based on lower-level error correction code rate, then higher-level redundancy information is written into those blocks (corresponding to Flash Die **710.15** and **710.49**). While if in a second R-block, blocks corresponding to flash die **710.9** and **710.35** happen to have the most usable data based on lower-level error correction code rate, then higher-level redundancy information is written into those blocks (corresponding to Flash Die **710.9** and **710.35**). In some embodiments, the higher-level redundancy data is written after all other data in an R-block is known.

[0259] Fig. 9 illustrates an embodiment of Read Units (**911**, **931** … **951**, **971**) having lower-level redundancy information of adaptive (e.g. differing and/or varying over time and/or by location) code rates protected by higher-level redundancy information stored in one or more of the read units. Each of the read units has a portion enabled and/or allocated to contain User Data (**911.U**, **931.U** … **951.U**, **971.U**), and a remaining portion enabled and/or allocated to contain lower-level redundancy information, such as check bits of an ECC code as Lower-Level ECC (**911.E**, **931.E** … **951.E**, **971.E**).

[0260] In the figure, vertical dimensions are relatively to scale and indicative of relative size. Thus Read Units **911** and **951** are of a same first size (in some embodiments, all read units are a same size for all blocks of all NVM devices), Lower-Level ECC portions **931.E** and **951.E** are of a same second size, and Lower-Level ECC portions **911.E** and **971.E** are of a same third

size. Read Unit **931** is larger than Read Units **911** and **951** that are in turn larger than Read Unit **971**. User Data portion **931.U** is larger than User Data portion **951.U**. User Data portion **951.U** is larger than User Data portion **911.U**. Lower-Level ECC portion **951.E** is smaller than Lower-Level ECC portion **911.E**.

[0261] As illustrated in the figure, respective read units have respective sizes, e.g. per respective block of NVM, enabling varying lower-level code rates as used in the respective blocks. More specifically, Read Units **931** and **951** have a same amount of lower-level redundancy information (**931.E** and **951.E** are a same size), but a lower-level code rate of Read Unit **931** is higher than a lower-level code rate of Read Unit **951**, as Read Unit **931** contains more User Data (**931.U**) than Read Unit **951** (containing User Data **951.U**).

[0262] As illustrated in the figure, respective read units have respective sizes of user data, enabling various sizes of user data in each of two or more read units of a same size. For example, a size of user data is varied to change a lower-level code rate used in a particular read unit. More specifically, Read Units **951** and **911** have a same size, but have different respective amounts of User Data (**951.U** and **911.U**), and thus different respective amounts of lower-level redundancy information (**951.E** and **911.E**), enabling Read Unit **951** to have a higher lower-level code rate than Read Unit **911**.

[0263] In some embodiments and/or usage scenarios, varying and/or changing a lower-level code rate advantageously enables providing a sufficient amount of lower-level ECC information to achieve lower-level error correction requirements while maximizing an amount of user data.

[0264] In some embodiments with a varying amount of user data in read units, higher-level redundancy information is stored in a one or more of the read units having a largest amount of user data. For example in Fig. 9, using a RASIE-1 mode, higher-level redundancy information is stored in User Data **931.U**, and using a RASIE-2 mode, higher-level redundancy information is stored in User Data **931.U** and User Data **951.U**. Storing the higher-level redundancy information in read units with the largest amount of user data (among the read units protected by the higher-level redundancy information) ensures that there is sufficient higher-level redundancy information to protect the user data in all of the other read units.

[0265] According to various embodiments, one or more techniques are used to determine which of one or more read units among a number of read units protected by higher-

level redundancy information are used to store higher-level redundancy information. In a first example, the latest-written one (for RASIE-1) or two (for RASIE-2) read units that have a largest amount of user data are used. In a second example, the earliest-written one (for RASIE-1) or two (for RASIE-2) read units that have a largest amount of user data are used. Similarly, any technique to deterministically select one or more read units having a largest amount of user data so as to protect all of the remaining user data in the other read units is within the scope of the techniques considered herein.

HIGHER-LEVEL REDUNDANCY INFORMATION COMPUTATION TECHNIQUES

[0266] In various embodiments and/or usage scenarios, higher-level redundancy information is computed with a variety of techniques, such as via parity, RS, and/or weighted-sum techniques. For example, in some higher-level redundancy modes enabling recovery from one (lower-level) failure (e.g. RASIE-1), higher-level redundancy information is computed via parity techniques. For another example, in some higher-level redundancy modes enabling recovery from two (lower-level) failures (e.g. RASIE-2), higher-level redundancy information is computed via a combination of parity and RS techniques. A first portion of the higher-level redundancy information is computed using parity coding and a second portion is computing using RS coding. For yet another example, in some higher-level redundancy modes enabling recovery from two failures (e.g. RASIE-2), higher-level redundancy information is computed via a combination of parity and weighted-sum techniques. A first portion of the higher-level redundancy information is computed using parity coding and a second portion is computing using weighted-sum coding. The higher-level redundancy information is managed, e.g. via reads and writes of pages of NVM, using lower-level failure detection techniques (such as ECC) identical to or similar to lower-level failure detection techniques used for pages of the NVM available for storing data protected by the higher-level redundancy information.

[0267] As a specific example for RASIE-2, a first page of higher-level redundancy information is computed using parity coding via an XOR of all first pages in a stripe across an R-block. More specifically, an XOR is performed of all of the first bytes of all of the first pages in the stripe across the R-block, resulting in a first byte of the first page of higher-level redundancy information. Similarly, a second byte of higher-level redundancy information is formed by XORing all of the second bytes of all of the first pages in the stripe across the R-block, and so forth for all of the bytes of all of the first pages in the stripe. A second page of higher-level redundancy information is computed using a weighted-sum technique as follows.

[0268] Arithmetic is performed over a finite field, such as a Galois Field (used as an example). Examples assume data being operated on is byte-wide, and thus a field such as GF(256) is used. In various embodiments, data is operated on in any units.

[0269] Each page in a stripe is assigned a unique non-zero "index". The values of the indices are chosen to simplify implementation complexity, and are not related to any form of generator polynomial. For example, the pages are labeled (e.g. by software) by die location in a stripe from 0 to N-1, and a suitable value for the indices is the ones-complement of the die number (ensured to be non-zero provided $N < 255$). Another selection of index values is the lowest-weight (fewest number of set bits or fewest number of clear bits) non-zero integers, e.g. to reduce and/or minimize hardware costs. Selecting gray-coded values for the indices, in some embodiments and/or usage scenarios, minimizes transitions and/or reduces power as pages are processed.

[0270] The index values are not selected according to finite field arithmetic, but are selected according to other principles. Notwithstanding this, each index value corresponds to a non-zero element in the finite field. Assume that page i has index value Ki (and page j has index value Kj). The weighted-sum redundancy is the (GF field) sum (over corresponding bytes Pi from each page i) of Ki*Pi, each byte multiplied (over the GF field) by its index value.

[0271] Thus weighted-sum redundancy information is computed for each byte as:

R0 = sum over all corresponding bytes Pi;

R1 = sum over all corresponding bytes Ki*Pi;

R0 is the XOR of all the corresponding bytes; and

R1 is a weighted sum of the bytes, where the weights are selected as the index values.

The foregoing computation is iterated for each of the corresponding bytes in a page, producing corresponding pages of bytes for each of R0 and R1. In the following discussion, R0 and R1 are described in some contexts respectively as single elements (e.g. each being a single byte) for clarity of exposition, but as in the foregoing computation, each are representative of a respective page of elements (e.g. each being a page of bytes).

[0272] Pi represents a byte in page i, and Pj represents a byte in page j. Processing is described with respect to one stripe of corresponding bytes from each page, and iteration is performed over all corresponding bytes. If some pages are "shorter" than others, due to for example to having a different (lower-level) redundancy code rate, then the shorter pages are

zero-padded (or padded by any known value used the same way on encode and decode) so that all pages that are processed effectively have the same size.

[0273] Summations to compute R0 and R1 are performable in any order, via various serial and/or parallel computations, according to various embodiments. Pages do not have to be processed in any specific order as whether Ki*Pi is added in prior to or subsequent to Kj*Pj has no effect on the result in R1. Computation of R0 and R1 values corresponding to various bytes of a page of redundancy information are independent of each other and are computable in any order, via various serial and/or parallel computations, according to various embodiments. Further, subtracting Ki*Pi from R1 (and subtracting Pi from R0) enables "backing out" of computation effects on pages. Since over a GF field, addition and subtraction are both XOR (thus subtracting is equivalent to simply adding in a second time), in some embodiments and/or usage scenarios, no special hardware is needed for GF field implementations (e.g. a logical XOR capability is sufficient) to "back out" a page.

[0274] In the event of an uncorrectable lower-level error, higher-level correction begins, in some embodiments, by re-computing R0 and R1, but by omitting page(s) (sometimes referred to as column(s)) that have uncorrectable lower-level errors. Correction proceeds by subtracting the recomputed R0 from the original R0 to produce ΔR0, and subtracting the recomputed R1 from the original R1 to produce ΔR1.

[0275] If there are no uncorrectable lower-level errors, then the recomputed R0 and R1 are both zero. If there are uncorrectable lower-level errors, then the recomputed R0 and R1 (after the subtraction) reflect the "missing" data (that was not added in the second time, but was present in the original values).

[0276] If there is one uncorrectable lower-level error, then the recomputed R0 is used to correct the error (and the recomputed R1 is not needed).

[0277] If there are two uncorrectable lower-level errors, then the recomputed R0 and R1 are used to correct the errors. If both the pages of R0 and R1 values are the pages with uncorrectable lower-level errors, then no correction of data storage pages is needed. If the page of R1 values is one of the pages with uncorrectable lower-level errors, then correction is via R0 (the recomputed R0 value is the value of the data storage page with uncorrectable lower-level errors).

[0278] If there are two uncorrectable lower-level errors in data storage pages, or if the R0 page is one of the pages with uncorrectable lower-level errors, then correction starts by computing ΔR0 and ΔR1 as above. If the R0 page is one of the pages with uncorrectable lower-level errors, computation of the ΔR0 page is optionally omitted. If page i and page j are pages with uncorrectable lower-level errors, then the recomputed ΔR0 = Pi + Pj, and the recomputed ΔR1 = Ki*Pi + Kj*Pj. Equation solving produces:

Pi = (ΔR1 – Kj*ΔR0)/(Ki – Kj)

Pj = ΔR0 - Pi

If R0 is one of the pages with uncorrectable lower-level errors, then (because R0 is not included in R1), ΔR1 = Ki*Pi, or Pi = ΔR1/Ki; the same result obtained by setting Kj = 0 in the formulas above (to ignore ΔR0).

[0279] In an alternative embodiment, a finite field defined by integers mod p, where p is a prime, is used instead of a Galois Field. The computations are identical to those described above, with addition being integer addition mod p, and multiplication being integer multiplication mod p. For example, if pages are pages of bytes, a finite field of integers mod 257 is used. All user data bytes are in a range 0 – 255 and are stored in one byte each. R1 results, however, have values ranging from 0 – 256, requiring more than one byte for representation. There are many ways to encode values from 0 – 256 to minimize storage space and enable the R1 page to be stored with reduced overhead. For example, values 0 and 256 are stored as nine-bit sequences 000000000 and 000000001 respectively, and all other values are stored in eight bits. Given a random distribution of R1 values, storage overhead is < 0.1%. As described with respect to Fig. 9, the R1 page is selected to have a largest amount of user data, enabling the storage overhead to be hidden in some usage scenarios.

[0280] Fig. 10 illustrates selected details of an embodiment of higher-level redundancy information result and data source correspondences, for instance as used by RASIE-2 mode 2-2 in a context such as Fig. 7A and as further illustrated in Fig. 7B. Fig. 10 explicitly illustrates Flash Die **710.0**, **710.63**, **710.64** and, **710.65**, and by ellipsis (…) implicitly illustrates Flash Die **710.1** … **710.62**. Selected details of blocks, pages, and bytes within pages of the flash die are illustrated. A first portion of higher-level redundancy information is illustrated as R0 **1010** (stored in Flash Die **710.64**), and is computed using parity coding. A second portion of higher-level redundancy information is illustrated as R1 **1011** (stored in Flash Die **710.65**), and is computed using weighted-sum coding. Storage for data information is illustrated as Data **1019** (stored in Flash Die **710.0** … **710.63**).

[0281] Dashed-arrow **1001** conceptually indicates a two-byte redundancy computation result (one byte for each of R0 **1010** and R1 **1011**) based on corresponding first bytes of all first pages (across all flash die) of Data **1019**. As illustrated, the two-byte result is the first byte on each of the first R0 and R1 pages. Dashed arrow **1002** conceptually indicates a two-byte redundancy computation result (one byte for each of R0 **1010** and R1 **1011**) based on corresponding last bytes of all first pages of Data **1019**. As illustrated, the two-byte result is the last byte on each of the first R0 and R1 pages. Dashed-arrow **1003** conceptually indicates a two-page redundancy computation result (one page for each of R0 **1010** and R1 **1011**) based on corresponding last pages of the first blocks (across all flash die) of Data **1019**.

[0282] Note that as described elsewhere herein, in some embodiments higher-level redundancy information is stored in different die for different portions of data. Thus, R0 and R1, in some embodiments, are stored across various die, rather than two "dedicated" die, as illustrated in Fig. 10.

[0283] Fig. 11 illustrates selected details of an embodiment of higher-level redundancy information computations, for instance as used by RASIE-2 mode 2-2 in a context such as Fig. 7A and as further illustrated in Fig. 7B and Fig. 10, with various operating conditions according to various embodiments. More specifically, Fig. 11 illustrates parity coding computation for a byte of R0 and weighted-sum coding computation for a byte of R1, in accordance with, e.g., the two-byte redundancy computation result illustrated conceptually by dashed-arrow **1001** of Fig. 10. The operating conditions include one or more of: performing arithmetic over a finite field (such as a Galois Field), indices (corresponding to dummy summation variable "i" values in Fig. 11) being ones-complement of flash die number (or any other unique and non-zero numbering), and indices corresponding to non-zero elements in the finite field. The computation as illustrated in Fig. 11 is repeated for all of the bytes of R0 and R1, based on the corresponding data bytes. Note that non-zero indices enable R1 values to include a contribution from every element of Pi.

[0284] Thus there are no dependencies on computing any of the bytes of R0 on each other or between any of the bytes of R1. Therefore various embodiments are contemplated where R0 and R1 values are computed entirely (massively) in parallel, partially in parallel, or entirely in serial. For example, all of the R0 and/or R1 bytes of a page are computed in parallel. For another example, all of the R0 bytes of a page are computed in parallel followed (or preceded) by computing in parallel all of the R1 bytes of a page.

[0285] Further, there are no ordering requirements on computing any of the bytes of R0 with respect to each other or with respect to computing any of the bytes of R1. Therefore various embodiments are contemplated where R0 and R1 values are computed entirely in-order with respect to each other, in-order with respect to each other separately (e.g. R0 computations are in-order with respect to each other but with no ordering with respect to R1, and vice-versa), or with no particular ordering (e.g. entirely out-of-order, unordered, or randomly ordered). For example, all of the R0 bytes of a page are computed in a particular order (e.g. from lowest byte to highest byte), followed by all of the R1 bytes of the page in the particular order. For another example, all of the R0 bytes of a page are computed in the particular order, and independently all of the R1 bytes of the page are computed in the particular order. For yet another example, all of the R0 bytes of a page as well as all of the R1 bytes of a page are computed in no particular order with respect to each other (e.g. as data operands become available).

[0286] For yet another example, all of the R0 and R1 bytes of one or more pages are computed in an order determined by an order of completion of one or more read operations performed on one or more NVMs (each having, e.g., one or more flash die), the read operations for reading the data bytes referenced by the summation and weighted-summation computations (Pi) illustrated in Fig. 11. Performing the computations in the order determined by the completion of the read operations enables, in some embodiments and/or usage scenarios, reduction or elimination of buffering between the NVMs and elements enabled to perform the computations. Performing the computations in the order determined by the completion of the read operations enables, in some embodiments and/or usage scenarios, reduction in memory bandwidth used to perform the computations, such as memory bandwidth of the NVMs.

[0287] For yet another example, all of the R0 and R1 bytes of one or more pages are computed in an order determined by an order of data returned and/or data available in response to read operations performed on one or more NVMs (each having, e.g., one or more flash die), the data returned and/or data available corresponding to the data bytes referenced by the summation and weighted-summation computations (Pi) illustrated in Fig. 11. In some embodiments and/or usage scenarios, performing the computations in the order determined by the data returned and/or data available reduces or eliminates buffering between the NVMs and elements enabled to perform the computations. In some embodiments and/or usage scenarios, performing the computations in the order determined by the data returned and/or data available reduces memory bandwidth used to perform the computations, such as memory bandwidth of the NVMs. In some embodiments, the read operations are performed in a particular order (e.g. from

lowest byte to highest byte of Pi), while in other embodiments, the read operations are performed in no particular order.

[0288] Fig. 12 illustrates selected details of an embodiment of recovery from one (lower-level) failure (during a single operation), for instance in a context such as associated with Fig. 11, and where the lower-level failure has occurred on page j. Note that if the lower-level failure is on an R0 or an R1 page, then R0 (or R1) is re-determined as described by Fig. 11. Fig. 12 illustrates a computation for one byte of a recovered value for page j (note that the summation omits page j where the failure occurred). The computation as illustrated in Fig. 12 is repeated to determine recovered values for all of the bytes of page j, based on corresponding bytes of R0 and corresponding bytes from all of the data pages except for page j.

[0289] Thus there are no dependencies on computing any of the bytes of the recovered values of page j on each other. Therefore various embodiments are contemplated where Pj recovery values are computed ranging from highly parallel to highly serial, similarly to the aforementioned computations for R0 and R1 values. Further, there are no order requirements on computing any of the recovery values of page j with respect to each other. Therefore various embodiments are contemplated where Pj recovery values are computed in varying orderings with respect to each other, similarly to the aforementioned computations for R0 and R1 values.

[0290] Some exemplary embodiments perform computations related to determining recovery values in orderings at least in part determined by ordering of completion of one or more read operations performed on one or more NVMs (each having, e.g., one or more flash die), the read operations for reading the NVMs to obtain R0 and/or Pi values as illustrated by Fig. 12. Performing the computations in the order of the completion of the read operations enables, in some embodiments and/or usage scenarios, reduction or elimination of buffering between the NVMs and elements enabled to perform the computations. Performing the computations in the order of the completion of the read operations enables, in some embodiments and/or usage scenarios, reduction in memory bandwidth used to perform the computations, such as memory bandwidth of the NVMs.

[0291] Some exemplary embodiments perform computations related to determining recovery values in orderings at least in part determined by ordering of data returned and/or data available from one or more NVMs (each having, e.g., one or more flash die), the data returned and/or data available being in response to read operations performed on the NVMs to obtain R0 and/or Pi values as illustrated by Fig. 12. In some embodiments and/or usage scenarios,

performing the computations in the order of the data returned and/or data available from the read operations reduces or eliminates buffering between the NVMs and elements enabled to perform the computations. In some embodiments and/or usage scenarios, performing the computations in the order of the data returned and/or data available from the read operations reduces memory bandwidth used to perform the computations, such as memory bandwidth of the NVMs. In some embodiments, the read operations are performed in a particular order (e.g. from lowest byte to highest byte of Pi), while in other embodiments, the read operations are performed in no particular order.

[0292] Figs. 13A-13D illustrate selected details of an embodiment of recovery from two (lower-level) failures (during a single operation), for instance in a context such as associated with Fig. 11, and where the lower-level failures have occurred on pages m and n. Note that if the lower-level failures are on R0 and R1 pages, then R0 and R1 are unneeded for data recovery, and there is no processing to recover data. Otherwise, note that if one of the lower-level failures is an R1 page, then data recovery proceeds as described by Fig. 12. If neither of the lower-level failures are on R0 or R1 pages, then recovery of Pm and Pn values proceeds as follows. Fig. 13A illustrates computations for one byte of a revised R0 as R0' and for one byte of a revised R1 as R1' (note that the summations omit pages m and n where the failures occurred). Fig. 13B illustrates equalities relating one byte of the original redundancy information (R0, R1) to the revised redundancy information (R0', R1'), and the terms omitted from the summations used to form the revised R0 and R1 (Pm + Pn and Km*Pm + Kn*Pn). Fig. 13C illustrates an algebraic rearrangement of Fig. 13B, with introduced terms delta R0 (ΔR0) and delta R1 (ΔR1). Fig. 13D illustrates a solution of Fig. 13C for Pn and Pm, and thus represents computations for one byte of a recovered value for page n and one byte of a recovered value for page m, based on the corresponding bytes of R1, R0, and corresponding bytes from all of the data pages except for pages m and n. Note that unique indices enable a non-zero denominator, as Kn is unique with respect to Km.

[0293] In various embodiments, computations as represented by Figs. 13A-13D are performed to determine one byte of a recovered value for page m and one byte of a recovered value for page n. The computations are repeated to determine recovered values for all of the bytes of pages m and n, based on corresponding bytes of R0, R1, and corresponding bytes from all of the data pages except for pages m and n.

[0294] Thus there are no dependencies on computing any of the bytes of the recovered values of pages m or n on each other. Therefore various embodiments are contemplated where

Pm and/or Pn recovery values are computed ranging from highly parallel to highly serial, similarly to the aforementioned computations for Pj recovery values. Further, there are no order requirements on computing any of the recovery values of page m and/or page n with respect to each other. Therefore various embodiments are contemplated where Pm and/or Pn recovery values are computed in varying orderings with respect to each other, similarly to the aforementioned computations for Pj recovery values.

[0295] Some exemplary embodiments perform computations related to determining recovery values (such as computations for R0' and R1') in orderings at least in part determined by ordering of completion of one or more read operations performed on one or more NVMs (each having, e.g., one or more flash die), the read operations to obtain any one or more of values illustrated as sources for computations in any of Figs. 13A-13D. Performing the computations in the order of the completion of the read operations enables, in some embodiments and/or usage scenarios, reduction or elimination of buffering between the NVMs and elements enabled to perform the computations. Performing the computations in the order of the completion of the read operations enables, in some embodiments and/or usage scenarios, reduction in memory bandwidth used to perform the computations, such as memory bandwidth of the NVMs.

[0296] Some other exemplary embodiments perform computations related to determining recovery values (such as computations for R0' and R1') in orderings at least in part determined by ordering of data returned and/or data available from one or more NVMs (each having, e.g., one or more flash die), the data returned and/or data available being in response to read operations performed on the NVMs to obtain any one or more of values illustrated as sources for computations in any of Figs. 13A-13D. In some embodiments and/or usage scenarios, performing the computations in the order of the data returned and/or data available from the read operations reduces or eliminates buffering between the NVMs and elements enabled to perform the computations. In some embodiments and/or usage scenarios, performing the computations in the order of the data returned and/or data available from the read operations reduces memory bandwidth used to perform the computations, such as memory bandwidth of the NVMs. In some embodiments, the read operations are performed in a particular order (e.g. from lowest byte to highest byte of Pi), while in other embodiments, the read operations are performed in no particular order.

[0297] Figs. 14A and 14B illustrate selected details of an embodiment of computing higher-level redundancy information with respect to pages received from NVMs. Fig. 14A

illustrates selected details of an embodiment of sending a plurality of read commands to one or more NVMs via issuing (Issue Read Operation **1402A**), checking if all of the commands have been sent (All Issued? **1403A**), and if not, then looping back to send another of the commands. Note that other embodiments are contemplated where a plurality of commands are issued at a time, rather than one at a time.

[0298] Fig. 14B illustrates selected details of an embodiment of processing pages received from the NVMs in response to the read commands sent as illustrated in Fig. 14A. A check is made to determine if a page is available (Page Ready? **1402B**). If not, then processing loops back to perform the check again. If a page is available, then higher-level redundancy information processing relating to the page is carried out (Perform Page-Based Computations **1403B**). Then a check is made to determine if all pages have been processed (Pages Finished? **1404B**). If so, then processing is complete (End **1499B**), otherwise flow loops back to determine if another page is available.

[0299] Other than reception of pages in response to the commands sent as illustrated in Fig. 14A, the processing illustrated in Fig. 14B is independent of processing illustrated in Fig. 14A. In various scenarios, arrival order of the pages varies according to the NVM type, state, operating environment, and other factors, and in some circumstances is different than sending order or arrival order of the read commands corresponding to the arriving pages. As processing of Fig. 14A is independent of Fig. 14B (other than page data arrival being dependent on a corresponding read command being sent), in some embodiments and/or usage scenarios, read commands are being sent (Fig. 14A) while read data is being received/processed (Fig. 14B). In some embodiments and/or usage scenarios, some of the pages are provided from one or more buffers rather than being requested via read commands directed to the NVMs, for example if a particular page is present in a particular buffer before a read command for the particular page is to be sent. In some embodiments and/or usage scenarios, pages other than in response to the commands sent are provided by the NVMs intermixed with the pages provided in response to the commands sent, e.g. pages provided in response to read commands sent for other activities.

[0300] In various embodiments, computations for R0 and R1, as illustrated by Fig. 11, are performed at least in part as illustrated by Figs. 14A and 14B. For a first example, read commands for all data pages necessary to compute corresponding R0 and R1 pages are sent to one or more NVMs as illustrated by Fig. 14A. The pages of data received in response to the read commands are processed as the pages are received to compute the R0 and the R1 pages, as illustrated by Fig. 11. For a second example, read commands for a pair (corresponding, e.g., to

two planes of a dual-plane NVM) of R0 and R1 pages are sent to one or more NVMs as illustrated by Fig. 14A. The pages of data received in response to the read commands are processed as the pages are received to compute the R0 and the R1 pages, as illustrated by Fig. 11.

[0301] In various embodiments, computations for Pj, as illustrated by Fig. 12, are performed at least in part as illustrated by Figs. 14A and 14B. For a first example, read commands for all data pages necessary to compute a particular Pj page are sent to one or more NVMs as illustrated by Fig. 14A. The pages of data received in response to the read commands are processed as the pages are received to compute the Pj page, as illustrated by Fig. 12. For a second example, read commands for a pair of Pj pages (corresponding, e.g., to two planes of a dual-plane NVM) are sent to one or more NVMs as illustrated by Fig. 14A, and the pages of data received are processed as received, as illustrated by Fig. 14B, to compute the pair of Pj pages.

[0302] In various embodiments, computations related to determining recovery values (such as computations for R0' and R1'), as illustrated by any of Figs. 13A-13D, are performed at least in part as illustrated by Figs. 14A and 14B. For a first example, read commands for all data pages necessary to compute a particular R0' page and a particular R1' are sent to one or more NVMs as illustrated by Fig. 14A. The pages of data received in response to the read commands are processed as the pages are received to compute the R0' and R1' pages, as illustrated by Fig. 13A. For a second example, read commands for a pair of R0' and R1' pages (corresponding, e.g., to two planes of a dual-plane NVM) are sent to one or more NVMs as illustrated by Fig. 14A, and the pages of data received are processed as received, as illustrated by Fig. 14B, to compute the pair of R0' and R1' pages.

[0303] Figs. 15A-15C illustrate selected details of an embodiment of backing out of a computation of higher-level redundancy information with respect to a write provided to NVMs, for instance in a context such as associated with Fig. 11. Fig. 15A illustrates selected details of an embodiment of sending a plurality of write commands to one or more NVMs via issuing (Issue Write Operation **1502A**), checking if all of the commands have been sent (All Issued? **1503A**), and if not, then looping back to send another of the commands. Note that other embodiments are contemplated where a plurality of commands are issued at a time, rather than one at a time.

[0304] Fig. 15B illustrates selected details of an embodiment of processing write completion and status information received from the NVMs in response to the write commands sent as illustrated in Fig. 15A. A check is made to determine if a write has completed without errors (Write OK? **1502B**). If so, then a check is made to determine if all writes have been completed (Writes Finished? **1504B**). If so, then processing is complete (End **1599B**). If a write has been completed but with a (lower-level) error such as a program failure, then the flow proceeds to "undo" the effect of the write with respect to higher-level redundancy information computation (Backout Write from Redundancy **1503B**). More specifically, data for the write with the lower-level error is de-computed from any corresponding higher-level redundancy information computations (assuming that the data for the write had already been included in the corresponding higher-level redundancy computations under a presumption that the write would succeed). For example, a lower-level write failure is detected on a particular page j. In response, revised R0 and R1 pages are computed such that page j data (Pj) is set to zero. Fig. 15C illustrates selected details of an embodiment of a computation for a single byte of a revised R0 (nR0) and a single byte of a revised R1 (nR1), where j is the page of the lower-level write failure. Note that in contexts of Fig. 12 such as associated with Fig. 11, if the finite field is a Galois Field the subtraction operation ("-") illustrated in Fig. 15C is equivalent to a logical XOR operation. Other processing (not illustrated) is performed, in various embodiments, to store the page with the lower-level write failure (Pj), as well as the revised higher-level redundancy pages (nR0 and nR1).

[0305] Other than reception of write completion and status information in response to the commands sent as illustrated in Fig. 15A, the processing illustrated in Fig. 15B is independent of processing illustrated in Fig. 15A. In various scenarios, arrival order of the write completion and status information varies according to the NVM type, state, operating environment, and other factors, and in some circumstances is different than sending order or arrival order of the write commands corresponding to the arriving write completion and status information. As processing of Fig. 15A is independent of Fig. 15B (other than write completion and status information arrival being dependent on a corresponding write command being sent), in some embodiments and/or usage scenarios, write commands are being sent (Fig. 15A) while write completion and status information is being received/processed (Fig. 15B).

[0306] In some embodiments and/or usage scenarios, a significant latency transpires between write commands being sent to the NVMs and reception of write completion and status information from the NVMs in response to the write commands. In some embodiments and/or usage scenarios, write completion and status information other than in response to the

commands sent as illustrated in Fig. 15A are provided by the NVMs intermixed with the write completion and status information provided in response to the commands sent as illustrated in Fig. 15A, e.g. write completion and status information provided in response to write commands sent for other activities.

[0307] In various embodiments and/or usage scenarios, one or more pages that would otherwise be used for data information are unused. In various scenarios an unused page is a first, middle, or last page of a block, R-block, stripe, or sequence of pages any kind. In some circumstances, unused pages are unused a priori ("left out"), and in some circumstances unused pages are unused after some use ("removed from service"). An example of an unused page that is left out is a page that is manufactured incorrectly. An example of an unused page that is removed from service is a page that fails to write properly (e.g. as described as a lower-level write error with respect to Fig. 15B). Processing, e.g. relating to Figs. 10-12, 13A-13D, 14A-14B, and 15A-15B, skips over any unused pages (whether left out or removed from service), such as by omitting the unused pages entirely from computations, or by performing computations as if all data on the unused pages were zero.

[0308] In various embodiments, processing of or relating to one or more elements of Figs. 10-12, 13A-13D, 14A-14B, and 15A-15B is performed entirely or in part by one or more elements (or portions thereof) of Fig. 1A. For example, a portion of software execution capabilities of CPU **171** is used to manage higher-level redundancy processing, such as by directing data recovery operations as illustrated by Fig. 12 and Figs. 13A-13D. For another example, Data Processing **121** includes hardware elements dedicated to and/or specialized for computation of higher-level redundancy information in accordance with computations such as illustrated by Fig. 11.

EXAMPLE IMPLEMENTATION TECHNIQUES

[0309] In some embodiments, various combinations of all or portions of operations performed by a system implementing dynamic higher-level redundancy mode management, such as with independent silicon elements (e.g. flash memories), a computing-host flash memory controller, and/or an SSD controller (such as SSD Controller **100** of Fig. 1A), and portions of a processor, microprocessor, system-on-a-chip, application-specific-integrated-circuit, hardware accelerator, or other circuitry providing all or portions of the aforementioned operations, are specified by a specification compatible with processing by a computer system. The specification

is in accordance with various descriptions, such as hardware description languages, circuit descriptions, netlist descriptions, mask descriptions, or layout descriptions. Example descriptions include: Verilog, VHDL, SPICE, SPICE variants such as PSpice, IBIS, LEF, DEF, GDS-II, OASIS, or other descriptions. In various embodiments, the processing includes any combination of interpretation, compilation, simulation, and synthesis to produce, to verify, or to specify logic and/or circuitry suitable for inclusion on one or more integrated circuits. Each integrated circuit, according to various embodiments, is designable and/or manufacturable according to a variety of techniques. The techniques include a programmable technique (such as a field or mask programmable gate array integrated circuit), a semi-custom technique (such as a wholly or partially cell-based integrated circuit), and a full-custom technique (such as an integrated circuit that is substantially specialized), any combination thereof, or any other technique compatible with design and/or manufacturing of integrated circuits.

[0310] In some embodiments, various combinations of all or portions of operations as described by a computer readable medium having a set of instructions stored therein, are performed by execution and/or interpretation of one or more program instructions, by interpretation and/or compiling of one or more source and/or script language statements, or by execution of binary instructions produced by compiling, translating, and/or interpreting information expressed in programming and/or scripting language statements. The statements are compatible with any standard programming or scripting language (such as C, C++, Fortran, Pascal, Ada, Java, VBscript, and Shell). One or more of the program instructions, the language statements, or the binary instructions, are optionally stored on one or more computer readable storage medium elements. In various embodiments some, all, or various portions of the program instructions are realized as one or more functions, routines, sub-routines, in-line routines, procedures, macros, or portions thereof.

CONCLUSION

[0311] Certain choices have been made in the description merely for convenience in preparing the text and drawings and unless there is an indication to the contrary the choices should not be construed per se as conveying additional information regarding structure or operation of the embodiments described. Examples of the choices include: the particular organization or assignment of the designations used for the figure numbering and the particular organization or assignment of the element identifiers (the callouts or numerical designators, e.g.) used to identify and reference the features and elements of the embodiments.

[0312] The words "includes" or "including" are specifically intended to be construed as abstractions describing logical sets of open-ended scope and are not meant to convey physical containment unless explicitly followed by the word "within."

[0313] Although the foregoing embodiments have been described in some detail for purposes of clarity of description and understanding, the invention is not limited to the details provided. There are many embodiments of the invention. The disclosed embodiments are exemplary and not restrictive.

[0314] It will be understood that many variations in construction, arrangement, and use are possible consistent with the description, and are within the scope of the claims of the issued patent. For example, interconnect and function-unit bit-widths, clock speeds, and the type of technology used are variable according to various embodiments in each component block. The names given to interconnect and logic are merely exemplary, and should not be construed as limiting the concepts described. The order and arrangement of flowchart and flow diagram process, action, and function elements are variable according to various embodiments. Also, unless specifically stated to the contrary, value ranges specified, maximum and minimum values used, or other particular specifications (such as flash memory technology types; and the number of entries or stages in registers and buffers), are merely those of the described embodiments, are expected to track improvements and changes in implementation technology, and should not be construed as limitations.

[0315] Functionally equivalent techniques known in the art are employable instead of those described to implement various components, sub-systems, operations, functions, routines, sub-routines, in-line routines, procedures, macros, or portions thereof. It is also understood that many functional aspects of embodiments are realizable selectively in either hardware (i.e.,

generally dedicated circuitry) or software (i.e., via some manner of programmed controller or processor), as a function of embodiment dependent design constraints and technology trends of faster processing (facilitating migration of functions previously in hardware into software) and higher integration density (facilitating migration of functions previously in software into hardware). Specific variations in various embodiments include, but are not limited to: differences in partitioning; different form factors and configurations; use of different operating systems and other system software; use of different interface standards, network protocols, or communication links; and other variations to be expected when implementing the concepts described herein in accordance with the unique engineering and business constraints of a particular application.

[0316] The embodiments have been described with detail and environmental context well beyond that required for a minimal implementation of many aspects of the embodiments described. Those of ordinary skill in the art will recognize that some embodiments omit disclosed components or features without altering the basic cooperation among the remaining elements. It is thus understood that much of the details disclosed are not required to implement various aspects of the embodiments described. To the extent that the remaining elements are distinguishable from the prior art, components and features that are omitted are not limiting on the concepts described herein.

[0317] All such variations in design are insubstantial changes over the teachings conveyed by the described embodiments. It is also understood that the embodiments described herein have broad applicability to other computing and networking applications, and are not limited to the particular application or industry of the described embodiments. The invention is thus to be construed as including all possible modifications and variations encompassed within the scope of the claims of the issued patent.

WHAT IS CLAIMED IS:

1. A system comprising:
means for computing k redundant data portions to protect n-k useful data portions;
means for storing each of the k redundant data portions and each of the n-k useful data portions in separate corresponding areas, each of the corresponding areas in a respective one of n physical devices;
means for, subsequent to a failure such that one of the corresponding areas is no longer usable, computing j redundant data portions to protect n-1-j useful data portions; and
wherein j <= k.

2. The system of claim 1, wherein the means for computing k redundant data portions to protect n-k useful data portions is operable in a first data protection mode, the means for computing j redundant data portions to protect n-1-j useful data portions is operable in a second data protection mode, and further comprising means for, in response to the failure, switching from operating in the first data protection mode to operating in the second data protection mode.

3. The system of claim 1, wherein when j == k, a total amount of the useful data portions is decreased and a degree of protection is at least preserved.

4. The system of claim 3, wherein the degree of protection is protection against failures of up to j of the corresponding areas.

5. The system of claim 1, wherein when j == k-1 and j >=1, a total amount of the useful data portions is preserved, a degree of protection is decreased, and the useful data portions are protected.

6. The system of claim 5, wherein the decreased degree of protection is protection against failures of up to j-1 of the corresponding areas.

7. The system of claim 1,

wherein each of the n physical devices comprises a plurality of the corresponding areas;

further comprising means for storing, in each of a plurality of sets of the corresponding areas, a respective set of data comprising a useful data subset and a redundant data subset protecting the useful data subset, the redundant data subset stored in distinct ones of the n physical devices from the useful data subset; and

wherein each of the sets of the corresponding areas comprises no more than one corresponding area of each one of the n physical devices, and the means for storing is in accordance with each of the respective sets of data optionally being stored in a different set of n or less of the n physical devices.

8. The system of claim 7, wherein prior to the failure, the useful data subset of a particular one of the sets of data comprises the n-k useful data portions, and the redundant data subset of the particular set of data comprises the k redundant data portions.

9. The system of claim 7, wherein the means for storing is in accordance with a first one of the corresponding areas in a particular one of the n physical devices being used to store a portion of the useful data subset of a first one of the sets of data, and is further in accordance with a second one of the corresponding areas in the particular physical device being used to store a portion or all of the redundant data subset of a second one of the sets of data.

10. The system of claim 7, wherein an amount of storage in a first one of the corresponding areas differs from an amount of storage in a second one of the corresponding areas.

11. The system of claim 10, wherein the first corresponding area and the second corresponding area are in a same one of the n physical devices.

12. A system comprising:

means for operating in a first higher-level redundancy mode comprising means for storing n information portions in n respective non-overlapping areas of memory;

means for operating in a second higher-level redundancy mode comprising means for storing n-1 information portions in n-1 respective non-overlapping areas of the memory;

means for detecting a failure of a failing one of the n respective non-overlapping areas;

means for, in response to the failure, dynamically transitioning from operating via the means for operating in the first higher-level redundancy mode to operating via the means for operating in the second higher-level redundancy mode, the n-1 of the n respective non-overlapping areas being the n respective non-overlapping areas excluding the failing one of the n respective non-overlapping areas;

wherein each of the n information portions comprises respective n-k data information portions and respective k redundant information portions computed to protect the respective n-k data information portions;

wherein each of the n-1 information portions comprises respective n-1-j data information portions and respective j redundant information portions computed to protect the respective n-1-j data information portions;

wherein j <= k; and

wherein each of the n respective non-overlapping areas is in a respective one of n physical devices of the memory.

13. The system of claim 12, wherein when j == k, a total amount of the data information portions is decreased and a degree of protection is at least preserved.

14. The system of claim 13, wherein the degree of protection is protection against failures of up to j of the n respective non-overlapping areas.

15. The system of claim 12, wherein when j == k-1 and j >=1, a total amount of the data information portions is preserved, a degree of protection is decreased, and the data information portions are protected.

16. The system of claim 15, wherein the decreased degree of protection is protection against failures of up to j-1 of the n respective non-overlapping areas.

17. A system comprising:

means for managing one or more flash memories in accordance with a current higher-level redundancy mode in response to read and/or write requests from a computing host; and

means for dynamically transitioning the current higher-level redundancy mode in response to detecting a failure of one or more portions of one or more of the flash memories.

18. The system of claim 17, wherein the means for dynamically transitioning comprises means for reducing free space.

19. The system of claim 17, wherein the means for dynamically transitioning comprises means for rearranging data storage in accordance with a target higher-level redundancy mode.

20. The system of claim 17, wherein the means for dynamically transitioning comprises means for determining and storing revised higher-level redundancy information in accordance with a target higher-level redundancy mode.

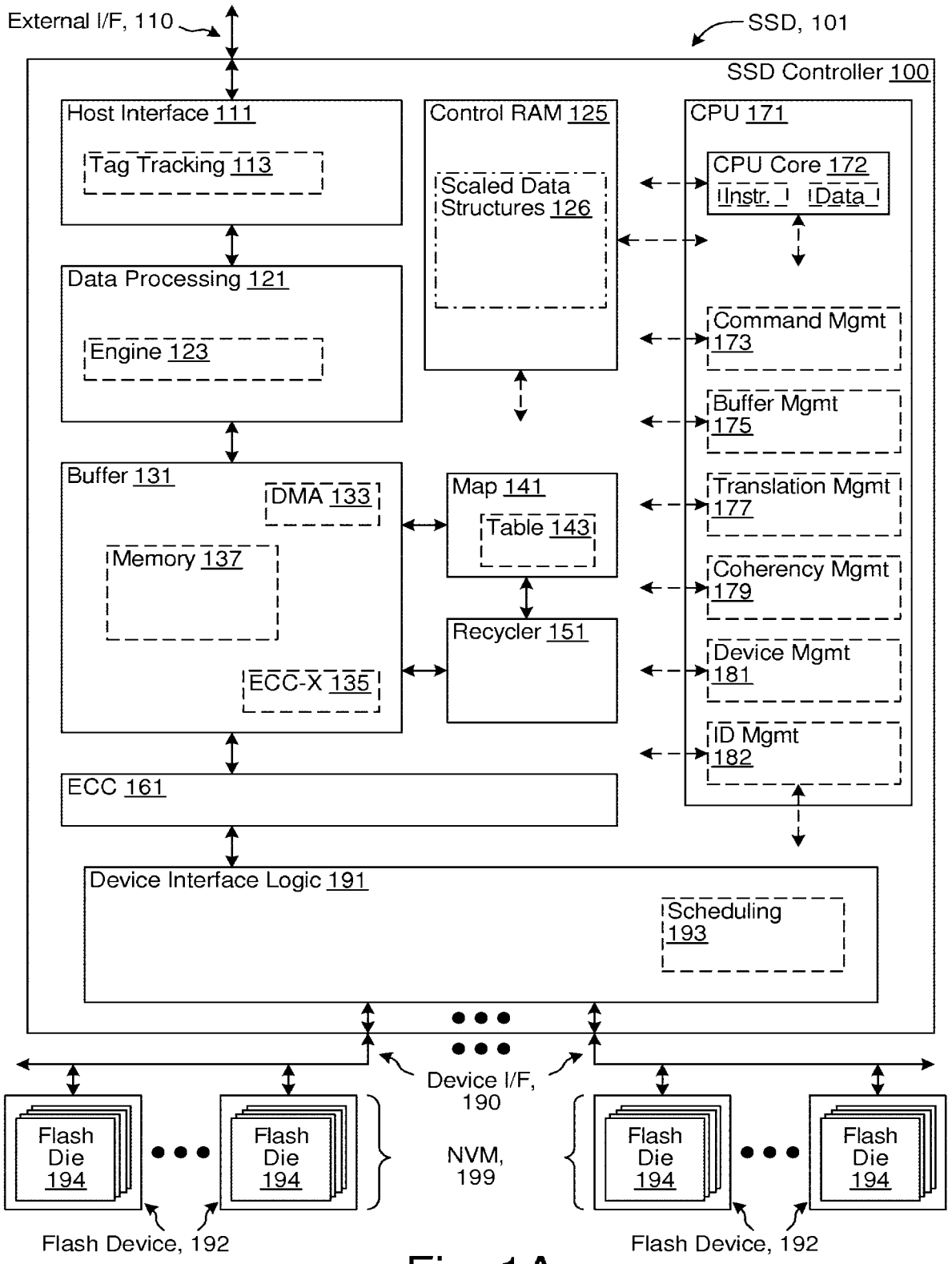

External I/F, 110
SSD, 101
SSD Controller 100
Host Interface 111
Tag Tracking 113
Data Processing 121
Engine 123
Buffer 131
DMA 133
Memory 137
ECC-X 135
ECC 161
Control RAM 125
Scaled Data Structures 126
Map 141
Table 143
Recycler 151
CPU 171
CPU Core 172
Instr.
Data
Command Mgmt 173
Buffer Mgmt 175
Translation Mgmt 177
Coherency Mgmt 179
Device Mgmt 181
ID Mgmt 182
Device Interface Logic 191
Scheduling 193
Device I/F, 190
NVM, 199
Flash Die 194
Flash Die 194
Flash Die 194
Flash Die 194
Flash Device, 192
Flash Device, 192

Fig. 1A

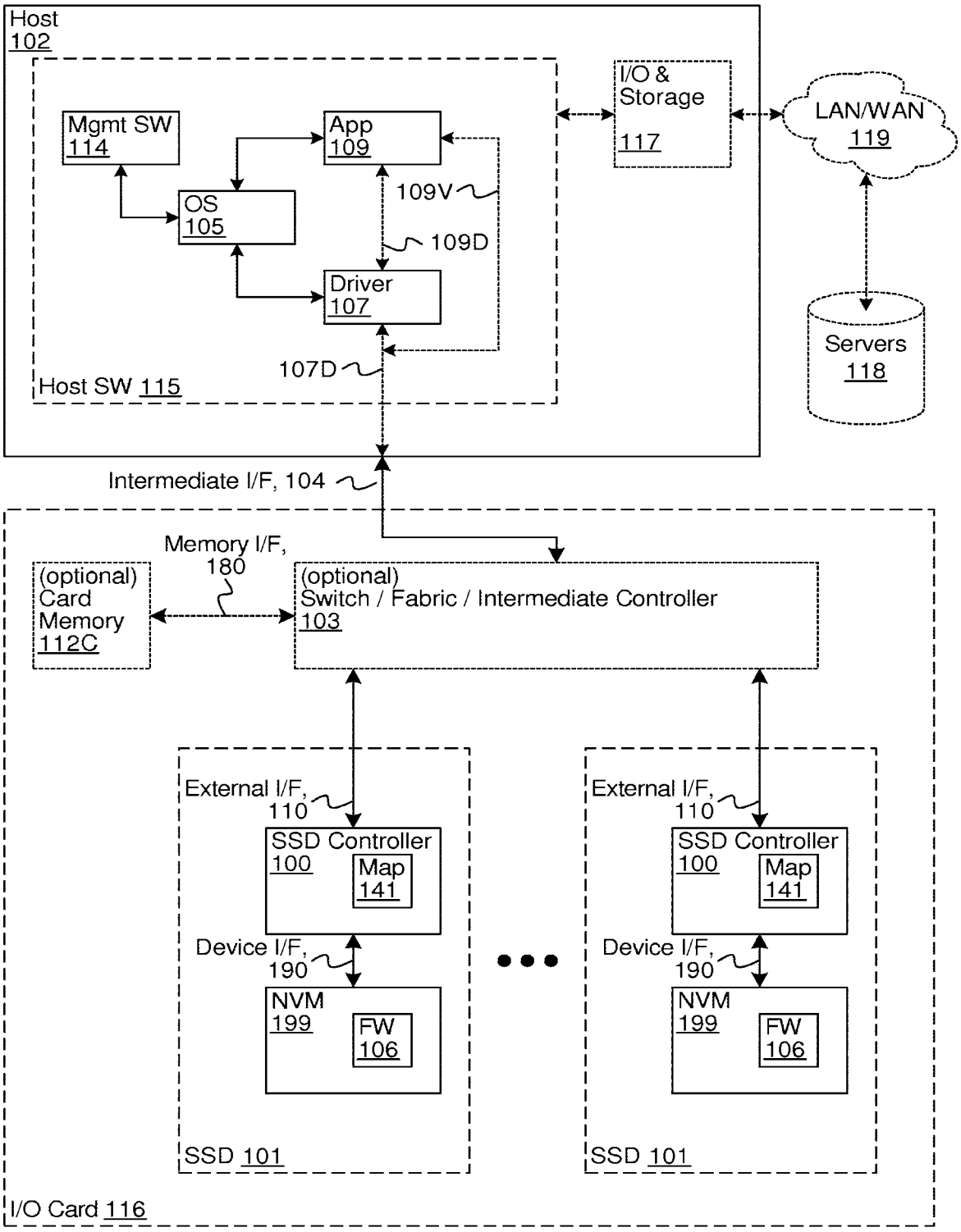

Host
102
Mgmt SW
114
App
109
OS
105
109V
109D
Driver
107
107D
Host SW 115
I/O &
Storage
117
LAN/WAN
119
Servers
118
Intermediate I/F, 104
Memory I/F,
180
(optional)
Card
Memory
112C
(optional)
Switch / Fabric / Intermediate Controller
103
External I/F,
110
SSD Controller
100
Map
141
Device I/F,
190
NVM
199
FW
106
SSD 101
External I/F,
110
SSD Controller
100
Map
141
Device I/F,
190
NVM
199
FW
106
SSD 101
I/O Card 116

Fig. 1B

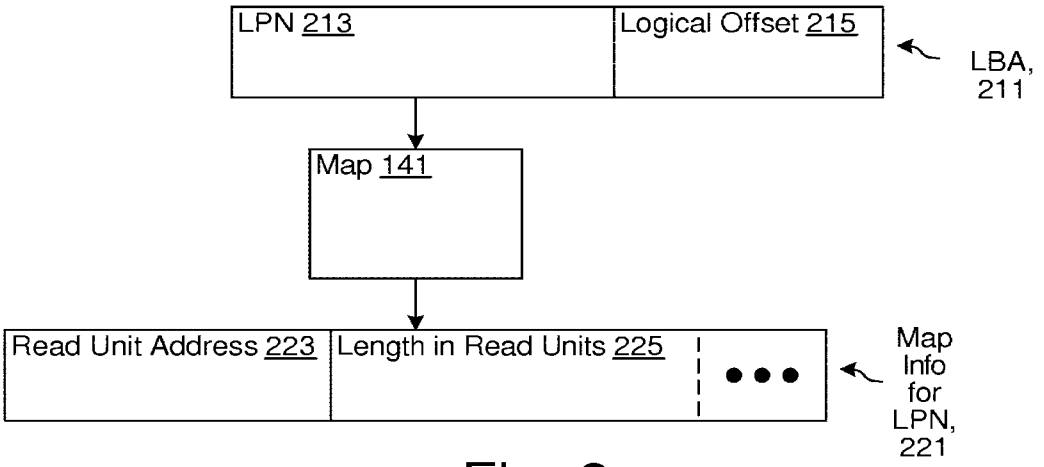

LPN 213
Logical Offset 215
LBA, 211
Map 141
Read Unit Address 223
Length in Read Units 225
Map Info for LPN, 221

Fig. 2

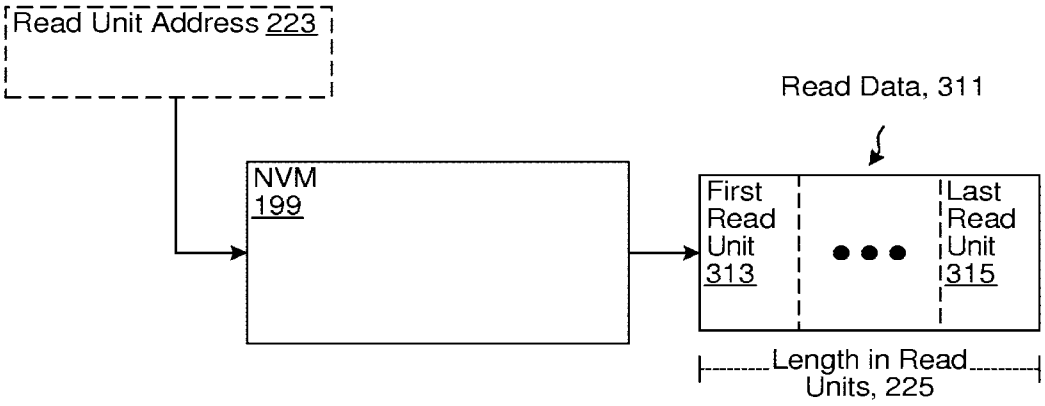

Read Unit Address 223
Read Data, 311
NVM 199
First Read Unit 313
Last Read Unit 315
Length in Read Units, 225

Fig. 3

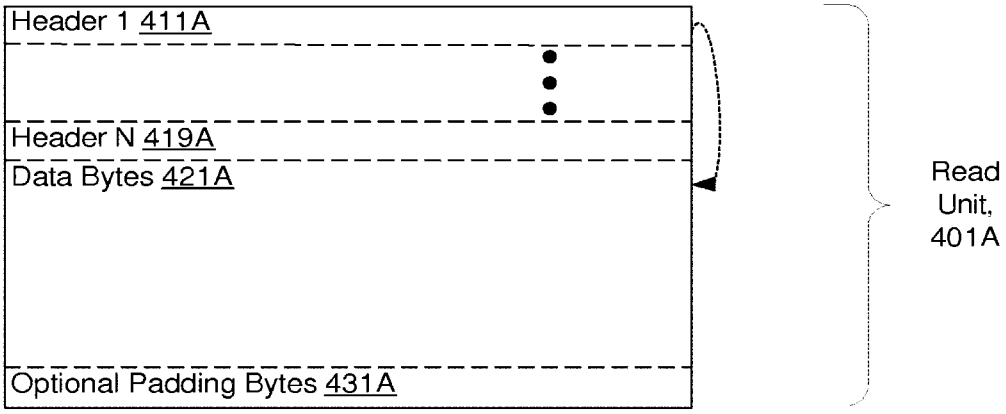

Header 1 411A
Header N 419A
Data Bytes 421A
Optional Padding Bytes 431A
Read Unit, 401A

Fig. 4A

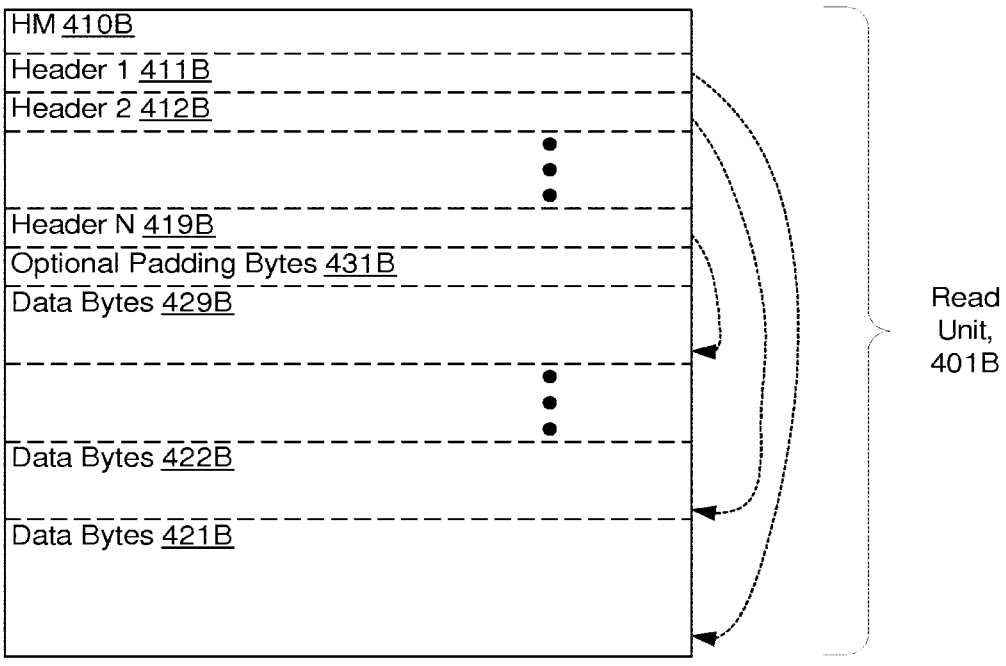

HM 410B
Header 1 411B
Header 2 412B
Header N 419B
Optional Padding Bytes 431B
Data Bytes 429B
Data Bytes 422B
Data Bytes 421B
Read Unit, 401B

Fig. 4B

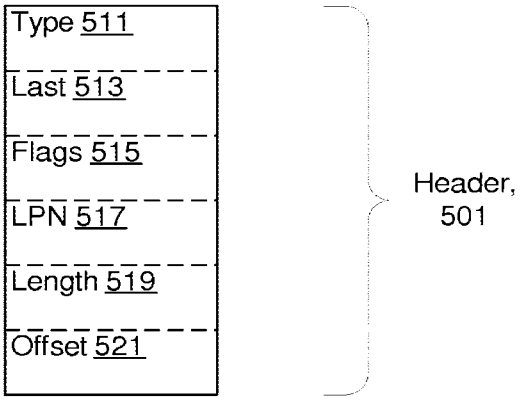

Type 511
Last 513
Flags 515
LPN 517
Length 519
Offset 521
Header, 501

Fig. 5

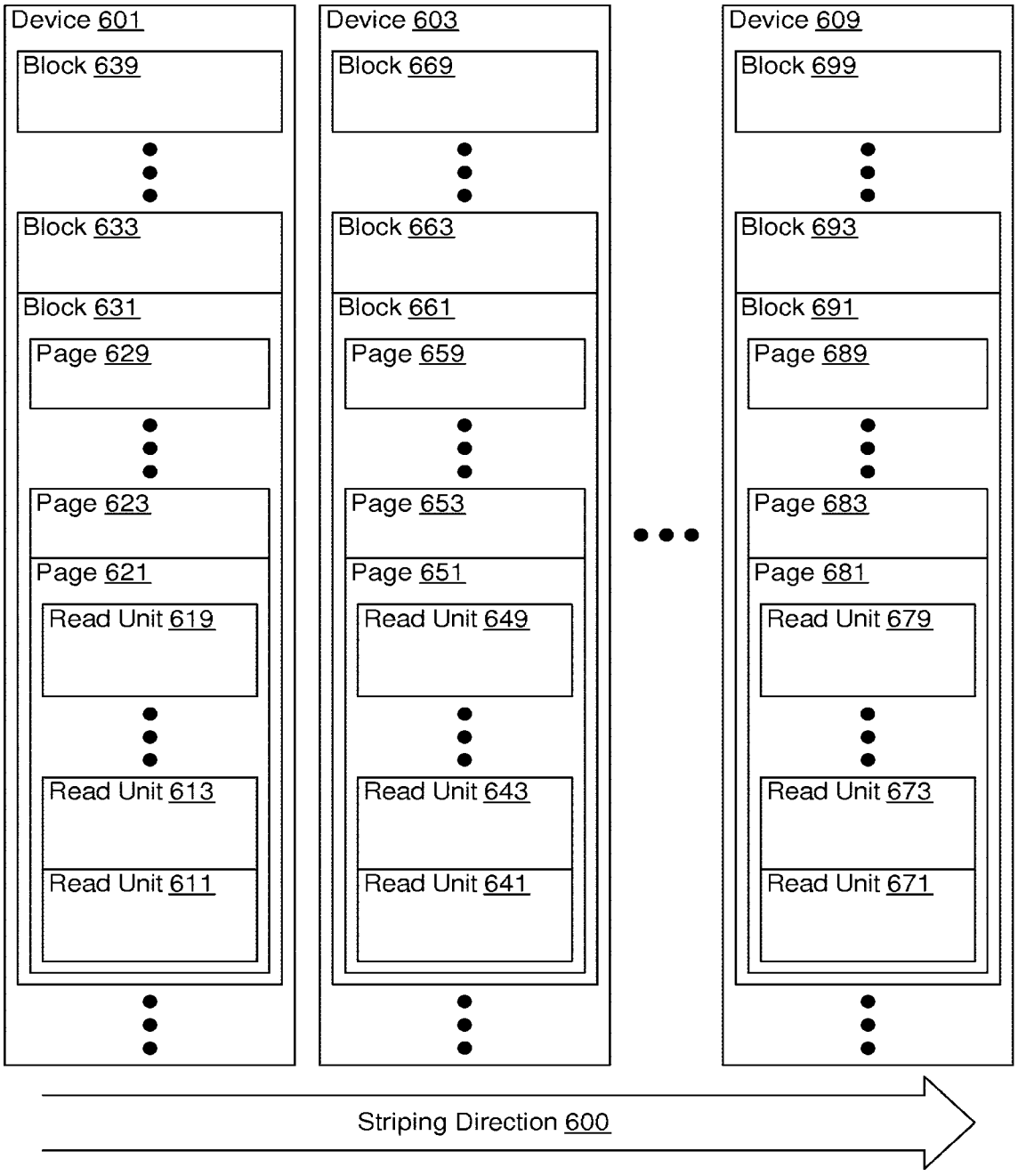

Device 601
Block 639
Block 633
Block 631
Page 629
Page 623
Page 621
Read Unit 619
Read Unit 613
Read Unit 611
Device 603
Block 669
Block 663
Block 661
Page 659
Page 653
Page 651
Read Unit 649
Read Unit 643
Read Unit 641
Device 609
Block 699
Block 693
Block 691
Page 689
Page 683
Page 681
Read Unit 679
Read Unit 673
Read Unit 671
Striping Direction 600

Fig. 6

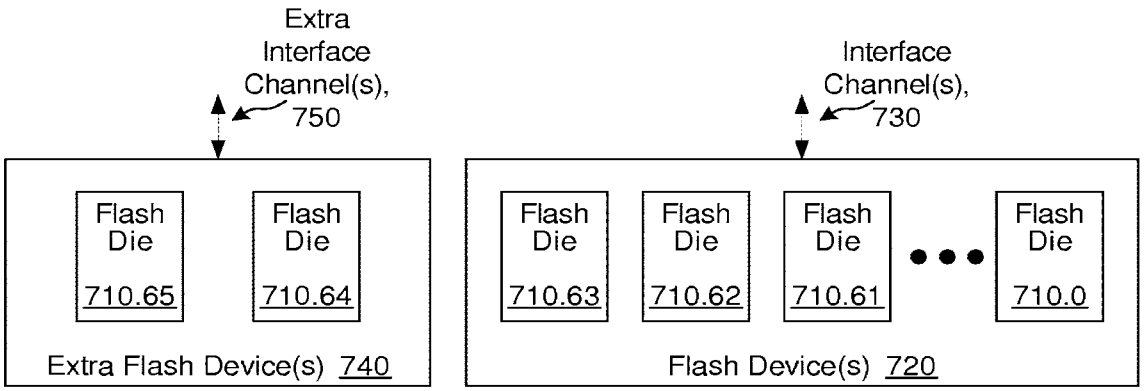

Extra
Interface
Channel(s),
750
Flash
Die
710.65
Flash
Die
710.64
Extra Flash Device(s) 740
Interface
Channel(s),
730
Flash
Die
710.63
Flash
Die
710.62
Flash
Die
710.61
Flash
Die
710.0
Flash Device(s) 720

Fig. 7A

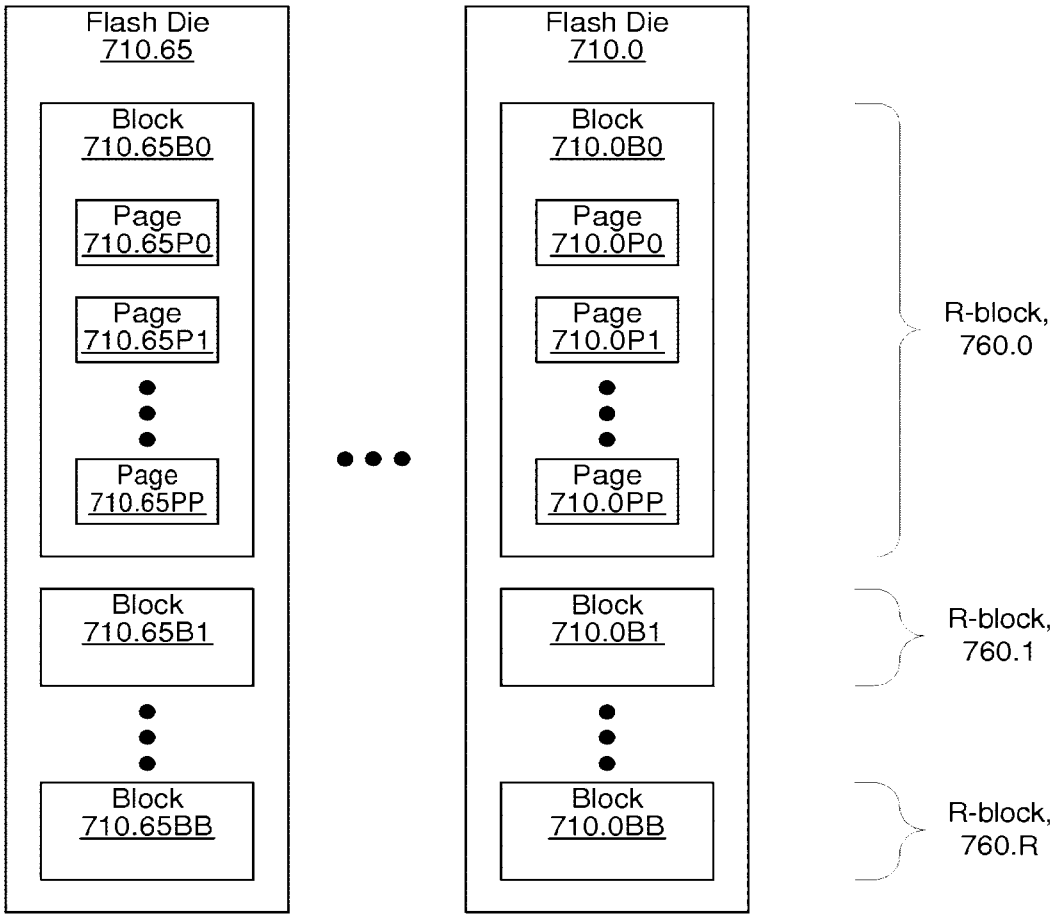

Flash Die
710.65
Block
710.65B0
Page
710.65P0
Page
710.65P1
Page
710.65PP
Block
710.65B1
Block
710.65BB
Flash Die
710.0
Block
710.0B0
Page
710.0P0
Page
710.0P1
Page
710.0PP
Block
710.0B1
Block
710.0BB
R-block,
760.0
R-block,
760.1
R-block,
760.R

Fig. 7B

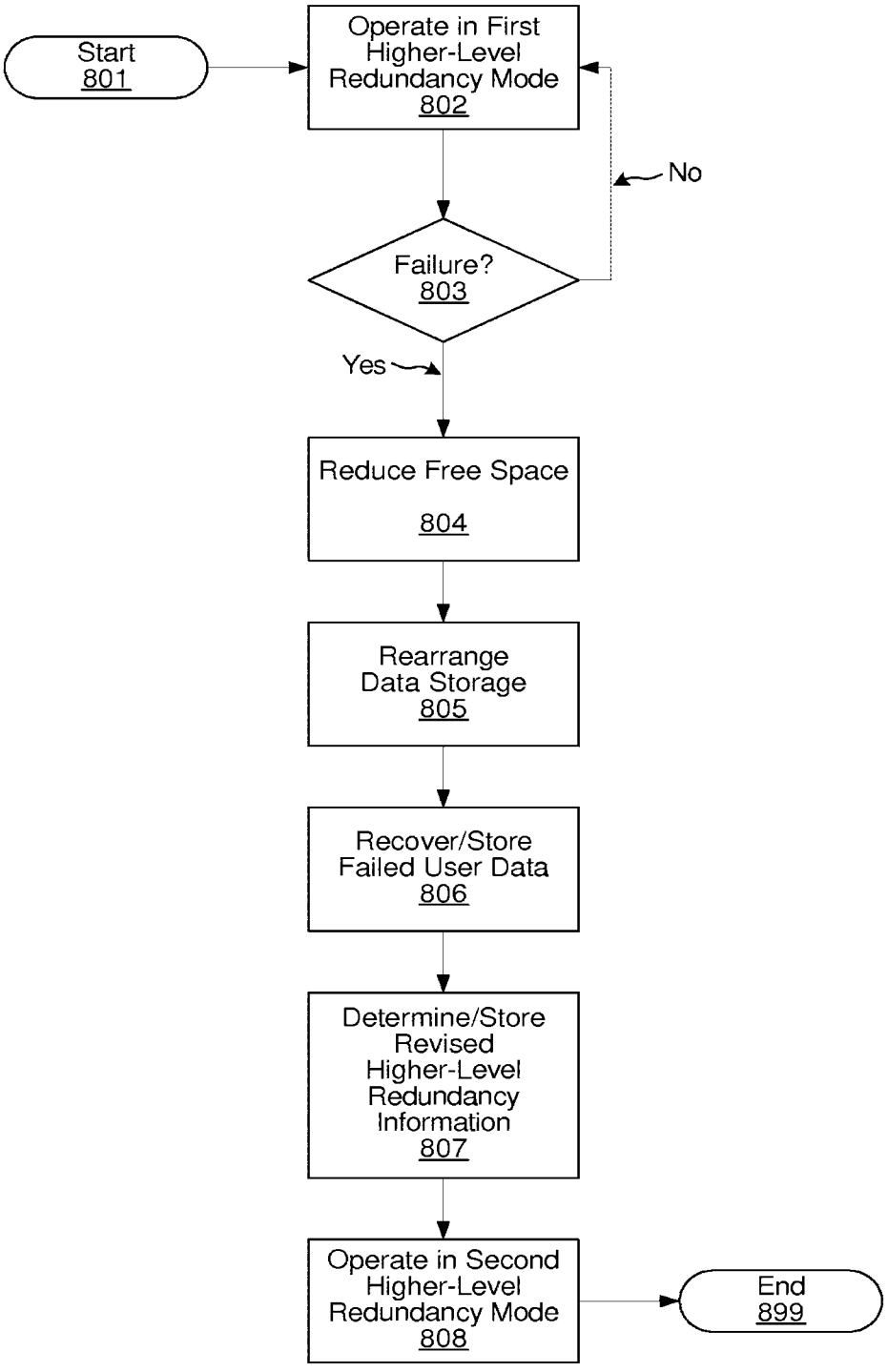

Start
801
Operate in First Higher-Level Redundancy Mode
802
No
Failure?
803
Yes
Reduce Free Space
804
Rearrange Data Storage
805
Recover/Store Failed User Data
806
Determine/Store Revised Higher-Level Redundancy Information
807
Operate in Second Higher-Level Redundancy Mode
808
End
899

Fig. 8

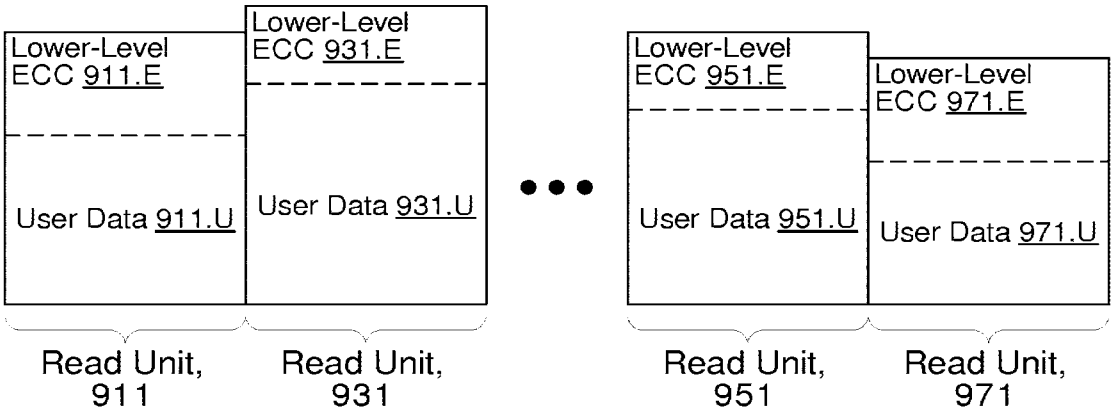

Lower-Level ECC 911.E
User Data 911.U
Lower-Level ECC 931.E
User Data 931.U
Lower-Level ECC 951.E
User Data 951.U
Lower-Level ECC 971.E
User Data 971.U
Read Unit, 911
Read Unit, 931
Read Unit, 951
Read Unit, 971

Fig. 9

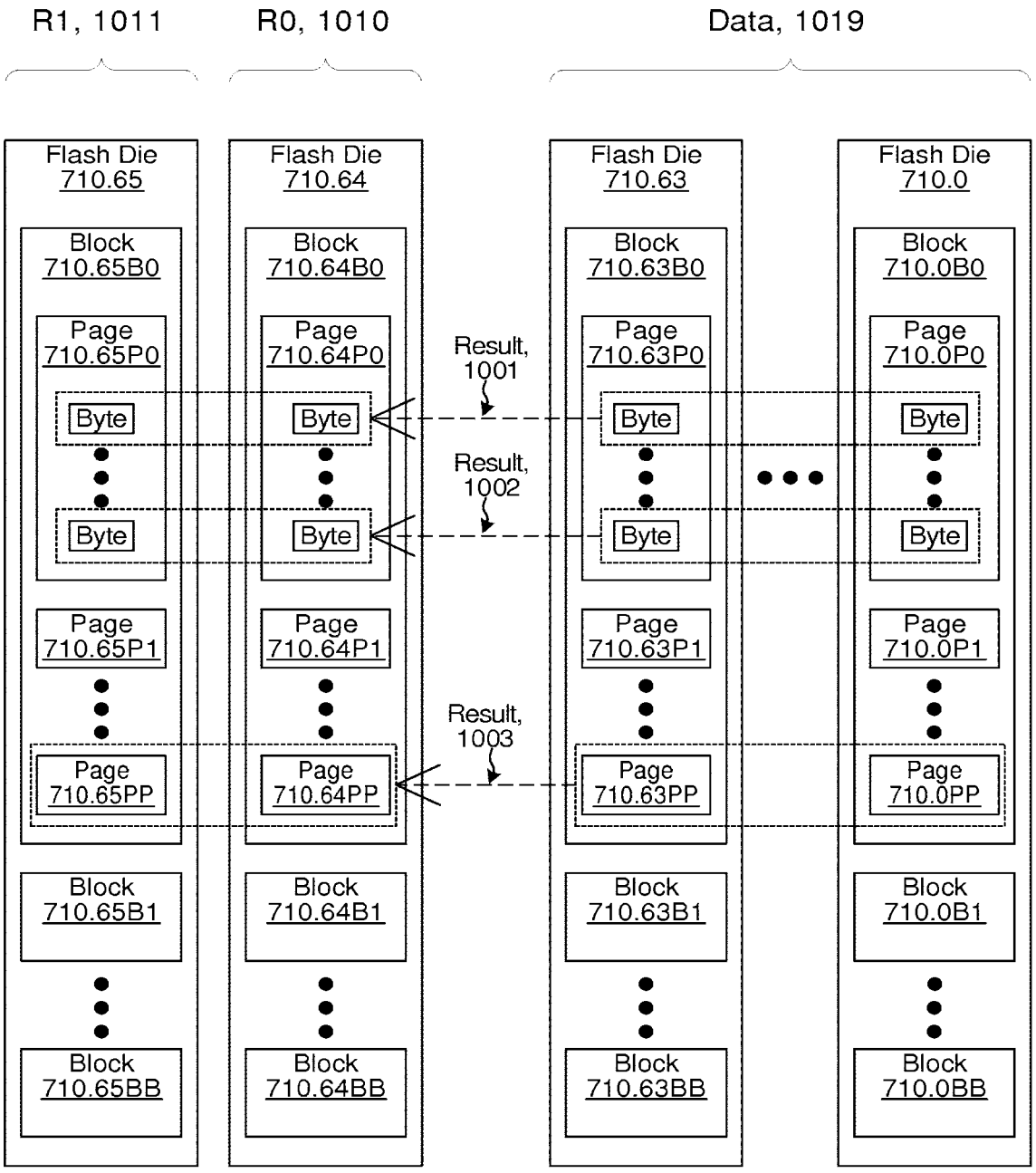

R1, 1011
R0, 1010
Data, 1019
Flash Die 710.65
Flash Die 710.64
Flash Die 710.63
Flash Die 710.0
Block 710.65B0
Block 710.64B0
Block 710.63B0
Block 710.0B0
Page 710.65P0
Page 710.64P0
Page 710.63P0
Page 710.0P0
Byte
Result, 1001
Result, 1002
Result, 1003
Page 710.65P1
Page 710.64P1
Page 710.63P1
Page 710.0P1
Page 710.65PP
Page 710.64PP
Page 710.63PP
Page 710.0PP
Block 710.65B1
Block 710.64B1
Block 710.63B1
Block 710.0B1
Block 710.65BB
Block 710.64BB
Block 710.63BB
Block 710.0BB

Fig. 10

$$R0 = \sum_i Pi \qquad R1 = \sum_i Ki * Pi$$

Fig. 11

$$Pj = R0 - \sum_{i!=j} Pi$$

Fig. 12

$$R0' = \sum_{i!=m,n} Pi \qquad R1' = \sum_{i!=m,n} Ki * Pi$$

Fig. 13A

$$R0 = R0' + Pm + Pn$$

$$R1 = R1' + Km * Pm + Kn * Pn$$

Fig. 13B

$$\Delta R0 = R0 - R0' = Pm + Pn$$

$$\Delta R1 = R1 - R1' = Km * Pm + Kn * Pn$$

Fig. 13C

$$Pn = (\Delta R1 - \Delta R0 * Km) / (Kn - Km)$$

$$Pm = \Delta R0 - Pn$$

Fig. 13D

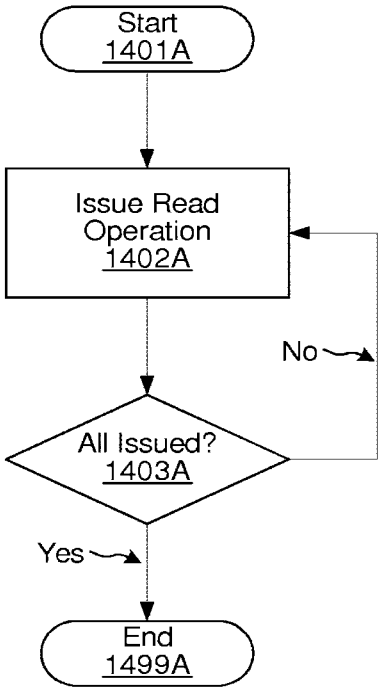


Fig. 14A

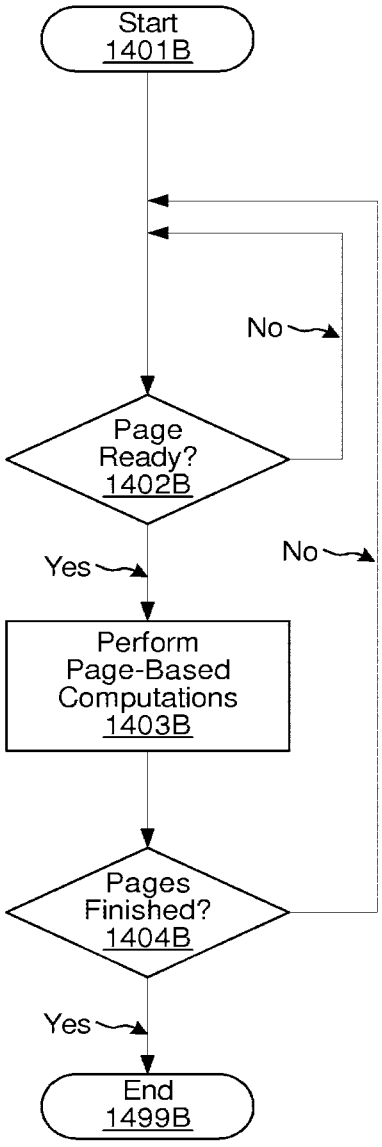


Fig. 14B

15/15

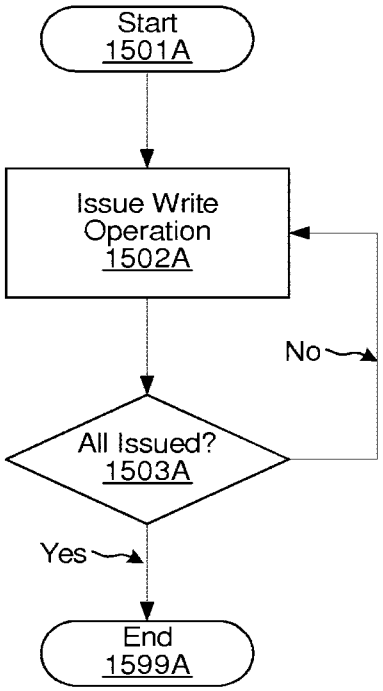


Fig. 15A

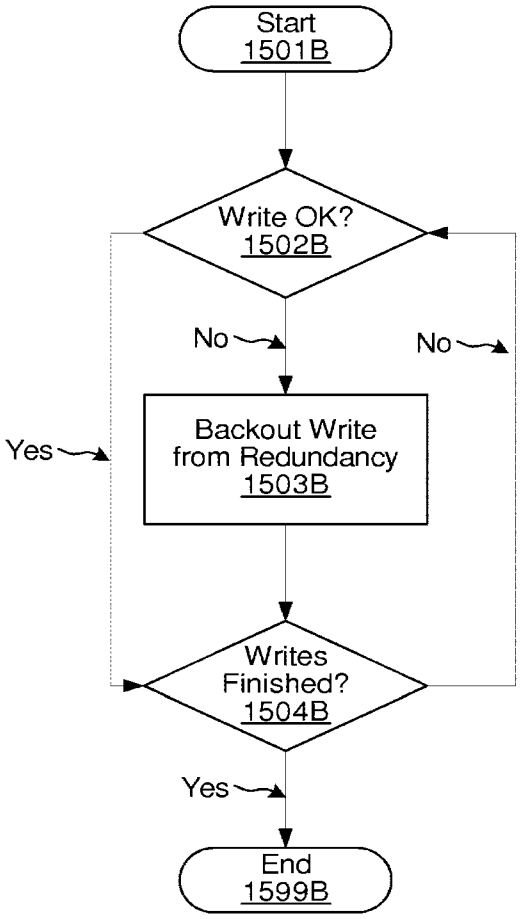


Fig. 15B

$$nR0 = R0 - Pj$$
$$nR1 = R1 - Kj * Pj$$

Fig. 15C

Electronic Acknowledgement Receipt

EFS ID:	14748676
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/Lori Larson
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	21-JAN-2013
Filing Date:	22-MAY-2012
Time Stamp:	15:15:42
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		IDS_DATUMTE076A1.pdf	79122 9e2fb47dc24dbb9998c8dc727fac36e8818df9d1	yes	2

	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Transmittal Letter		1	1	
	Information Disclosure Statement (IDS) Form (SB08)		2	2	
Warnings:					
Information:					
2	Foreign Reference	FP1_DATUMTE076A1.pdf	2747385 9ffae261403af6088cf61caa83d6b7caa92df404	no	54
Warnings:					
Information:					
3	Foreign Reference	FP2_DATUMTE076A1.pdf	4201664 cfeec40264393c2971a5f0d10e77ec6e671cf36e	no	100
Warnings:					
Information:					
		Total Files Size (in bytes):	7028171		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Docket No.: DATUMTE.076A1 **Customer No. 20995**

INFORMATION DISCLOSURE STATEMENT

Inventor	:	Philip L. Northcott
App. No.	:	13/477,600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
Examiner	:	DECADY, ALBERT
Art Unit	:	2112
Conf. No.	:	2672

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Submitted herewith in the above-identified application is an Information Disclosure Statement listing references for consideration. Copies of any listed foreign and non-patent literature references are being submitted.

Timing of Disclosure

This Information Disclosure Statement is being filed before the receipt of a first Office Action on the merits, and presumably no fee is required. If a first Office Action on the merits was mailed before the mailing date of this Statement, the Commissioner is authorized to charge the fee set forth in 37 CFR 1.17(p) to Deposit Account No. 11-1410.

Respectfully submitted,
KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: Jan. 21, 2013

By: Michael Okamoto

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

14698652 / 012113

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477,600
	Filing Date	May 22, 2012
	First Named Inventor	WANG, SHEN
	Art Unit	2117
(Multiple sheets used when necessary)	Examiner	Decady, Albert
SHEET 1 OF 1	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,675,318 B1	01-06-2004	Lee, Whay S.	
	2	8,335,951 B2	12-18-2012	Becker et al.	
	3	8,347,138, B2	01-01-2013	Moshayedi, Mark	
	4	8,392,791 B2	03-05-2013	Saliba et al.	
	5	8,433,979 B2	04-30-2013	Blaum et al.	
	6	2010/0281341 A1	11-04-2010	WU et al.	
	7	2012/0260146 A1	10-11-2012	LEE et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T^1

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T^1
	8	USPTO; Office Action dated July 24, 2013, from related application U.S. Patent Application No. 13/477,635 (filed May 22, 2012).	

15964224
080713

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T^1 - Place a check mark in this area when an English language Translation is attached.

Electronic Acknowledgement Receipt

EFS ID:	16536585
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto
Filer Authorized By:	
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	08-AUG-2013
Filing Date:	22-MAY-2012
Time Stamp:	14:11:44
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		IDS_DATUMTE076A1.pdf	85914 600fef1559f14ec74b92a9ac24c79467a9aef54e	yes	2

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Transmittal Letter	1	1
	Information Disclosure Statement (IDS) Form (SB08)	2	2
Warnings:			
Information:			

2	Non Patent Literature	NPL_DATUMTE076A1.pdf	700434 0c8c10892f66d02dae802df87e8846bb54f33b01	no	15
Warnings:					
Information:					
		Total Files Size (in bytes):	786348		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

INFORMATION DISCLOSURE STATEMENT

Inventor	:	Philip L. Northcott
App. No.	:	13/477,600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
Examiner	:	WANG, SHEN
Art Unit	:	2117
Conf. No.	:	2672

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Submitted herewith in the above-identified application is an Information Disclosure Statement listing references for consideration. Copies of any listed foreign and non-patent literature references are being submitted.

Timing of Disclosure

This Information Disclosure Statement is being filed before the receipt of a First Office Action on the merits, and presumably no fee is required. If a First Office Action on the merits was mailed before the mailing date of this Statement, the Commissioner is authorized to charge the fee set forth in 37 CFR 1.17(p) to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: Aug. 8, 2013

By: Michael S. Okamoto

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

15964210 / 080713

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

20995 7590 01/02/2014

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
DORMAN, CHRISTIAN M	
ART UNIT	**PAPER NUMBER**
4181	

NOTIFICATION DATE	DELIVERY MODE
01/02/2014	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

jayna.cartee@knobbe.com
efiling@knobbe.com

Office Action Summary	Application No. 13/477,600	Applicant(s) NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 4181	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *5/22/2012*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *1-30* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-30* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☒ The specification is objected to by the Examiner.

11)☒ The drawing(s) filed on *5/22/2012* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date *5/22/2012,1/21/2013,8/8/2013*.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

4) ☐ Other: _____.

DETAILED ACTION

1. This office action is sent in response to Applicant's Communication received on May 22nd, 2012 for application number 13/477600. The office hereby acknowledges receipt of the following and placed of record in file: Specification, Drawings, Abstract, Oath/Declaration, and Claims.

2. Claims 1-30 are presented for examination.

Notice of Pre-AIA or AIA Status

3. The present application is being examined under the pre-AIA first to invent provisions.

Information Disclosure Statement

4. The information disclosure statements (IDS) submitted on 5/22/2012, 1/21/2013, and 8/08/2013 were filed before the mailing date of the first office action. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

Specification

5. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

6. The abstract of the disclosure is objected to because the abstract does not adequately describe the function of the invention. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details. Correction is required. See MPEP § 608.01(b).

Claim Rejections - 35 USC § 112

7. The following is a quotation of 35 U.S.C. 112(b):
(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:
The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 4, 6, and 16 are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention.

9. Regarding claims 4 and 16, it is claimed that determining, comparing, and selecting are performed concurrently with at least one of flash initialization, read, or write operations for an end user. This claim does not follow with any particular notion of how this is performed in the specification and is contrary to figure 11 in how each step is performed iteratively. It is not seen how these steps could be performed at the same time as each being performed is dependent on one of the others.

10. Regarding claim 6, it is not seen why the gear that is selected should have the highest data payload size. It is seen in table III of the specification, though, how in that particular embodiment of the disclosed invention that a higher data payload size is used when there is less error correcting ability. So, it makes sense according to that embodiment that the claim could correspond to meaning using the least amount of error correcting ability as is necessary to still achieve the desired aUBER, but as written in

claim 6 this is indefinite. This claim is interpreted as the highest data payload size being used corresponding to having less error correcting ability in further rejections.

11. The following is a quotation of 35 U.S.C. 112(f):

> (f) Element in Claim for a Combination. – An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

The following is a quotation of pre-AIA 35 U.S.C. 112, sixth paragraph:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

12. Use of the word “means” (or “step for”) in a claim with functional language creates a rebuttable presumption that the claim element is to be treated in accordance with 35 U.S.C. 112(f) (pre-AIA 35 U.S.C. 112, sixth paragraph). The presumption that 35 U.S.C. 112(f) (pre-AIA 35 U.S.C. 112, sixth paragraph) is invoked is rebutted when the function is recited with sufficient structure, material, or acts within the claim itself to entirely perform the recited function.

Absence of the word “means” (or “step for”) in a claim creates a rebuttable presumption that the claim element **is not** to be treated in accordance with 35 U.S.C. 112(f) (pre-AIA 35 U.S.C. 112, sixth paragraph). The presumption that 35 U.S.C. 112(f) (pre-AIA 35 U.S.C. 112, sixth paragraph) is not invoked is rebutted when the claim element recites function but fails to recite sufficiently definite structure, material or acts to perform that function.

Claim elements in this application that use the word “means” (or “step for”) are presumed to invoke 35 U.S.C. 112(f) except as otherwise indicated in an Office action. Similarly, claim elements that do not use the word “means” (or “step for”) are presumed not to invoke 35 U.S.C. 112(f) except as otherwise indicated in an Office action.

13. The claim limitations:

a. A means for determining the bit error rate… (claim 24)

b. A means for comparing the determined bit error rate… (claim 24)

c. A means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds (claim 24)

d. A means for reading data from the region of flash memory… (claim 25)

e. A means for estimating the bit error rate… (claim 25)

f. A means for selecting a gear having a smaller data payload size... (claim 26)

g. A means for receiving a command to perform formatting (claim 27)

h. A means for performing a memory test... (claim 27)

have been interpreted under 35 U.S.C. 112, sixth paragraph, because it uses a non-structural term “means” coupled with functional language “for” without reciting sufficient structure to achieve the function. Furthermore, the non-structural term is not preceded by a structural modifier.

Since these claim limitation invokes 35 U.S.C. 112, sixth paragraph, claims 24-27 are interpreted to cover the corresponding structure described in the specification that achieves the claimed function, and equivalents thereof.

A review of the specification shows that the following appears to be the corresponding structure described in the specification for the 35 U.S.C. 112, sixth paragraph limitation:

i. A means for determining the bit error rate… (claim 24) --- an integrated circuit, controller

j. A means for comparing the determined bit error rate… (claim 24) --- an integrated circuit, controller

k. A means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds (claim 24) --- an integrated circuit, controller

l. A means for reading data from the region of flash memory… (claim 25) --- an integrated circuit, controller

m. A means for estimating the bit error rate… (claim 25) --- an integrated circuit, controller

n. A means for selecting a gear having a smaller data payload size... (claim 26) --- an integrated circuit, controller

o. A means for receiving a command to perform formatting (claim 27) --- an integrated circuit, controller

p. A means for performing a memory test... (claim 27) --- an integrated circuit, controller

If applicant wishes to provide further explanation or dispute the examiner's interpretation of the corresponding structure, applicant must identify the corresponding structure with reference to the specification by page and line number, and to the drawing, if any, by reference characters in response to this Office action.

If applicant does **not** wish to have the claim limitation treated under 35 U.S.C. 112, sixth paragraph, applicant may amend the claim so that it will clearly not invoke 35 U.S.C. 112, sixth paragraph, or present a sufficient showing that the claim recites sufficient structure, material, or acts for performing the claimed function to preclude application of 35 U.S.C. 112, sixth paragraph.

For more information, see *Supplementary Examination Guidelines for Determining Compliance with 35 U.S.C. § 112 and for Treatment of Related Issues in Patent Applications*, 76 FR 7162, 7167 (Feb. 9, 2011).

Claim Rejections - 35 USC § 103

14. The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 1-3, 5, 8, 9, 15, 17, 19, 20, 23-27, and 30 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651) in view of Manning (U.S. Patent Application Publication 2013/0246891).

16. Regarding claim 1, Diggs teaches a method of selecting an error correction coding (ECC) scheme, the method comprising: determining [by monitoring] a bit error rate associated with a region [paragraph 0029, lines 10-13 (the bit error rate of data is monitored)] associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]; comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different

data payload sizes)]; and selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used)]; wherein determining, comparing, and selecting are performed by an integrated circuit [paragraph 0021, lines 8-10 (controller is implemented as an integrated circuit), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)].

While Diggs does encompass the entirety of the claim, in teaching that that data blocks of different size and format can be used it is not explicitly taught that integer fractions of pages can be used. Furthermore, in teaching this it can be construed that different data payload for different ECC levels could be used, but this is also not explicitly taught by Diggs.

Manning teaches wherein portions of pages can be used for error correction encoding [figure 2 and paragraph 0032 (portions of pages are error coded)]. Manning also teaches wherein the data payload size can be programmed based on the error information corresponding to it [paragraph 0023, lines 13-15]. Therefore, different levels of ECC would have different data payload sizes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used.

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

17. Regarding claim 2, Diggs discloses the method of claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds such that the bit error rate is actively monitored [paragraph 0031, lines 1-4 (the controller can be set up to repeatedly monitor the data and set the ECC level based on the reading), paragraph 0051 and paragraph 0056 (the data monitored can be the bit error rate monitored and can be compared to threshold values)].

18. Regarding claim 3, Diggs discloses the method of claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] at least once every 4 program/erase cycles [paragraph 0045, lines 5-11 (the block erase count can be monitored) and paragraph 0031, lines 1-4 (events can be used to monitor the data and set the ECC level periodically)].

19. Regarding claim 5, Diggs discloses the method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region [paragraph

0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

20. Regarding claim 8, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations (paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs wherein selecting further comprises selecting a gear having a smaller payload size and a higher amount of correction capability based at least partly on the number of decoding iterations used to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

21. Regarding claim 9, Diggs discloses the method of claim 1, further comprising: receiving a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; performing a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

22. Regarding claim 15, Diggs teaches an apparatus comprising: a first circuit [monitoring circuit] configured to determine a bit error rate [paragraph 0029, lines 10-13 (the bit error rate of data is monitored) and paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] associated with a region associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]

and to compare the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and a second circuit [controller] configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used) and paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)].

While Diggs does encompass the entirety of the claim, in teaching that that data blocks of different size and format can be used it is not explicitly taught that integer fractions of pages can be used. Furthermore, in teaching this it can be construed that

different data payload for different ECC levels could be used, but this is also not explicitly taught by Diggs.

Manning teaches wherein portions of pages can be used for error correction encoding [figure 2 and paragraph 0032 (portions of pages are error coded)]. Manning also teaches wherein the data payload size can be programmed based on the error information corresponding to it [paragraph 0023, lines 13-15]. Therefore, different levels of ECC would have different data payload sizes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the apparatus of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used.

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

23. Regarding claim 17, Diggs discloses the apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

24. Regarding claim 19, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations (paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the apparatus of Diggs wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

25. Regarding claim 20, Diggs discloses the apparatus of Claim 15, wherein the processor is further configured to: receive a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; perform a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

26. Regarding claim 23, Diggs discloses the apparatus of claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive [paragraphs 0061 and 0062 (show how the apparatus could be embodied in this way) and claim 14].

27. Regarding claim 24, Diggs teaches an apparatus [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for selecting an error correction coding (ECC) scheme, the apparatus comprising: a means for determining a bit error rate [paragraph 0029, lines 10-13 (the bit error rate of data is monitored)] associated with a region associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and

paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]; a means for comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used)].

While Diggs does encompass the entirety of the claim, in teaching that that data blocks of different size and format can be used it is not explicitly taught that integer fractions of pages can be used. Furthermore, in teaching this it can be construed that different data payload for different ECC levels could be used, but this is also not explicitly taught by Diggs.

Manning teaches wherein portions of pages can be used for error correction encoding [figure 2 and paragraph 0032 (portions of pages are error coded)]. Manning also teaches wherein the data payload size can be programmed based on the error information corresponding to it [paragraph 0023, lines 13-15]. Therefore, different levels of ECC would have different data payload sizes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the apparatus of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used.

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

28. Regarding claim 25, Diggs discloses the apparatus [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

29. Regarding claim 26, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations [paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the apparatus of Diggs wherein the selecting means further comprises means [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

30. Regarding claim 27, Diggs discloses the apparatus of Claim 24, further comprising: means [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for receiving a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; means for performing a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

31. Regarding claim 30, Diggs discloses the apparatus of claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive [paragraphs 0061 and 0062 (show how the apparatus could be embodied in this way) and claim 14].

32. Claims 6, 7, 10-13, 18, 21, and 28 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651) in view of Manning (U.S. Patent Application Publication 2013/0246891) as applied to

claims 1, 15, and 24 above, further in view of Bennett (U.S. Patent Application Publication 2013/0080862).

33. Regarding claim 6, Diggs teaches wherein a level of ECC is encoded into a given set of data dependent on comparing information monitored from said data with certain threshold values, where the threshold can be the bit error rate. [paragraph 0008, lines11-15 and paragraph 0029, lines 12-13]. Diggs, however, does not explicitly teach wherein the highest data payload size is used that still has sufficient correction ability for the determined bit error rate.

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the higher the data payload size, the better the memory density (ratio of data to ECC bits) and the worse the error correcting ability (because of the higher data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs so that selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate.

Motivation to do so would have been to achieve the best memory density possible [paragraph 0023, lines 19-23].

Neither Diggs nor Manning teach wherein there is a predetermined acceptable uncorrectable bit error rate.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Bennets to the modified method of Diggs with Manning wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

34. Regarding claim 7, Diggs and Manning fail to teach where the aUBER is 1E-16.

Bennett teaches wherein the aUBER can be 1E-18 and furthermore can be set to required level [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Bennett to the method of Diggs and Manning such that the aUBER is 1E-16.

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

35. Regarding claim 10, Diggs further teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs

0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs and Manning fail to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the method of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that

the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

36. Regarding claim 11, Diggs further teaches wherein the threshold values used can be modified by different functions and set to any value necessary to achieve the desired level of error correction [paragraphs 0050 and 0056]. Thereby, it could be had the method of claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor SFx is in a range from 1.5 to 4.

37. Regarding claim 12, Diggs further teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056]. Furthermore Diggs teaches that a high monitored bit error rate anticipates a high probability of future errors occurring and because of this the ECC setting is adjusted [paragraph 0051]. This information corresponds to the excursion factor times the retention factor, which is the safety factor, because they both anticipate the increased probability of future errors occuring.

Diggs and Manning fail to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate that is set to a required value [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs and Manning with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction and have this value be divided by a safety factor, where the safety factor is found by the given expression as a product of the excursion factor and the retention factor. Diggs teaches where these threshold values can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system and the safety factor is itself found as a product of the excursion factor and the retention factor. Thereby, the method of Diggs would be modified by the teachings of Bennett wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor, where the safety factor is equal to the excursion factor times the retention factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

38. Regarding claim 13, Diggs teaches wherein the threshold values used can be modified by different functions and set to any value necessary to achieve the desired level of error correction [paragraphs 0050 and 0056]. Thereby, it can be had the method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

39. Regarding claim 18, Diggs teaches wherein a level of ECC is encoded into a given set of data dependent on comparing information monitored from said data with certain threshold values, where the threshold can be the bit error rate. [paragraph 0008, lines11-15 and paragraph 0029, lines 12-13]. Diggs, however, does not explicitly teach wherein the highest data payload size is used that still has sufficient correction ability for the determined bit error rate.

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the higher the data payload size, the better the memory density (ratio of data to ECC bits) and the worse the error correcting ability (because of the higher data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs so that selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate.

Motivation to do so would have been to achieve the best memory density possible [paragraph 0023, lines 19-23].

Diggs and Manning fail to teach wherein there is a predetermined acceptable uncorrectable bit error rate.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Manning and Bennets to the apparatus of Diggs wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

40. Regarding claim 21, Diggs teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs fails to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by

multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the apparatus of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

41. Regarding claim 28, Diggs teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be

based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs fails to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the apparatus of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety

factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

42. Claims 14, 22, and 29 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651) in view of Manning (U.S. Patent Application Publication 2013/0246891) as applied to claims 1, 15, and 24 above, further in view of Cherubini (U.S. Patent Application Publication 2013/0179754).

Regarding claim 14, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the method of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC

decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

43. Regarding claim 22, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

44. Regarding claim 29, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTIAN DORMAN whose telephone number is (571)272-7657. The examiner can normally be reached on Monday-Friday 9:30am-3pm, Friday alternating.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert Decady can be reached on 571-272-3819. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHRISTIAN DORMAN/
Examiner, Art Unit 4181

/NITIN PATEL/
Supervisory Patent Examiner, Art Unit 4181

Notice of References Cited	Application/Control No. 13/477,600	Applicant(s)/Patent Under Reexamination NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 4181	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2009/0070651	03-2009	Diggs et al.	714/752
*	B	US-2013/0246891	09-2013	Manning et al.	714/773
*	C	US-2013/0179754	07-2013	Cherubini et al.	714/773
*	D	US-2013/0080862	03-2013	BENNETT, Jon C.R.	714/782
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 1 OF 4	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	4,112,502	09-05-1978	James Herman Scheuneman	
	2	5,404,485	04-04-1995	Ban, Amir	
	3	5,942,004	08-24-1999	Paolo Cappelletti	
	4	5,956,743	09-21-1999	Bruce, et al.	
	5	6,023,781	02-08-2000	Katsuki Hazama	
	6	6,279,133	08-21-2001	Vafai, et al.	
	7	6,591,394	07-08-2003	Lee, et al.	
	8	6,628,723	09-30-2003	Gerlach, et al.	
	9	6,731,538	05-04-2004	Noda, et al.	
	10	6,732,322	05-04-2004	Miyauchi, et al.	
	11	6,839,870	01-04-2005	Fanfelle	
	12	6,871,303	03-22-2005	Steven J. Halter	
	13	6,895,543	05-17-2005	Katsuki Hazama	
	14	7,061,804	06-13-2006	Chun, et al.	
	15	7,184,356	02-27-2007	Noguchi, et al.	
	16	7,389,465	06-17-2008	Radke, et al.	
	17	7,451,383	11-11-2008	Kim et al.	
	18	7,478,314	01-13-2009	Cheong, et al.	
	19	7,643,342	01-05-2010	Litsyn, et al.	
	20	7,739,576	06-15-2010	William H. Radke	
	21	7,809,900	10-05-2010	Radoslav Danilak	
	22	7,844,879	11-30-2010	Ramamoorthy, et al.	
	23	7,860,200	12-28-2010	Furman et al.	
	24	7,904,619	03-08-2011	Radoslav Danilak	
	25	7,904,672	03-08-2011	Radoslav Danilak	
	26	7,904,780	03-08-2011	Brandman, Yigal	
	27	7,962,831	06-14-2011	Park, et al.	
	28	8,046,542	10-25-2011	William H. Radke	
	29	8,051,358	11-01-2011	William H. Radke	

Examiner Signature	/Christian Dorman/	Date Considered	12/19/2013

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STÁTEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 2 OF 4	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	30	8,065,583	11-22-2011	William H. Radke	
	31	8,090,980	01-03-2012	Radoslav Danilak	
	32	8,145,855	03-27-2012	Wan, et al.	
	33	2002/0120820	08-29-2002	Higuchi, et al.	
	34	2002/0159285	10-31-2002	Morley, et al.	
	35	2003/0088821	05-08-2003	Yokokawa et al.	
	36	2003/0112879	06-19-2003	Antia et al.	
	37	2003/0156454	08-21-2003	Wei, et al.	
	38	2004/0268065	12-30-2004	Hilton, et al.	
	39	2005/0114587	05-26-2005	Chou, et al.	
	40	2006/0221752	10-05-2006	Fasoli, et al.	
	41	2007/0061689	03-15-2007	Park, et al.	
	42	2007/0124647	05-31-2007	Chen, et al.	
	43	2007/0171714	07-26-2007	Wu, et al.	
	44	2007/0171730	07-26-2007	Ramamoorthy, et al.	
	45	2007/0266295	11-15-2007	Kevin M. Conley	
	46	2007/0266296	11-15-2007	Kevin M. Conley	
	47	2008/0010582	01-10-2008	Nieto, et al.	
	48	2008/0034272	02-07-2008	Wu, et al.	
	49	2008/0137414	06-12-2008	Park, et al.	
	50	2008/0163023	07-03-2008	Hong, et al.	
	51	2008/0168319	07-10-2008	Lee, et al.	
	52	2008/0294960	11-27-2008	Sharon et al.	
	53	2008/0320361	12-25-2008	Fukuda, et al.	
	54	2008/0320373	12-25-2008	Kim et al.	
	55	2009/0013233	01-08-2009	William H. Radke	
	56	2009/0013234	01-08-2009	William H. Radke	
	57	2009/0037627	02-05-2009	Rofougaran, Ahmadreze (Reza)	
	58	2009/0100307	04-16-2009	Lee, Schweiray Joseph	

Examiner Signature	/Christian Dorman/	Date Considered	12/19/2013

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 3 OF 4	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	59	2009/0132889	05-21-2009	William H. Radke	
	60	2009/0327589	12-31-2009	Mark Moshayedi	
	61	2009/0327840	12-31-2009	Mark Moshayedi	
	62	2010/0023800	01-28-2010	Harari, et al.	
	63	2010/0115376	05-06-2010	Shalvi, et al.	
	64	2010/0199153	08-05-2010	Okamura et al.	
	65	2011/0019475	01-27-2011	Mark Moshayedi	
	66	2011/0029716	02-03-2011	Mark Moshayedi	
	67	2011/0060969	03-10-2011	Ramamoorthy, et al.	
	68	2011/0167199	07-07-2011	Radoslav Danilak	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]
	69	JP 2000149592	05-30-2000	NEC		English Abstract
	70	KR 100766042	10-04-2007	Samsung		English Abstract only
	71	WO 2006070668	07-06-2006	IBM		English Abstract
	72	WO 2004062113	07-22-2004	Interdigital Tech Corp.		
	73	WO 2007084751	07-26-2007	Marvell World Trade Ltd.		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	74	DANESHGARAN, et al.; "An extensive search for good punctured rate-k/(k+1) recursive convolutional codes for serially concatenated convolutional codes," Information Theory, IEEE Transactions on, vol.50, no.1, pp. 208-217, Jan. 2004.	
	75	FRENGER, et al.; "Rate-compatible convolutional codes for multirate DS-CDMA systems," Communications, IEEE Transactions on, vol.47, no 6, pp.828-836, Jun 1999.	

Examiner Signature	/Christian Dorman/	Date Considered	12/19/2013

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	Unknown
	Filing Date	On Even Date Herewith
	First Named Inventor	Northcott
	Art Unit	Unknown
(Multiple sheets used when necessary)	Examiner	Unknown
SHEET 4 OF 4	Attorney Docket No.	DATUMTE.076A1

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	76	INTEL CORPORATION; Understanding the Flash Translation Layer (FTL) Specification, Application Note AP-684; retrieved from the Internet <URL: http://staff.ustc.edu.cn/~jpq/paper/flash/2006-Intel%20TR-Understanding%20the%20flash%20translation%20layer%20%28FTL%29%20specification.pdf>; December 1998; retrieved January 26, 2012; Intel Corporation.	
	77	JIM HANDY; Does MLC Flash Belong in Enterprise SSDs; retrieved from the Internet <URL: http://www.infostor.com/index/articles/display/3214572139/articles/infostor/volume-14/issue-1/special-report/does-mlc_flash_belong.html>; February 1, 2010; retrieved on January 24, 2012.	
	78	OHTSUKI, T.; "Rate adaptive indoor infrared wireless communication systems using repeated and punctured convolutional codes," Communications, 1999. ICC '99. 1999 IEEE International Conference on, vol.1, no., pp.609-613 vol.1, 1999.	
	79	SANDFORCE; Product Brief SF-1500 Enterprise SSD Processors; retrieved from the Internet <URL: http://www.sandforce.com/userfiles/file/downloads/LSI_SandForce_1500ENT_PB_120104.pdf>; 2012; retrieved January 24, 2012; LSI Corporation; Milpitas, CA.	
	80	SANDFORCE; Product Brief SF-2500 & SF-2600 Enterprise SSD Processors; retrieved from the Internet <URL: http://www.sandforce.com/userfiles/file/downloads/LSI_SandForce_2500-2600ENT_PB_120104.pdf>; 2012; retrieved on January 24, 2012; LSI Corporation; Milpitas, CA.	
	81	SANDFORCE; RAISE Improves Total SSD Reliability; retrieved from the Internet <URL: http://sandforce.com/index.php?id=174&parentId=3>; 2012; retrieved on January 24, 2012.	
	82	SANDFORCE; SandForce SSD Processors Transform Mainstream Data Storage; retrieved from the Internet <URL: http://www.sandforce.com/userfiles/file/downloads/SFI_Launch_PR_Final.pdf> April 13, 2009; retrieved on Jan. 24, 2012; Saratoga, CA.	
	83	ZHOU, et al.; "High rate turbo code using unevenly punctured convolutional constituent code," Communications, 1999. APCC/OECC '99. Fifth Asia-Pacific Conference on... and Fourth Optoelectronics and Communications Conference, vol. 1, no., pp.751-754 vol.1, 1999.	
	84	STEC; Engineering MLC Flash-Based SSDs to Reduce Total Cost of Ownership in Enterprise SSD Deployments; retrieved from the Internet <URL: http://www.stec-inc.com/downloads/whitepapers/SAFE_WP.pdf>; 2011; STEC, Inc.; retrieved on January 24, 2012.	
	85	STEC; STEC Secure Array of Flash Elements Technology Improves Reliability of Enterprise-Class SSDs; retrieved from the Internet <URL: http://www.stec-inc.com/downloads/whitepapers/SAFE_WP.pdf>; 2011; STEC, Inc.; retrieved on January 24, 2012.	

13056094_1
052212

Examiner Signature	/Christian Dorman/	Date Considered	12/19/2013

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477,600
	Filing Date	May 22, 2012
	First Named Inventor	WANG, SHEN
	Art Unit	2117
(Multiple sheets used when necessary)	Examiner	Decady, Albert
SHEET 1 OF 1	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,675,318 B1	01-06-2004	Lee, Whay S.	
	2	8,335,951 B2	12-18-2012	Becker et al.	
	3	8,347,138, B2	01-01-2013	Moshayedi, Mark	
	4	8,392,791 B2	03-05-2013	Saliba et al.	
	5	8,433,979 B2	04-30-2013	Blaum et al.	
	6	2010/0281341 A1	11-04-2010	WU et al.	
	7	2012/0260146 A1	10-11-2012	LEE et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T^1

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T^1
	8	USPTO; Office Action dated July 24, 2013, from related application U.S. Patent Application No. 13/477,635 (filed May 22, 2012).	

15964224
080713

Examiner Signature	/Christian Dorman/	Date Considered	12/19/2013

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T^1 - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	130147	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:36
L2	891	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:36
L3	0	(uncorrect$4 adj error) same (1E-16)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:36
L4	2	(uncorrect$4 adj error) same (10^ -16 or 1E-16 or -16)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:37
L5	9	(US-20120144270-$ or US-20130080862-$ or US-20130246891-$ or US-20090070651-$ or US-20110051521-$ or US-20100131806-$ or US-20100131809-$ or US-20100088557-$).did. or (US-8305812-$).did.	US-PGPUB; USPAT	OR	ON	2013/12/19 09:39
L6	3	L5 and (uncorrect$4 adj error)	US-PGPUB; USPAT	OR	ON	2013/12/19 09:39
L7	2	L5 and (wear adj level)	US-PGPUB; USPAT	OR	ON	2013/12/19 10:00
L8	2	L5 and raw	US-PGPUB; USPAT	OR	ON	2013/12/19 10:11
L9	4	L5 and format	US-PGPUB; USPAT	OR	ON	2013/12/19 10:11
L10	20	((raw adj bit adj error adj rate) or rber) and (format$4 with command)	US-PGPUB; USPAT	OR	ON	2013/12/19 10:35
L11	3	L5 and uncorrect$5	US-PGPUB; USPAT	OR	ON	2013/12/19 10:44
L12	1	L5 and latency	US-PGPUB; USPAT	OR	ON	2013/12/19 10:50

L13	1	L5 and laten$4	US-PGPUB; USPAT	OR	ON	2013/12/19 10:50
L14	392	(latency with (option or parameter)) same (ECC or (error adj correction) or decode)	US-PGPUB; USPAT	OR	ON	2013/12/19 10:53
L15	1	L14 and L2	US-PGPUB; USPAT	OR	ON	2013/12/19 10:53
S1	886	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:08
S2	130050	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:09
S3	50326	(parameter same (chang$4 or determin$3 or depend$4 or adjust$4)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:13
S4	50338	(parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:14
S5	7928	(parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) and (error adj correct$4) and ((bit error rate) or BER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:15
S6	38	(parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) and (error adj correct$4) and ((bit error rate) or BER) and ((uncorrectable bit error rate) or UBER or aUBER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:16
S7	95	((parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:18
S8	1018	((chang$4 or determin$4 or depend$4 or adjust$4) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER)	US-PGPUB; USPAT; USOCR; FPRS;	ADJ	ON	2013/12/16 19:19

			EPO; JPO; DERWENT; IBM_TDB			
S9	1	((chang$4 or determin$4 or depend$4 or adjust$4) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER) and ((uncorrectable bit error rate) or UBER or aUBER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:19
S10	877	((chang$4 or determin$4 or depend$4 or adjust$4) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER) and memory	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:21
S11	71	S10 and S2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:22
S12	4	S10 and S1	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:22
S13	0	("2013/0318420").URPN.	USPAT	OR	ON	2013/12/16 19:33
S14	9	((integer adj fraction) with page) same memory	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 08:51
S15	0	((integer adj fraction) with page) same memory same (page stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 09:05
S16	36	(((fraction or portion or part) with page) same memory same (page stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 09:05
S17	887	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:05

S18	130116	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:05
S19	1	S16 and S17	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 09:06
S20	2	"20120144270".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:22
S21	2	"20090070651".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:30
S22	0	("2012/0144270").URPN.	USPAT	OR	ON	2013/12/18 09:31
S23	230	("20020063774" | "20020085419" | "20020154769" | "20030065876" | "20030101404" | "20030105620" | "20030192007" | "20040015771" | "20040030971" | "20040153722" | "20040153817" | "20040181735" | "20050013165" | "20050018482" | "20050083735" | "20050117401" | "20050120265" | "20050128811" | "20050138533" | "20050144213" | "20050144368" | "20050169057" | "20050172179" | "20050213393" | "20060059406" | "20060059409" | "20060064537" | "20060101193" | "20060203587" | "20060221692" | "20060248434" | "20060268608" | "20060294312" | "20070002633" | "20070025157" | "20070063180" | "20070103992" | "20070104004" | "20070109858" | "20070121386" | "20070124652" | "20070143561" | "20070150694" | "20070168625" | "20070171714" | "20070171730" | "20070180346" | "20070223277" | "20070226582" | "20070226592" | "20070228449" | "20070253249" | "20070253250" | "20070263439" | "20070266291" | "20070271494" | "20080010581" | "20080028014" | "20080055989" | "20080082897" | "20080092026" | "20080104309" | "20080116509" | "20080126686"	US-PGPUB; USPAT; USOCR	OR	ON	2013/12/18 09:31

		"20080127104" | "20080128790" | "20080130341" | "20080137413" | "20080148115" | "20080158958" | "20080159059" | "20080162079" | "20080168216" | "20080168320" | "20080181001" | "20080198650" | "20080198652" | "20080219050" | "20080225599" | "20080263262" | "20080282106" | "20080285351" | "20080301532" | "20090024905" | "20090043951" | "20090072303" | "20090091979" | "20090103358" | "20090106485" | "20090113275" | "20090125671" | "20090144600" | "20090150748").PN. OR ("20090157964" | "20090158126" | "20090168524" | "20090187803" | "20090199074" | "20090213653" | "20090213654" | "20090228761" | "20090240872" | "20100005270" | "20100058146" | "20100064096" | "20100088557" | "20100091535" | "20100095186" | "20100110787" | "20100115376" | "20100122113" | "20100124088" | "20100131580" | "20100131806" | "20100131809" | "20100131826" | "20100131827" | "20100131831" | "20100146191" | "20100146192" | "20100149881" | "20100180073" | "20100199149" | "20100211724" | "20100211833" | "20100211856" | "20100251066" | "20100253555" | "20100257309" | "20100293321" | "20110051521" | "20110055461" | "20110096612" | "20110119562" | "20110153919" | "20110161775" | "20110214029" | "20110214039" | "20110246792" | "20110246852" | "20110252187" | "20110252188" | "20110271043" | "20110302428" | "20120001778" | "20120005554" | "20120005558" | "20120005560" | "20120008401" | "20120008414" | "20120051144" | "20120063227" | "20120066441" | "20120110250" | "4463375" | "4584686" | "4589084" | "4866716" | "5077737" | "5297153" | "5657332" | "5729490" | "5793774" | "5926409" | "5956268" | "5982659" | "6038634" | "6094465" | "6119245" | "6182261" | "6192497" | "6195287" | "6199188" | "6209114" | "6259627" | "6278633" | "6279133" | "6301151" | "6370061" | "6374383" | "6504891" | "6532169" | "6532556" | "6553533" | "6560747" | "6637002" | "6639865" | "6674665" | "6704902" | "6751766" | "6772274" | "6781910" | "6792569" | "6873543" | "6891768" | "6914809" | "6915477").PN. OR ("6952365" | "6961890" | "6990012" | "6996004" | "6999854" | "7010739" | "7012835" | "7038950" | "7068539" | "7079436" | "7149950" | "7177977" | "7191379" |				

		"7196946" | "7203874" | "7290203" | "7292365" | "7301928" | "7441067" | "7466575" | "7533328" | "7558109" | "7593263" | "7697326" | "7706182" | "7804718" | "7805663" | "7805664" | "7844877" | "7961797" | "8020073" | "8122328").PN. OR ("8305812").URPN.				
S24	230	("20020063774" | "20020085419" | "20020154769" | "20030065876" | "20030101404" | "20030105620" | "20030192007" | "20040015771" | "20040030971" | "20040153722" | "20040153817" | "20040181735" | "20050013165" | "20050018482" | "20050083735" | "20050117401" | "20050120265" | "20050128811" | "20050138533" | "20050144213" | "20050144368" | "20050169057" | "20050172179" | "20050213393" | "20060059406" | "20060059409" | "20060064537" | "20060101193" | "20060203587" | "20060221692" | "20060248434" | "20060268608" | "20060294312" | "20070002633" | "20070025157" | "20070063180" | "20070103992" | "20070104004" | "20070109858" | "20070121386" | "20070124652" | "20070143561" | "20070150694" | "20070168625" | "20070171714" | "20070171730" | "20070180346" | "20070223277" | "20070226582" | "20070226592" | "20070228449" | "20070253249" | "20070253250" | "20070263439" | "20070266291" | "20070271494" | "20080010581" | "20080028014" | "20080055989" | "20080082897" | "20080092026" | "20080104309" | "20080116509" | "20080126686" | "20080127104" | "20080128790" | "20080130341" | "20080137413" | "20080148115" | "20080158958" | "20080159059" | "20080162079" | "20080168216" | "20080168320" | "20080181001" | "20080198650" | "20080198652" | "20080219050" | "20080225599" | "20080263262" | "20080282106" | "20080285351" | "20080301532" | "20090024905" | "20090043951" | "20090072303" | "20090091979" | "20090103358" | "20090106485" | "20090113275" | "20090125671" | "20090144600" | "20090150748").PN. OR ("20090157964" | "20090158126" | "20090168524" | "20090187803" | "20090199074" | "20090213653" | "20090213654" | "20090228761" | "20090240872" | "20100005270" | "20100058146" | "20100064096" | "20100088557" | "20100091535" | "20100095186" | "20100110787" | "20100115376" | "20100122113" | "20100124088" | "20100131580" | "20100131806"	US-PGPUB; USPAT; USOCR	OR	ON	2013/12/18 09:32

		"20100131809" | "20100131826" | "20100131827" | "20100131831" | "20100146191" | "20100146192" | "20100149881" | "20100180073" | "20100199149" | "20100211724" | "20100211833" | "20100211856" | "20100251066" | "20100253555" | "20100257309" | "20100293321" | "20110051521" | "20110055461" | "20110096612" | "20110119562" | "20110153919" | "20110161775" | "20110214029" | "20110214039" | "20110246792" | "20110246852" | "20110252187" | "20110252188" | "20110271043" | "20110302428" | "20120001778" | "20120005554" | "20120005558" | "20120005560" | "20120008401" | "20120008414" | "20120051144" | "20120063227" | "20120066441" | "20120110250" | "4463375" | "4584686" | "4589084" | "4866716" | "5077737" | "5297153" | "5657332" | "5729490" | "5793774" | "5926409" | "5956268" | "5982659" | "6038634" | "6094465" | "6119245" | "6182261" | "6192497" | "6195287" | "6199188" | "6209114" | "6259627" | "6278633" | "6279133" | "6301151" | "6370061" | "6374383" | "6504891" | "6532169" | "6532556" | "6553533" | "6560747" | "6637002" | "6639865" | "6674665" | "6704902" | "6751766" | "6772274" | "6781910" | "6792569" | "6873543" | "6891768" | "6914809" | "6915477").PN. OR ("6952365" | "6961890" | "6990012" | "6996004" | "6999854" | "7010739" | "7012835" | "7038950" | "7068539" | "7079436" | "7149950" | "7177977" | "7191379" | "7196946" | "7203874" | "7290203" | "7292365" | "7301928" | "7441067" | "7466575" | "7533328" | "7558109" | "7593263" | "7697326" | "7706182" | "7804718" | "7805663" | "7805664" | "7844877" | "7961797" | "8020073" | "8122328").PN. OR ("8305812").URPN.				
S25	1	"20110051521".pn.	US-PGPUB; USPAT; USOCR	OR	ON	2013/12/18 09:32
S26	9	(US-20120144270-$ or US-20130080862-$ or US-20130246891-$ or US-20090070651-$ or US-20110051521-$ or US-20100131806-$ or US-20100131809-$ or US-20100088557-$).did. or (US-8305812-$).did.	US-PGPUB; USPAT	OR	ON	2013/12/18 09:57
S27	5	S26 and (data adj payload)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:57

S28	5	S26 and payload	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:58
S29	1	S26 and payload and (page adj stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:02
S30	1	S26 and (page adj stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:02
S31	2473	(parameter same (chang$4 or determin$3 or depend$4 or adjust$4 or adapt$5)) and (error adj correct$4) and (data adj payload)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:05
S32	8	S26 and (BER or (bit adj error adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:13
S33	4	S26 and (UBER or aUBER or (uncorrectable adj bit adj error adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:13
S34	757	((data adj payload) with (chang$4 or determin$3 or depend$4 or adjust$4 or adapt$5)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:36
S35	1129	((data adj payload) with (chang$4 or determin$3 or depend$4 or adjust$4 or adapt$5 or different)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:37
S36	5	S35 and S17	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2013/12/18 10:37

			DERWENT; IBM_TDB			
S37	3	S34 and S17	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:39
S38	0	S34 and "L45"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:39
S39	90	S34 and S18	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:39
S40	9	(US-20120144270-$ or US-20130080862-$ or US-20130246891-$ or US-20090070651-$ or US-20110051521-$ or US-20100131806-$ or US-20100131809-$ or US-20100088557-$).did. or (US-8305812-$).did.	US-PGPUB; USPAT	OR	ON	2013/12/18 18:45
S41	887	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S42	130116	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S43	0	S40 and (auber)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S44	0	S40 and (uncorrectable adj bit)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S45	4	S40 and uber	US-PGPUB; USPAT; USOCR;	OR	ON	2013/12/18 18:47

			FPRS; EPO; JPO; DERWENT; IBM_TDB			
S46	4	S40 and (auber or uber or (uncorrect$4 adj bit))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:48
S47	9483	(auber or uber or (uncorrect$4 adj bit adj error))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:53
S48	2	((allow$4 or accept$4) adj uncorrect$4 adj bit adj error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:53
S49	9225	(auber or uber or ((allow$4 or accept$4) adj uncorrect$4 adj bit adj error))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:54

12/ 19/ 2013 2:07:10 PM
C:\ Users\ cdorman\ Documents\ EAST\ Workspaces\ 13477600.wsp

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477,600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	DECADY, ALBERT
SHEET 1 OF 1	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	2012/0089767	04-12-2012	Lee, et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]
	2	WO 2012047500 A1	04-12-2012	Colgrove, et al.		
	3	WO 2012075200 A2	06-07-2012	Werner, et al.		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]

14698633
012113

Examiner Signature	/Christian Dorman/	Date Considered	12/19/2013

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

47	(adjustable ECC parameter)	423 Metadata + Full Text Dec. 18, 2013 15:08 UTC	
46	(((error correction) AND adjustable parameter) AND flash memory)	405 Metadata + Full Text Dec. 18, 2013 15:07 UTC	
45	((error correction) AND adjustable parameter)	8695 Metadata + Full Text Dec. 18, 2013 15:06 UTC	
44	(((adjustable) AND payload) AND error correction)	0 Metadata Dec. 18, 2013 15:05 UTC	
43	(((adjustable) AND data payload) AND error correction)	0 Metadata Dec. 18, 2013 15:05 UTC	
42	((adjustable parameter) AND error correction)	59 Metadata Dec. 18, 2013 15:04 UTC	
41	(((adjustable parameter) AND error correction) AND flash memory)	0 Metadata Dec. 18, 2013 15:03 UTC	
40	((error correction) AND adjustable)	145 Metadata Dec. 18, 2013 15:03 UTC	
39	adjustable error correction	145 Metadata Dec. 18, 2013 15:01 UTC	
38	adjustable ECC	18 Metadata Dec. 18, 2013 15:00 UTC	
37	adjustable ECC for flash memory	5935 Metadata Dec. 18, 2013 15:00 UTC	

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	4181

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/19/2013								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								
	21	✓								
	22	✓								
	23	✓								
	24	✓								
	25	✓								
	26	✓								
	27	✓								
	28	✓								
	29	✓								
	30	✓								

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2672

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
13/477,600	05/22/2012 RULE	714	4181	DATUMTE.076A1

APPLICANTS

INVENTORS

Philip L. Northcott, Coquitlam, CANADA;

**** CONTINUING DATA ***************************

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

06/01/2012

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ Yes ☑ No	☐ Yes ☑ No	/CHRISTIAN M DORMAN/ Examiner's Signature	☐ Initials	CANADA	16	30	3

ADDRESS

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614
UNITED STATES

TITLE

SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
1980	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	4181

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
714	773	12/16/2013, 12/18/2013, 12/19/2013	CD

SEARCH NOTES		
Search Notes	Date	Examiner
EAST US_PGPUB, USPAT, USOCR, FPRS, EPO, JPO, DERWENT, IBM_TDB - search results are attached	12/16/2013, 12/18/2013, 12/19/2013	CD
Inventor Name Search, PALM	12/16/2013, 12/19/2013	CD
Double Patenting Search, PALM	12/16/2013, 12/19/2013	CD
IEEE NPL Search	12/18/2013	CD

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor	:	Philip L. Northcott
App. No.	:	13/477,600
Filed	:	May 22, 2012
Examiner	:	Dorman, Christian M.
Art Unit	:	4181
Conf. No.	:	2672

Amendment and Response to Office Action of January 2, 2014

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully responds to the Office Action of January 2, 2014 as set forth below.

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Abstract begin on page 3 of this paper.

Amendments to the Claims are reflected in the listing of claims which begins on page 4 of this paper.

Summary of Interview begins on page 10 of this paper.

Remarks/Arguments begin on page 11 of this paper.

AMENDMENTS TO THE SPECIFICATION

Please replace the **title** with the following amended **title**:

SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY

Ex.1002 / Page 528 of 744
Sandisk Technologies, Inc. et al

AMENDMENTS TO THE ABSTRACT

Please replace the **Abstract** with the following replacement **Abstract**:

Apparatus and methods provide relatively low uncorrectable bit error rates, low write amplification, long life, fast and efficient retrieval, and efficient storage density such that a solid-state drive (SSD) can be implemented using relatively inexpensive MLC Flash for an enterprise storage application. Predefined gears correspond to different predefined ECC schemes. Based on an observed bit error rate, a gear from a set of predefined gears is selected for use for a particular region of memory. Each gear of the set of predefined gears includes a lower-latency ECC decode option and one or more higher-latency ECC decode options.

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method of selecting an error correction coding (ECC) scheme, the method comprising:

determining a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof;

comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity symbols alone and one or more higher-latency ECC decode options with secondary parity symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower-latency ECC decode option and the higher-latency ECC decode option are different for the first gear and the second gear, wherein the first gear has a different data payload size and correction capability than the second gear; and

selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds;

wherein determining, comparing, and selecting are performed by an integrated circuit.

2. (Original) The method of Claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds such that the bit error rate is actively monitored.

3. (Original) The method of Claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds at least once every 4 program/erase cycles.

4. (Currently amended) The method of Claim 1, wherein determining, comparing, and selecting are performed ~~concurrently~~ in connection with at least one of flash initialization, read, or write operations for an end user.

5. (Original) The method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

6. (Original) The method of Claim 1, wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

7. (Original) The method of Claim 6, wherein the aUBER is 1E-16.

8. (Original) The method of Claim 1, wherein selecting further comprises selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

9. (Original) The method of Claim 1, further comprising:

receiving a command to perform formatting;

performing a memory test to determine raw bit error rates among the regions of the flash drive; and

allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

10. (Original) The method of Claim 1, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

11. (Original) The method of Claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor *SFx* is in a range from 1.5 to 4.

12. (Original) The method of Claim 10, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx = ECC\_CAPx / (Ex * RETx).$$

13. (Original) The method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

14. (Currently amended) The method of Claim 1, ~~wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options,~~ wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

15. (Currently amended) An apparatus comprising:

a first circuit configured to determine a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof and to compare the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes, wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity symbols alone and one or more higher-latency ECC decode options with secondary parity symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower-latency ECC decode option and the higher-latency ECC decode option are different for the first gear and the second gear, wherein the first gear has a different data payload size and correction capability than the second gear; and

a second circuit configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

16. (Currently amended) The apparatus of Claim 15, wherein the ECC encoder is configured to determine the bit error rate and the processor is configured to compare and to select

~~concurrently~~ in connection with at least one of flash initialization, read, or write operations for an end user.

17. (Original) The apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region.

18. (Original) The apparatus of Claim 15, wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

19. (Original) The apparatus of Claim 15, wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

20. (Original) The apparatus of Claim 15, wherein the processor is further configured to:

receive a command to perform formatting;

perform a memory test to determine raw bit error rates among the regions of the flash drive; and

allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

21. (Original) The apparatus of Claim 15, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

22. (Currently amended) The apparatus of Claim 15, ~~wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options,~~ wherein

the gear is selected such that performance of ECC decoding with the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

23. (Original) The apparatus of Claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

24. (Currently amended) An apparatus for selecting an error correction coding (ECC) scheme, the apparatus comprising:

a means for determining a bit error rate associated with a region associated with a plurality of flash memory pages or integer fractions thereof;

a means for comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity symbols alone and one or more higher-latency ECC decode options with secondary parity symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower-latency ECC decode option and the higher-latency ECC decode option are different for the first gear and the second gear, wherein the first gear has a different data payload size and correction capability than the second gear; and

a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

25. (Original) The apparatus of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

26. (Original) The apparatus of Claim 24, wherein the selecting means further comprises means for selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

27. (Original) The apparatus of Claim 24, further comprising:

means for receiving a command to perform formatting;

means for performing a memory test to determine raw bit error rates among the regions of the flash drive; and

means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

28. (Original) The apparatus of Claim 24, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

29. (Currently amended) The apparatus of Claim 24, ~~wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options,~~ wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

30. (Original) The apparatus of Claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

Application No.: 13/477,600
Filing Date: May 22, 2012

SUMMARY OF INTERVIEW

Attendees, Date and Type of Interview

The interview was conducted by telephone on April 28, 2014 and attended by Examiner Christian Dorman, Examiner Albert Decady, and the undersigned, Michael S. Okamoto.

Exhibits and/or Demonstrations

None

Identification of Claims Discussed

1, 4, and 6.

Identification of Prior Art Discussed

None.

Proposed Amendments

Applicant proposed a clarifying amendment to Claim 1 based on Claim 14, and proposed a clarifying amendment to Claim 4.

Principal Arguments and Other Matters

The amendment to Claim 1 clarifies that there is at least a primary and secondary ECC. Applicant offered a clarifying amendment for Claim 4. Applicant argued that Claim 6 was not indefinite.

Results of Interview

No agreement was reached with respect to Claim 1.

Agreement was reached on the rejections under 35 U.S.C. § 112(b) of Claims 4 and 6.

REMARKS

The January 2, 2014 Office Action objects to the title and abstract; rejects Claims 4, 6, and 16 under 35 U.S.C. § 112, second paragraph; rejects Claims 1-3, 5, 8, 9, 15, 17, 19, 20, 23-27, and 30 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2009/0070651 by Diggs ("Diggs") in view of U.S. Patent Application Publication No. 2013/0246891 by Manning ("Manning"); rejects Claims 6, 7, 10-13, 18, 21, and 28 under 35 U.S.C. § 103(a) as being unpatentable over Diggs and Manning, further in view of U.S. Patent Application No. 2013/0080862 by Bennett; rejects Claims 14, 22, and 29 under 35 U.S.C. § 103(a) as being unpatentable over Diggs in view of Manning, further in view of U.S. Patent Application Publication No. 2013/0179754 by Cherubini. Applicant requests reconsideration of the rejections in view of the foregoing amendments and the following comments.

Objections to the Title and Abstract

Applicant has amended the Title and Abstract and respectfully requests the Examiner to withdraw the objections to the same.

Rejections of Claims 4, 6, and 16 under 35 U.S.C. § 112, second paragraph

Applicant has amended Claims 4 and 16, and respectfully submits that Claims 4 and 16 now comply with the requirements of 35 U.S.C. § 112, second paragraph. The amendments are the same as those agreed upon in the interview.

In addition, the Examiner agreed to withdraw the rejection to Claim 6 under 35 U.S.C. § 112, second paragraph.

Rejections of Claims 1, 15, and 24 under 35 U.S.C. § 103(a)

The Office Action rejects Claims 1, 15, and 24 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2009/0070651 by Diggs ("Diggs") in view of U.S. Patent Application Publication No. 2013/0246891 by Manning ("Manning"). Applicant has amended Claims 1, 15, and 24 to clarify distinctions over the cited references.

Claims 1, 15, and 24 have each been amended to recite "wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity symbols alone and one or more higher-latency ECC decode options with secondary parity

symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower latency ECC decode option and the higher latency ECC decode option are different for the first gear and the second gear."

Some of these features were previously recited in Claims 14, 22, and 29, which were rejected under 35 U.S.C. § 103(a) as being unpatentable over Diggs in view of Manning, further in view of U.S. Patent Application Publication No. 2013/0179754 by Cherubini and thus Applicant discusses Cherubini in connection with the rejections of Claims 1, 15, and 24.

Cherubini discusses latency in connection with a tradeoff with complexity and parallel processing. Applicant has clarified that the latency being referred to is from decoding of primary parity symbols alone versus with secondary parity and/or tertiary parity symbols. The latency referred to is of a different type. Further, there is no indication in Cherubini that the complexity or parallel processing can change within a memory device so that the alternatives described in Cherubini could not be options. Applicant respectfully submits that Diggs and Manning do not provide the missing elements. In addition, note that the ECC parameters for both the lower-latency decode option and the higher-latency decode option are different for the first gear and the second gear.

Therefore, the combination of Diggs, Manning, and Cherubini does not teach all the claimed elements. As stated in M.P.E.P. at § 2143.02, a "rationale to support a conclusion that a claim would have been obvious is that ***all the claimed elements were known in the prior art*** and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded nothing more than predictable results to one of ordinary skill in the art." M.P.E.P. § 2143.02; *see KSR v. Teleflex*, 550 U.S. at 398, 82 USPQ2d at 1395.

Accordingly, Applicant requests the Examiner to withdraw the rejections of Claims 1, 15, and 24, and to allow the same.

Rejection of Dependent Claims 2-3, 5-14, 17-23, and 25-30

Dependent Claims 2-3, 5-14, 17-23, and 25-30 depend from and further define Claims 1, 1, 15, and 24, respectively. The dependent claims recite numerous additional distinctions over the cited references.

Claims 2-3, 5-14, 17-23, and 25-30 add additional limitations that Applicants believe are not taught or suggested by the prior art. In addition, Applicants respectfully submit that dependent Claims 2-3, 5-14, 17-23, and 25-30 are patentable for at least the reasons described for Claims 1, 1, 15, and 24, respectively, and Applicants accordingly request allowance of Claims 2-3, 5-14, 17-23, and 25-30.

No Disclaimers or Disavowals

Although the present communication may include alterations to the application or claims, or characterizations of claim scope or referenced art, Applicant is not conceding in this application that previously pending claims are not patentable over the cited references. Rather, any alterations or characterizations are being made to facilitate expeditious prosecution of this application. Applicant reserves the right to pursue at a later date any previously pending or other broader or narrower claims that capture any subject matter supported by the present disclosure, including subject matter found to be specifically disclaimed herein or by any prior prosecution. Accordingly, reviewers of this or any parent, child or related prosecution history shall not reasonably infer that Applicant has made any disclaimers or disavowals of any subject matter supported by the present application.

Co-Pending Applications of Assignee

Applicant wishes to draw the Examiner's attention to the following co-pending applications of the present application's assignee.

Docket No.	Serial No.	Title (as filed)	Filed
DATUMTE.076A2	13/477,633	SYSTEMS AND METHODS FOR INITIALIZING REGIONS OF A FLASH DRIVE HAVING DIVERSE ERROR CORRECTION CODING (ECC) SCHEMES	May 22, 2012
DATUMTE.076A3	13/477,629	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING AMONG DIFFERENT ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A4	13/477,598	SYSTEMS AND METHODS FOR REDUNDANTLY STORING ERROR CORRECTION CODES IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A5	13/477,595	SYSTEMS AND METHODS FOR LOW	May 22,

Docket No.	Serial No.	Title (as filed)	Filed
		LATENCY, HIGH RELIABILITY ERROR CORRECTION IN A FLASH DRIVE	2012
DATUMTE.076A6	13/477,568	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A7	13/477,599	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A8	13/477,845	SYSTEMS AND METHODS FOR RECLAIMING FLASH BLOCKS OF A FLASH DRIVE	May 22, 2012
DATUMTE.076A9	13/477,635	SYSTEMS AND METHODS FOR REDUNDANTLY STORING DATA IN A FLASH DRIVE SUCH THAT A FAILED PAGE CAN BE TOLERATED	May 22, 2012
DATUMTE.076A10	13/477,601	SYSTEMS AND METHODS FOR RECOVERING DATA FROM FAILED PORTIONS OF A FLASH DRIVE	May 22, 2012

Please charge any additional fees, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: May 2, 2014

By: 

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

17807638_1
050214

Electronic Patent Application Fee Transmittal

Application Number:	13477600
Filing Date:	22-May-2012
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Filer:	Michael S. Okamoto/Kenneth Watanabe
Attorney Docket Number:	DATUMTE.076A1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 1 month with $0 paid	1251	1	200	200

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**200**

Electronic Acknowledgement Receipt	
EFS ID:	18934126
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/Valerie Jones
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	02-MAY-2014
Filing Date:	22-MAY-2012
Time Stamp:	20:40:27
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$200
RAM confirmation Number	6921
Deposit Account	111410
Authorized User	KNOBBE MARTENS OLSON AND BEAR
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees) Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		20140502_Amendment_DATUMTE076A1.pdf	601930 5d998c54e81c63a195ac8c3b74f0c69c311b3689	yes	14

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Specification	2	2
Abstract	3	3
Claims	4	9
Applicant summary of interview with examiner	10	10
Applicant Arguments/Remarks Made in an Amendment	11	14

Warnings:

Information:

2	Fee Worksheet (SB06)	fee-info.pdf	30577 8e187d942d3b05095a5c02889e70d9f9393b8541	no	2

Warnings:

Information:

Total Files Size (in bytes):	632507

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/477,600	Filing Date 05/22/2012	☒ To be Mailed

ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 = *		X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 = *		X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT **05/02/2014**

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 30	Minus	** 30	= 0	X $80 =	0
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $420 =	0
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/Katrina Harling/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	4181
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 2	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	5,592,497	01-07-1997	Lokhoff	
	2	5,974,544	10-26-1999	Jeffries	
	3	6,574,774	06-03-2003	Vasiliev, Peter I.	
	4	6,651,212	11-18-2003	Katayama, et al.	
	5	6,732,325	05-04-2004	Tash et al.	
	6	6,751,766	06-15-2004	Guterman, et al.	
	7	7,739,576	06-15-2010	Radke, William H.	
	8	7,783,955	08-24-2010	Mark Murin	
	9	7,944,748	05-17-2011	Eggleston, et al.	
	10	8,495,465	07-23-2013	Anholt, et al.	
	11	8,560,881	10-15-2013	Frost, et al.	
	12	8,612,680	12-17-2013	Madnani, et al.	
	13	8,621,318	12-31-2013	Micheloni, et al.	
	14	2007/0233752	10-04-2007	Bangalore, et al.	
	15	2007/0277066	11-29-2007	Gajapathy, et al.	
	16	2008/0172589	07-17-2008	Gallezot, et al.	
	17	2009/0182939	07-16-2009	Hluchyj, et al.	
	18	2009/0164836	06-25-2009	Richard Carmichael	
	19	2010/0008214	01-14-2010	Siaud, et al.	
	20	2010/0332894	12-30-2010	Bowers, et al.	
	21	2010/0332922	12-30-2010	Chang, et al.	
	22	2011/0040926	02-17-2011	Frost, et al.	
	23	2011/0167319	07-07-2011	Jeddeloh, Joe M.	
	24	2011/0213920	09-01-2011	Frost, et al.	
	25	2012/0072680	03-22-2012	Kimura, et al.	
	26	2013/0073895	03-21-2013	Earl T. Cohen	
	27	2013/0151914	06-13-2013	Cadigan	
	28	2013/0173955	07-04-2013	Hallak, et al.	

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	4181
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 2 OF 2	Attorney Docket No.	DATUMTE.076A1

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	29	USPTO; Office Action dated December 31, 2013, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	
	30	USPTO; Office Action dated January 15, 2014, from related application U.S. Patent Application No. 13/477,568 (filed May 22, 2012).	
	31	USPTO; Office Action dated January 21, 2014, from related application U.S. Patent Application No. 13/477,633 (filed May 22, 2012).	
	32	USPTO; Office Action dated February 28, 2014, from related application U.S. Patent Application No. 13/477,599 (filed May 22, 2012).	
	33	USPTO; Office Action dated May 7, 2014, from related application U.S. Patent Application No. 13/477,601 (filed May 22, 2012).	

16999441
050514

Examiner Signature	Date Considered
***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T[1] - Place a check mark in this area when an English language Translation is attached.

Electronic Patent Application Fee Transmittal

Application Number:	13477600
Filing Date:	22-May-2012
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Filer:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	18994441
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto
Filer Authorized By:	
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	09-MAY-2014
Filing Date:	22-MAY-2012
Time Stamp:	17:54:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$180
RAM confirmation Number	4142
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

1		IDS_DATUMTE076A1.pdf	366798 51eb6e4c7e25732fda282fe3c8731cfc5c657d71	yes	3
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	
	Transmittal Letter		1	1	
	Information Disclosure Statement (IDS) Form (SB08)		2	3	
Warnings:					
Information:					
2	Non Patent Literature	NPL1_DATUMTE076A1.pdf	1312595 9ed685ffb41b0ca5e11a9a77b80ac85fbdeb0c90	no	33
Warnings:					
Information:					
3	Non Patent Literature	NPL2_DATUMTE076A1.pdf	1041379 b53c728976eedcbc8c686295e348680c82a89b6c	no	23
Warnings:					
Information:					
4	Non Patent Literature	NPL3_DATUMTE076A1.pdf	1214322 16d3aac2dc9b9b25c31deb9521705fb9968ecc56	no	31
Warnings:					
Information:					
5	Non Patent Literature	NPL5_DATUMTE076A1.pdf	876767 62998f88a1b2ee63c76ddd81eeed4a24ff8d3ccdc	no	18
Warnings:					
Information:					
6	Non Patent Literature	NPL4_DATUMTE076A1.pdf	1085695 2528bfc1a6b022f6b02b1b74cd6bf77a36137e31	no	20
Warnings:					
Information:					
7	Fee Worksheet (SB06)	fee-info.pdf	30625 0bfff256e61ddf1909c1bcc2ea19f3db1c9a4a6e	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	5928181		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

INFORMATION DISCLOSURE STATEMENT

Inventor	:	Philip L. Northcott
App. No.	:	13/477600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE
Examiner	:	Dorman, Christian M.
Art Unit	:	2112
Conf. No.	:	2672

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Submitted herewith in the above-identified application is an Information Disclosure Statement listing references for consideration. Copies of any listed foreign and non-patent literature references are being submitted.

Timing of Disclosure

This Information Disclosure Statement is being filed after receipt of a First Office Action, but before the mailing date of a Final Action and before the mailing date of a Notice of Allowance. This Statement is accompanied by the fees set forth in 37 C.F.R. 1.17(p). The Commissioner is hereby authorized to charge any additional fees which may be required or to credit any overpayment to Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: May 9, 2014

By: Michael Okamoto

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

16999405 / 010914

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

20995 7590 07/16/2014

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
DORMAN, CHRISTIAN M	
ART UNIT	**PAPER NUMBER**
2112	

NOTIFICATION DATE	DELIVERY MODE
07/16/2014	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

jayna.cartee@knobbe.com
efiling@knobbe.com

Office Action Summary	Application No. 13/477,600	Applicant(s) NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 2112	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *5/2/2014*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *1-30* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-30* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☒ The drawing(s) filed on *5/22/2012* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date *5/09/2014*.

3) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *4/28/2014* .

4) ☐ Other: _____.

DETAILED ACTION

1. This office action is sent in response to Applicant's Communication received on May 2nd, 2014 for application number 13/477600. The office hereby acknowledges receipt of the following and placed of record in file: Specification, Drawings, Abstract, Oath/Declaration, and Claims.

2. Claims 1-30 are presented for examination.

Notice of Pre-AIA or AIA Status

3. The present application is being examined under the pre-AIA first to invent provisions.

Information Disclosure Statement

4. The information disclosure statement (IDS) submitted on 5/09/2014 was filed before the mailing date of this office action. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

Response to Amendment

5. The applicant's modifications to the title and abstract of the application have been accepted and the objections to those areas have been removed.

6. The examiner has withdrawn the rejections of Claims 4, 16, and 6 under 35 USC 112(b).

7. The applicant's amendments made to claims 1, 15, and 24 consisted of clarifying distinctions over the cited references wherein the distinction amended into these claims (same idea in each amendment) was in moving elements of claims 14, 22, and 29

respectively into claims 1, 15, and 24 of each gear including a lower latency decode option and one or more higher latency ECC decode options with the additional claim language of where the lower latency decode option consists of primary parity symbols alone and where the higher latency decode option(s) have secondary and/or tertiary parity symbols. Further added to the claims was wherein the ECC parameters for the lower latency and higher latency decode options for the first and second gears are different. The applicant argues that Diggs [U.S. Patent Application Publication 2009/0070651], in view of Manning [U.S. Patent Application Publication 2013/0246891], further in view of Cherubini [U.S. Patent Application Publication 2013/0179754], specifically Cherubini as this reference was used to reject the part of the subject matter that was moved into the independent claims from claims 14, 22, and 29, discusses latency in connection with a tradeoff with complexity and parallel processing. The examiner argued that Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084]. In order to overcome this rejection, the applicant clarified that the latency being referred to in the claims is from the decoding of primary parity symbols alone versus the decoding with secondary parity and/or tertiary parity symbols, as opposed to the latency as it is referred to in the Cherubini reference. The applicant further added to the claims that the ECC parameters for both the lower latency decode option and the higher latency decode option are different for the first gear and the second gear.

Thereby, the applicant's amendments made to claims 1, 15, and 24, filed 5/02/2014, have been fully considered and have been rejected upon the application of further prior art. Diggs, in view of Manning, in view of Cherubini, further in view of Anholt [U.S. Patent 8,495,465], wherein Anholt is the further prior art, teach the amended claim limitations. Specifically, Anholt teaches wherein for an error correction scheme there can be encoding of data where there is a first ECC level using primary parity and a second ECC level using secondary parity symbols that is higher latency and of stronger error correcting ability [Anholt, column 2, lines 44-48 and 64-68; column 9, lines 1-7 and 44-47]. Further, because Diggs was already cited to teach wherein the gears used for encoding would have different levels of error correcting ability [Diggs, paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)], it would have been obvious to one of ordinary skill in the art at the time of the invention that if the gears where to consist of multiple decode options as taught by Anholt then the different options could have different ECC parameters in order to retain the varying levels of correction ability for each gear [Anholt, column 9, lines 36-40, any suitable ECC, including LDPC could be used for the ECC2; column 6, lines 48-56, the ECC may be any of multiple options].

8. The rejection is maintained, see below.

Claim Rejections - 35 USC § 103

9. The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-5, 8, 9, 14-16, 17, 19, 20, 22-27, 29, and 30 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651), in view of Manning (U.S. Patent Application Publication 2013/0246891), in view of Cherubini (U.S. Patent Application Publication 2013/0179754), further in view of Anholt [U.S. Patent 8,495,465].

Regarding claim 1, Diggs teaches a method of selecting an error correction coding (ECC) scheme, the method comprising: determining [by monitoring] a bit error rate associated with a region [paragraph 0029, lines 10-13 (the bit error rate of data is monitored)] associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]; comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different

predefined ECC schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used)]; wherein determining, comparing, and selecting are performed by an integrated circuit [paragraph 0021, lines 8-10 (controller is implemented as an integrated circuit), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)].

While Diggs does encompass the entirety of the claim in teaching that data blocks of different size and format can be used, it is not explicitly taught that integer fractions of pages can be used. Furthermore, in teaching this it can be construed that different data payload for different ECC levels could be used, but this is also not explicitly taught by Diggs.

Manning teaches wherein portions of pages can be used for error correction encoding [figure 2 and paragraph 0032 (portions of pages are error coded)]. Manning

also teaches wherein the data payload size can be programmed based on the error information corresponding to it [paragraph 0023, lines 13-15]. Therefore, different levels of ECC would have different data payload sizes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used.

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to explicitly teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the method of Diggs wherein the gear

includes a lower-latency ECC decode option and one or more higher-latency ECC decode options.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

While Diggs teaches wherein the gears used for encoding would have different levels of error correcting ability [Diggs, paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)], Diggs, Manning, and Cherubini fail to explicitly teach wherein the latency being referred to in the claims is from the decoding of primary parity symbols alone versus the decoding with secondary parity and/or tertiary parity symbols, as opposed to the latency as it is referred to in the Cherubini reference. Further, they fail explicitly to teach that the ECC parameters for both the lower latency decode option and the higher latency decode option are different for the first gear and the second gear.

Anholt in an analogous art teaches wherein for an error correction scheme there can be encoding of data where there is a first ECC level using primary parity and a second ECC level using secondary parity symbols that is higher latency and of stronger error correcting ability [Anholt, column 2, lines 44-48 and 64-68; column 9, lines 1-7 and 44-47]. Further, Anholt teaches wherein for the first and second levels of ECC, different

ECC parameters in order to retain the varying levels of correction ability for each gear as taught by Diggs [Anholt, column 9, lines 36-40, any suitable ECC, including LDPC could be used for the ECC2; column 6, lines 48-56, the ECC may be any of multiple options].

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the method of Diggs, Manning, and Cherubini and apply the teachings of Anholt wherein the lower latency decode option consists of primary parity symbols alone and the one or more higher latency decode options have secondary parity symbols and/or tertiary parity symbols. Further, if the gears where to consist of multiple decode options as taught by Anholt then it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the method of Diggs, Manning, and Cherubini and apply the teachings of Anholt wherein the different gears could have different ECC parameters in order to retain the varying levels of correction ability for each gear as taught by Diggs.

Motivation to do so would have been to be able to have multiple levels of ECC to allow for recovery from failure to reconstruct a given data item with a first level of ECC using a second level of ECC [Anholt, column 2, lines 44-48]. Motivation to have wherein the different gears levels of ECC have different parameters would have been to allow for various decoding options with different error correcting capabilities [Diggs, paragraphs 0024 and 0025].

Regarding claim 2, Diggs discloses the method of claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to

one or more predetermined thresholds such that the bit error rate is actively monitored [paragraph 0031, lines 1-4 (the controller can be set up to repeatedly monitor the data and set the ECC level based on the reading), paragraph 0051 and paragraph 0056 (the data monitored can be the bit error rate monitored and can be compared to threshold values)].

Regarding claim 3, Diggs discloses the method of claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] at least once every 4 program/erase cycles [paragraph 0045, lines 5-11 (the block erase count can be monitored) and paragraph 0031, lines 1-4 (events can be used to monitor the data and set the ECC level periodically)].

Regarding claim 4, Diggs teaches the method of claim 1, wherein determining, comparing, and selecting are performed in connection with at least one of flash initialization, read, or write operations for an end user [figure 2, item 203, during write operations, the ECC setting is determined before encoding the data with the ECC level that is determined].

Regarding claim 5, Diggs discloses the method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region [paragraph

0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

Regarding claim 8, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations (paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs wherein selecting further comprises selecting a gear having a smaller payload size and a higher amount of correction capability based at least partly on the number of decoding iterations used to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

Regarding claim 9, Diggs discloses the method of claim 1, further comprising: receiving a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; performing a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

Regarding claim 14, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to explicitly teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the method of Diggs wherein n the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

Regarding claim 15, Diggs teaches an apparatus comprising: a first circuit [monitoring circuit] configured to determine a bit error rate [paragraph 0029, lines 10-13 (the bit error rate of data is monitored) and paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] associated with a region associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)] and to compare the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes [paragraph 0025 (in one embodiment two different

ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and a second circuit [controller] configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used) and paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)].

While Diggs does encompass the entirety of the claim, in teaching that that data blocks of different size and format can be used it is not explicitly taught that integer fractions of pages can be used. Furthermore, in teaching this it can be construed that different data payload for different ECC levels could be used, but this is also not explicitly taught by Diggs.

Manning teaches wherein portions of pages can be used for error correction encoding [figure 2 and paragraph 0032 (portions of pages are error coded)]. Manning also teaches wherein the data payload size can be programmed based on the error

information corresponding to it [paragraph 0023, lines 13-15]. Therefore, different levels of ECC would have different data payload sizes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the apparatus of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used.

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

However, Diggs and Manning fail to explicitly teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

While Diggs teaches wherein the gears used for encoding would have different levels of error correcting ability [Diggs, paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)], Diggs, Manning, and Cherubini fail to explicitly teach wherein the latency being referred to in the claims is from the decoding of primary parity symbols alone versus the decoding with secondary parity and/or tertiary parity symbols, as opposed to the latency as it is referred to in the Cherubini reference. Further, they fail explicitly to teach that the ECC parameters for both the lower latency decode option and the higher latency decode option are different for the first gear and the second gear.

Anholt in an analogous art teaches wherein for an error correction scheme there can be encoding of data where there is a first ECC level using primary parity and a second ECC level using secondary parity symbols that is higher latency and of stronger error correcting ability [Anholt, column 2, lines 44-48 and 64-68; column 9, lines 1-7 and 44-47]. Further, Anholt teaches wherein for the first and second levels of ECC, different ECC parameters in order to retain the varying levels of correction ability for each gear as taught by Diggs [Anholt, column 9, lines 36-40, any suitable ECC, including LDPC

could be used for the ECC2; column 6, lines 48-56, the ECC may be any of multiple options].

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the apparatus of Diggs, Manning, and Cherubini and apply the teachings of Anholt wherein the lower latency decode option consists of primary parity symbols alone and the one or more higher latency decode options have secondary parity symbols and/or tertiary parity symbols. Further, if the gears where to consist of multiple decode options as taught by Anholt then it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the apparatus of Diggs, Manning, and Cherubini and apply the teachings of Anholt wherein the different gears could have different ECC parameters in order to retain the varying levels of correction ability for each gear as taught by Diggs.

Motivation to do so would have been to be able to have multiple levels of ECC to allow for recovery from failure to reconstruct a given data item with a first level of ECC using a second level of ECC [Anholt, column 2, lines 44-48]. Motivation to have wherein the different gears levels of ECC have different parameters would have been to allow for various decoding options with different error correcting capabilities [Diggs, paragraphs 0024 and 0025].

Regarding claim 16, Diggs teaches the apparatus of claim 1, wherein the ECC encoder is configured to determine the bit error rate and the processor is configured to compare and to select in connection with at least one of flash initialization, read, or write

operations for an end user [figure 2, item 203, during write operations, the ECC setting is determined before encoding the data with the ECC level that is determined].

Regarding claim 17, Diggs discloses the apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

Regarding claim 19, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations (paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the apparatus of Diggs wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

Regarding claim 20, Diggs discloses the apparatus of Claim 15, wherein the processor is further configured to: receive a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; perform a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

Regarding claim 22, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

Regarding claim 23, Diggs discloses the apparatus of claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive [paragraphs 0061 and 0062 (show how the apparatus could be embodied in this way) and claim 14].

Regarding claim 24, Diggs teaches an apparatus [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for selecting an error correction coding (ECC) scheme, the apparatus comprising: a means for determining a bit error rate [paragraph 0029, lines 10-13 (the bit error rate of data is monitored)] associated with a region associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11

(blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]; a means for comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used)].

While Diggs does encompass the entirety of the claim, in teaching that that data blocks of different size and format can be used it is not explicitly taught that integer fractions of pages can be used. Furthermore, in teaching this it can be construed that

different data payload for different ECC levels could be used, but this is also not explicitly taught by Diggs.

Manning teaches wherein portions of pages can be used for error correction encoding [figure 2 and paragraph 0032 (portions of pages are error coded)]. Manning also teaches wherein the data payload size can be programmed based on the error information corresponding to it [paragraph 0023, lines 13-15]. Therefore, different levels of ECC would have different data payload sizes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the apparatus of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used.

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

However, Diggs and Manning fail to explicitly teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the method of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

While Diggs teaches wherein the gears used for encoding would have different levels of error correcting ability [Diggs, paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)], Diggs, Manning, and Cherubini fail to explicitly teach wherein the latency being referred to in the claims is from the decoding of primary parity symbols alone versus the decoding with secondary parity and/or tertiary parity symbols, as opposed to the latency as it is referred to in the Cherubini reference. Further, they fail explicitly to teach that the ECC parameters for both the lower latency decode option and the higher latency decode option are different for the first gear and the second gear.

Anholt in an analogous art teaches wherein for an error correction scheme there can be encoding of data where there is a first ECC level using primary parity and a second ECC level using secondary parity symbols that is higher latency and of stronger

error correcting ability [Anholt, column 2, lines 44-48 and 64-68; column 9, lines 1-7 and 44-47]. Further, Anholt teaches wherein for the first and second levels of ECC, different ECC parameters in order to retain the varying levels of correction ability for each gear as taught by Diggs [Anholt, column 9, lines 36-40, any suitable ECC, including LDPC could be used for the ECC2; column 6, lines 48-56, the ECC may be any of multiple options].

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the apparatus of Diggs, Manning, and Cherubini and apply the teachings of Anholt wherein the lower latency decode option consists of primary parity symbols alone and the one or more higher latency decode options have secondary parity symbols and/or tertiary parity symbols. Further, if the gears where to consist of multiple decode options as taught by Anholt then it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the apparatus of Diggs, Manning, and Cherubini and apply the teachings of Anholt wherein the different gears could have different ECC parameters in order to retain the varying levels of correction ability for each gear as taught by Diggs.

Motivation to do so would have been to be able to have multiple levels of ECC to allow for recovery from failure to reconstruct a given data item with a first level of ECC using a second level of ECC [Anholt, column 2, lines 44-48]. Motivation to have wherein the different gears levels of ECC have different parameters would have been to allow for various decoding options with different error correcting capabilities [Diggs, paragraphs 0024 and 0025].

Regarding claim 25, Diggs discloses the apparatus [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

Regarding claim 26, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations [paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the apparatus of Diggs wherein the selecting means further comprises means [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

Regarding claim 27, Diggs discloses the apparatus of Claim 24, further comprising: means [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for receiving a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; means for performing a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to

determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

Regarding claim 29, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

Regarding claim 30, Diggs discloses the apparatus of claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive [paragraphs 0061 and 0062 (show how the apparatus could be embodied in this way) and claim 14].

11. Claims 6, 7, 10-13, 18, 21, and 28 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651) in view of Manning (U.S. Patent Application Publication 2013/0246891), in view of Cherubini (U.S. Patent Application Publication 2013/0179754), in view of Anholt (U.S. Patent 8,495,465) as applied to claims 1, 15, and 24 above, further in view of Bennett (U.S. Patent Application Publication 2013/0080862).

Regarding claim 6, Diggs teaches wherein a level of ECC is encoded into a given set of data dependent on comparing information monitored from said data with certain threshold values, where the threshold can be the bit error rate. [paragraph 0008, lines11-15 and paragraph 0029, lines 12-13]. Diggs, however, does not explicitly teach wherein the highest data payload size is used that still has sufficient correction ability for the determined bit error rate.

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the higher the data payload size, the better the memory density (ratio of data to ECC bits) and the worse the error correcting ability (because of the higher data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs so that selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate.

Motivation to do so would have been to achieve the best memory density possible [paragraph 0023, lines 19-23].

Neither Diggs nor Manning teach wherein there is a predetermined acceptable uncorrectable bit error rate.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Bennets to the modified method of Diggs with Manning wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

Regarding claim 7, Diggs and Manning fail to teach where the aUBER is 1E-16.

Bennett teaches wherein the aUBER can be 1E-18 and furthermore can be set to required level [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Bennett to the method of Diggs and Manning such that the aUBER is 1E-16.

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

Regarding claim 10, Diggs further teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs and Manning fail to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the method of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such

that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Regarding claim 11, Diggs further teaches wherein the threshold values used can be modified by different functions and set to any value necessary to achieve the desired level of error correction [paragraphs 0050 and 0056]. Thereby, it could be had the method of claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor SFx is in a range from 1.5 to 4.

Regarding claim 12, Diggs further teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056]. Furthermore Diggs teaches that a high monitored bit error rate anticipates a high probability of future errors occurring and because of this the ECC setting is adjusted [paragraph 0051]. This information corresponds to the excursion factor times the retention factor, which is the

safety factor, because they both anticipate the increased probability of future errors occuring.

Diggs and Manning fail to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate that is set to a required value [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs and Manning with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction and have this value be divided by a safety factor, where the safety factor is found by the given expression as a product of the excursion factor and the retention factor. Diggs teaches where these threshold values can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system and the safety factor is itself found as a product of the excursion factor and the retention factor. Thereby, the method of Diggs would be modified by the teachings of Bennett wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor, where the safety factor is equal to the excursion factor times the retention factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Regarding claim 13, Diggs teaches wherein the threshold values used can be modified by different functions and set to any value necessary to achieve the desired level of error correction [paragraphs 0050 and 0056]. Thereby, it can be had the method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

Regarding claim 18, Diggs teaches wherein a level of ECC is encoded into a given set of data dependent on comparing information monitored from said data with certain threshold values, where the threshold can be the bit error rate. [paragraph 0008, lines11-15 and paragraph 0029, lines 12-13]. Diggs, however, does not explicitly teach wherein the highest data payload size is used that still has sufficient correction ability for the determined bit error rate.

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the higher the data payload size, the better the memory density (ratio of data to ECC bits) and the worse the error correcting ability (because of the higher data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs so that selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate.

Motivation to do so would have been to achieve the best memory density possible [paragraph 0023, lines 19-23].

Diggs and Manning fail to teach wherein there is a predetermined acceptable uncorrectable bit error rate.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Manning and Bennets to the apparatus of Diggs wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

Regarding claim 21, Diggs teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs fails to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the apparatus of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Regarding claim 28, Diggs teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs fails to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the apparatus of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such

that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTIAN DORMAN whose telephone number is

(571)272-7657. The examiner can normally be reached on Monday-Friday 9:30am-3pm, Friday alternating.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert Decady can be reached on 571-272-3819. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHRISTIAN DORMAN/
Examiner, Art Unit 2112

/ALBERT DECADY/

Supervisory Patent Examiner, Art Unit 2112

Applicant-Initiated Interview Summary	**Application No.** 13/477,600	**Applicant(s)** NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	**Art Unit** 2112	

All participants (applicant, applicant's representative, PTO personnel):

(1) *CHRISTIAN DORMAN*. (3) *Mike Okamoto*.

(2) *Albert Decady*. (4)_____.

Date of Interview: *28 April 2014*.

Type: ☒ Telephonic ☐ Video Conference
☐ Personal [copy given to: ☐ applicant ☐ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☒ No.
If Yes, brief description: _____.

Issues Discussed ☐101 ☒112 ☐102 ☒103 ☐Others
(For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: *1, 4, and 6*.

Identification of prior art discussed: *Diggs [US 2009/0070651], Manning [US 2013/0246891], Cherubini [US 2013/0179754]*.

Substance of Interview
(For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Proposed amendment was made for claim 1 wherein dependent claim 14 was drawn into it with additional changes in terminology. Discussed use of lower latency codes in relation to claim 14 in how want to use lower latency codes for at least 90% of decoding (as is claimed). Discussed how Diggs teaches selecting the level of ECC to meet certain thresholds and meeting needed error rates and how Cherubini teaches wherein latency of the ECC selected is a factor in decoding.
For 112 (b) for claim 4, applicant proposed changing language from "concurrently" to "in connection" wherein this change in language would overcome the 112 (b).
For the 112(b) for claim 6, examiner withdraws rejection after discussion (lacked merit).

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

☐ Attachment

/CHRISTIAN DORMAN/ Examiner, Art Unit 2112	

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

———

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Notice of References Cited	Application/Control No. 13/477,600	Applicant(s)/Patent Under Reexamination NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 2112	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2009/0070651	03-2009	Diggs et al.	714/752
*	B	US-2013/0246891	09-2013	Manning et al.	714/773
*	C	US-2013/0179754	07-2013	Cherubini et al.	714/773
*	D	US-8,495,465	07-2013	Anholt et al.	714/763
*	E	US-2013/0080862	03-2013	BENNETT, Jon C.R.	714/782
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	2112

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/19/2013	06/16/2014							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	✓							
	5	✓	✓							
	6	✓	✓							
	7	✓	✓							
	8	✓	✓							
	9	✓	✓							
	10	✓	✓							
	11	✓	✓							
	12	✓	✓							
	13	✓	✓							
	14	✓	✓							
	15	✓	✓							
	16	✓	✓							
	17	✓	✓							
	18	✓	✓							
	19	✓	✓							
	20	✓	✓							
	21	✓	✓							
	22	✓	✓							
	23	✓	✓							
	24	✓	✓							
	25	✓	✓							
	26	✓	✓							
	27	✓	✓							
	28	✓	✓							
	29	✓	✓							
	30	✓	✓							

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	886	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:08
S2	130050	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:09
S3	50326	(parameter same (chang$4 or determin$3 or depend$4 or adjust$4)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:13
S4	50338	(parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/16 19:14
S5	7928	(parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) and (error adj correct$4) and ((bit error rate) or BER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:15
S6	38	(parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) and (error adj correct$4) and ((bit error rate) or BER) and ((uncorrectable bit error rate) or UBER or aUBER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:16
S7	95	((parameter same (chang$4 or determin$4 or depend$4 or adjust$4)) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:18
S8	1018	((chang$4 or determin$4 or depend$4 or	US-PGPUB;	ADJ	ON	2013/12/16

		adjust$4) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER)	USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			19:19
S9	1	((chang$4 or determin$4 or depend$4 or adjust$4) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER) and ((uncorrectable bit error rate) or UBER or aUBER)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:19
S10	877	((chang$4 or determin$4 or depend$4 or adjust$4) same2 data payload) and (error adj correct$4) and ((bit error rate) or BER) and memory	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:21
S11	71	S10 and S2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:22
S12	4	S10 and S1	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/16 19:22
S13	0	("2013/0318420").URPN.	USPAT	OR	ON	2013/12/16 19:33
S14	9	((integer adj fraction) with page) same memory	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 08:51
S15	0	((integer adj fraction) with page) same memory same (page stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 09:05
S16	36	((fraction or portion or part) with page) same memory same (page stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 09:05
S17	887	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS;	OR	ON	2013/12/18 09:05

			EPO; JPO; DERWENT; IBM_TDB			
S18	130116	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:05
S19	1	S16 and S17	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2013/12/18 09:06
S20	2	"20120144270".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:22
S21	2	"20090070651".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:30
S22	0	("2012/0144270").URPN.	USPAT	OR	ON	2013/12/18 09:31
S23	230	("20020063774" | "20020085419" | "20020154769" | "20030065876" | "20030101404" | "20030105620" | "20030192007" | "20040015771" | "20040030971" | "20040153722" | "20040153817" | "20040181735" | "20050013165" | "20050018482" | "20050083735" | "20050117401" | "20050120265" | "20050128811" | "20050138533" | "20050144213" | "20050144368" | "20050169057" | "20050172179" | "20050213393" | "20060059406" | "20060059409" | "20060064537" | "20060101193" | "20060203587" | "20060221692" | "20060248434" | "20060268608" | "20060294312" | "20070002633" | "20070025157" | "20070063180" | "20070103992" | "20070104004" | "20070109858" | "20070121386" | "20070124652" | "20070143561" | "20070150694" | "20070168625" | "20070171714" | "20070171730" | "20070180346" | "20070223277" | "20070226582" | "20070226592" | "20070228449" | "20070253249" | "20070253250" | "20070263439" | "20070266291" | "20070271494" | "20080010581" | "20080028014" |	US-PGPUB; USPAT; USOCR	OR	ON	2013/12/18 09:31

		"20080055989" | "20080082897" | "20080092026" | "20080104309" | "20080116509" | "20080126686" | "20080127104" | "20080128790" | "20080130341" | "20080137413" | "20080148115" | "20080158958" | "20080159059" | "20080162079" | "20080168216" | "20080168320" | "20080181001" | "20080198650" | "20080198652" | "20080219050" | "20080225599" | "20080263262" | "20080282106" | "20080285351" | "20080301532" | "20090024905" | "20090043951" | "20090072303" | "20090091979" | "20090103358" | "20090106485" | "20090113275" | "20090125671" | "20090144600" | "20090150748").PN. OR ("20090157964" | "20090158126" | "20090168524" | "20090187803" | "20090199074" | "20090213653" | "20090213654" | "20090228761" | "20090240872" | "20100005270" | "20100058146" | "20100064096" | "20100088557" | "20100091535" | "20100095186" | "20100110787" | "20100115376" | "20100122113" | "20100124088" | "20100131580" | "20100131806" | "20100131809" | "20100131826" | "20100131827" | "20100131831" | "20100146191" | "20100146192" | "20100149881" | "20100180073" | "20100199149" | "20100211724" | "20100211833" | "20100211856" | "20100251066" | "20100253555" | "20100257309" | "20100293321" | "20110051521" | "20110055461" | "20110096612" | "20110119562" | "20110153919" | "20110161775" | "20110214029" | "20110214039" | "20110246792" | "20110246852" | "20110252187" | "20110252188" | "20110271043" | "20110302428" | "20120001778" | "20120005554" | "20120005558" | "20120005560" | "20120008401" | "20120008414" | "20120051144" | "20120063227" | "20120066441" | "20120110250" | "4463375" | "4584686" | "4589084" | "4866716" | "5077737" | "5297153" | "5657332" | "5729490" | "5793774" | "5926409" | "5956268" | "5982659" | "6038634" | "6094465" | "6119245" | "6182261" | "6192497" | "6195287" | "6199188" | "6209114" | "6259627" | "6278633" | "6279133" | "6301151" | "6370061" | "6374383" | "6504891" | "6532169" | "6532556" | "6553533" | "6560747" | "6637002" | "6639865" | "6674665" | "6704902" | "6751766" | "6772274" | "6781910" | "6792569" | "6873543" | "6891768" | "6914809" | "6915477").PN. OR ("6952365" | "6961890" | "6990012" | "6996004" |				

		"6999854" | "7010739" | "7012835" | "7038950" | "7068539" | "7079436" | "7149950" | "7177977" | "7191379" | "7196946" | "7203874" | "7290203" | "7292365" | "7301928" | "7441067" | "7466575" | "7533328" | "7558109" | "7593263" | "7697326" | "7706182" | "7804718" | "7805663" | "7805664" | "7844877" | "7961797" | "8020073" | "8122328").PN. OR ("8305812").URPN.				
S24	230	("20020063774" | "20020085419" | "20020154769" | "20030065876" | "20030101404" | "20030105620" | "20030192007" | "20040015771" | "20040030971" | "20040153722" | "20040153817" | "20040181735" | "20050013165" | "20050018482" | "20050083735" | "20050117401" | "20050120265" | "20050128811" | "20050138533" | "20050144213" | "20050144368" | "20050169057" | "20050172179" | "20050213393" | "20060059406" | "20060059409" | "20060064537" | "20060101193" | "20060203587" | "20060221692" | "20060248434" | "20060268608" | "20060294312" | "20070002633" | "20070025157" | "20070063180" | "20070103992" | "20070104004" | "20070109858" | "20070121386" | "20070124652" | "20070143561" | "20070150694" | "20070168625" | "20070171714" | "20070171730" | "20070180346" | "20070223277" | "20070226582" | "20070226592" | "20070228449" | "20070253249" | "20070253250" | "20070263439" | "20070266291" | "20070271494" | "20080010581" | "20080028014" | "20080055989" | "20080082897" | "20080092026" | "20080104309" | "20080116509" | "20080126686" | "20080127104" | "20080128790" | "20080130341" | "20080137413" | "20080148115" | "20080158958" | "20080159059" | "20080162079" | "20080168216" | "20080168320" | "20080181001" | "20080198650" | "20080198652" | "20080219050" | "20080225599" | "20080263262" | "20080282106" | "20080285351" | "20080301532" | "20090024905" | "20090043951" | "20090072303" | "20090091979" | "20090103358" | "20090106485" | "20090113275" | "20090125671" | "20090144600" | "20090150748").PN. OR ("20090157964" | "20090158126" | "20090168524" | "20090187803" | "20090199074" | "20090213653" | "20090213654" | "20090228761" | "20090240872" | "20100005270" | "20100058146" | "20100064096" | "20100088557" | "20100091535" | "20100095186" |	US-PGPUB; USPAT; USOCR	OR	ON	2013/12/18 09:32

		"20100110787" | "20100115376" | "20100122113" | "20100124088" | "20100131580" | "20100131806" | "20100131809" | "20100131826" | "20100131827" | "20100131831" | "20100146191" | "20100146192" | "20100149881" | "20100180073" | "20100199149" | "20100211724" | "20100211833" | "20100211856" | "20100251066" | "20100253555" | "20100257309" | "20100293321" | "20110051521" | "20110055461" | "20110096612" | "20110119562" | "20110153919" | "20110161775" | "20110214029" | "20110214039" | "20110246792" | "20110246852" | "20110252187" | "20110252188" | "20110271043" | "20110302428" | "20120001778" | "20120005554" | "20120005558" | "20120005560" | "20120008401" | "20120008414" | "20120051144" | "20120063227" | "20120066441" | "20120110250" | "4463375" | "4584686" | "4589084" | "4866716" | "5077737" | "5297153" | "5657332" | "5729490" | "5793774" | "5926409" | "5956268" | "5982659" | "6038634" | "6094465" | "6119245" | "6182261" | "6192497" | "6195287" | "6199188" | "6209114" | "6259627" | "6278633" | "6279133" | "6301151" | "6370061" | "6374383" | "6504891" | "6532169" | "6532556" | "6553533" | "6560747" | "6637002" | "6639865" | "6674665" | "6704902" | "6751766" | "6772274" | "6781910" | "6792569" | "6873543" | "6891768" | "6914809" | "6915477").PN. OR ("6952365" | "6961890" | "6990012" | "6996004" | "6999854" | "7010739" | "7012835" | "7038950" | "7068539" | "7079436" | "7149950" | "7177977" | "7191379" | "7196946" | "7203874" | "7290203" | "7292365" | "7301928" | "7441067" | "7466575" | "7533328" | "7558109" | "7593263" | "7697326" | "7706182" | "7804718" | "7805663" | "7805664" | "7844877" | "7961797" | "8020073" | "8122328").PN. OR ("8305812").URPN.				
S25	1	"20110051521".pn.	US-PGPUB; USPAT; USOCR	OR	ON	2013/12/18 09:32
S26	9	(US-20120144270-$ or US-20130080862-$ or US-20130246891-$ or US-20090070651-$ or US-20110051521-$ or US-20100131806-$ or US-20100131809-$ or US-20100088557-$).did. or (US-8305812-$).did.	US-PGPUB; USPAT	OR	ON	2013/12/18 09:57
S27	5	S26 and (data adj payload)	US-PGPUB; USPAT; USOCR; FPRS;	OR	ON	2013/12/18 09:57

			EPO; JPO; DERWENT; IBM_TDB			
S28	5	S26 and payload	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 09:58
S29	1	S26 and payload and (page adj stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:02
S30	1	S26 and (page adj stripe)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:02
S31	2473	(parameter same (chang$4 or determin$3 or depend$4 or adjust$4 or adapt$5)) and (error adj correct$4) and (data adj payload)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:05
S32	8	S26 and (BER or (bit adj error adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:13
S33	4	S26 and (UBER or aUBER or (uncorrectable adj bit adj error adj rate))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:13
S34	757	((data adj payload) with (chang$4 or determin$3 or depend$4 or adjust$4 or adapt$5)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:36
S35	1129	((data adj payload) with (chang$4 or determin$3 or depend$4 or adjust$4 or adapt$5 or different)) and (error adj correct$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:37
S36	5	S35 and S17	US-PGPUB; USPAT;	OR	ON	2013/12/18 10:37

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S37	3	S34 and S17	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:39
S38	0	S34 and "L45"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:39
S39	90	S34 and S18	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 10:39
S40	9	(US-20120144270-$ or US-20130080862-$ or US-20130246891-$ or US-20090070651-$ or US-20110051521-$ or US-20100131806-$ or US-20100131809-$ or US-20100088557-$).did. or (US-8305812-$).did.	US-PGPUB; USPAT	OR	ON	2013/12/18 18:45
S41	887	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S42	130116	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S43	0	S40 and (auber)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46
S44	0	S40 and (uncorrectable adj bit)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:46

S45	4	S40 and uber	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:47
S46	4	S40 and (auber or uber or (uncorrect$4 adj bit))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:48
S47	9483	(auber or uber or (uncorrect$4 adj bit adj error))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:53
S48	2	((allow$4 or accept$4) adj uncorrect$4 adj bit adj error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:53
S49	9225	(auber or uber or ((allow$4 or accept$4) adj uncorrect$4 adj bit adj error))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/18 18:54
S50	130147	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:36
S51	891	714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:36
S52	0	(uncorrect$4 adj error) same (1E-16)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2013/12/19 09:36
S53	2	(uncorrect$4 adj error) same (10^ -16 or 1E-16 or -16)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2013/12/19 09:37

			DERWENT; IBM_TDB			
S54	9	(US-20120144270-$ or US-20130080862-$ or US-20130246891-$ or US-20090070651-$ or US-20110051521-$ or US-20100131806-$ or US-20100131809-$ or US-20100088557-$).did. or (US-8305812-$).did.	US-PGPUB; USPAT	OR	ON	2013/12/19 09:39
S55	3	S54 and (uncorrect$4 adj error)	US-PGPUB; USPAT	OR	ON	2013/12/19 09:39
S56	2	S54 and (wear adj level)	US-PGPUB; USPAT	OR	ON	2013/12/19 10:00
S57	2	S54 and raw	US-PGPUB; USPAT	OR	ON	2013/12/19 10:11
S58	4	S54 and format	US-PGPUB; USPAT	OR	ON	2013/12/19 10:11
S59	20	((raw adj bit adj error adj rate) or rber) and (format$4 with command)	US-PGPUB; USPAT	OR	ON	2013/12/19 10:35
S60	3	S54 and uncorrect$5	US-PGPUB; USPAT	OR	ON	2013/12/19 10:44
S61	1	S54 and latency	US-PGPUB; USPAT	OR	ON	2013/12/19 10:50
S62	1	S54 and laten$4	US-PGPUB; USPAT	OR	ON	2013/12/19 10:50
S63	392	(latency with (option or parameter)) same (ECC or (error adj correction) or decode)	US-PGPUB; USPAT	OR	ON	2013/12/19 10:53
S64	1	S63 and S51	US-PGPUB; USPAT	OR	ON	2013/12/19 10:53
S65	2	"20130179754".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/04/28 12:48
S66	2	"20130179754".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 14:30
S67	68	("20070233752" | "20070277066" | "20080172589" | "20090164836" | "20090182939" | "20100008214" | "20100332894" | "20100332922" | "20110040926" | "20110167319" | "20110213920" | "20120072680" | "20130072680" | "20130073895" | "20130151914" | "20130173955" | "5592497" | "5974544" | "6574774" | "6651212" | "6732325" | "6751766" | "7739576" | "7783955" | "7944748" | "8495465" | "8560881" | "8621318").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 14:32
S68	1077	714/773.ccls.	US-PGPUB;	OR	ON	2014/06/02

			USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			14:45
S69	136480	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 14:45
S70	42	primary with (secondary tertiary) with (parity ECC) with decod$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 14:45
S71	14	latency with decod$3 near4 option	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 15:29
S72	108	primary with (secondary tertiary) with option$4 same decod$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 16:25
S73	0	primary with (secondary tertiary) with option$4 same decod$3 same latency	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 16:25
S74	0	primary with (secondary tertiary) with (parity ECC) with option$4 same decod$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 16:26
S75	3	primary with (secondary tertiary) with option$4 same decod$3 same (parity ECC (error adj correct$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/02 16:26
S76	68	("20070233752" | "20070277066" | "20080172589" | "20090164836" | "20090182939" | "20100008214" | "20100332894" | "20100332922" | "20110040926" | "20110167319" | "20110213920" | "20120072680" |	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2014/06/03 12:21

		"20130072680" | "20130073895" | "20130151914" | "20130173955" | "5592497" | "5974544" | "6574774" | "6651212" | "6732325" | "6751766" | "7739576" | "7783955" | "7944748" | "8495465" | "8560881" | "8621318").PN.	IBM_TDB			
S77	7	S76 and latency	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:21
S78	438	decod$3 adj option	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:43
S79	2189	decod$3 near4 option	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:43
S80	1129	S79 and L!	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:43
S81	1	S79 and S76	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:44
S82	29	S79 same latency	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:44
S83	38	S79 same (ECC parity)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/03 12:53
S84	38	S83 not S82	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2014/06/03 12:59

			DERWENT; IBM_TDB			
S85	2	"20130179754".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 16:55
S86	113174	first with second with decod$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 16:58
S87	82	first with second with decod$3 near3 option	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 16:59
S88	0	first with second with decod$3 near3 option same (ECC parity)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 16:59
S89	0	first with second with decod$3 near3 option same (ECC parity (error adj correct$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 16:59
S90	6397	first with second with decod$3 same (ECC parity (error adj correct$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:00
S91	0	first near3 decod$3 near3 option same (ECC parity (error adj correct$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:00
S92	0	first with decod$3 near3 option same (ECC parity (error adj correct$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:01
S93	5140	(first with (ECC parity (error adj correct$4)) with decod$3) same (second with (ECC parity (error adj correct$4))	US-PGPUB; USPAT; USOCR;	OR	ON	2014/06/05 17:01

		with decod$3)	FPRS; EPO; JPO; DERWENT; IBM_TDB			
S94	5236	((first primary) with (ECC parity (error adj correct$4)) with decod$3) same ((second$4 tertiary) with (ECC parity (error adj correct$4)) with decod$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:02
S95	2223	((first primary) near4 (ECC parity (error adj correct$4)) with decod$3) same ((second$4 tertiary) near4 (ECC parity (error adj correct$4)) with decod$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:02
S96	0	((first primary) near4 (ECC parity (error adj correct$4)) with decod$3) same ((second$4 tertiary) near4 (ECC parity (error adj correct$4)) with decod$3) same (option choice)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:02
S97	13	((first primary) near4 (ECC parity (error adj correct$4)) with decod$3) same ((second$4 tertiary) near4 (ECC parity (error adj correct$4)) with decod$3) same latency	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2014/06/05 17:03

6/16/2014 12:42:56 PM
C:\Users\cdorman\Documents\EAST\Workspaces\13477600.wsp

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	4181
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 2	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	5,592,497	01-07-1997	Lokhoff	
	2	5,974,544	10-26-1999	Jeffries	
	3	6,574,774	06-03-2003	Vasiliev, Peter I.	
	4	6,651,212	11-18-2003	Katayama, et al.	
	5	6,732,325	05-04-2004	Tash et al.	
	6	6,751,766	06-15-2004	Guterman, et al.	
	7	7,739,576	06-15-2010	Radke, William H.	
	8	7,783,955	08-24-2010	Mark Murin	
	9	7,944,748	05-17-2011	Eggleston, et al.	
	10	8,495,465	07-23-2013	Anholt, et al.	
	11	8,560,881	10-15-2013	Frost, et al.	
	12	8,612,680	12-17-2013	Madnani, et al.	
	13	8,621,318	12-31-2013	Micheloni, et al.	
	14	2007/0233752	10-04-2007	Bangalore, et al.	
	15	2007/0277066	11-29-2007	Gajapathy, et al.	
	16	2008/0172589	07-17-2008	Gallezot, et al.	
	17	2009/0182939	07-16-2009	Hluchyj, et al.	
	18	2009/0164836	06-25-2009	Richard Carmichael	
	19	2010/0008214	01-14-2010	Siaud, et al.	
	20	2010/0332894	12-30-2010	Bowers, et al.	
	21	2010/0332922	12-30-2010	Chang, et al.	
	22	2011/0040926	02-17-2011	Frost, et al.	
	23	2011/0167319	07-07-2011	Jeddeloh, Joe M.	
	24	2011/0213920	09-01-2011	Frost, et al.	
	25	2012/0072680	03-22-2012	Kimura, et al.	
	26	2013/0073895	03-21-2013	Earl T. Cohen	
	27	2013/0151914	06-13-2013	Cadigan	
	28	2013/0173955	07-04-2013	Hallak, et al.	

Examiner Signature	Date Considered
***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	4181
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 2 OF 2	Attorney Docket No.	DATUMTE.076A1

FOREIGN PATENT DOCUMENTS						
Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	29	USPTO; Office Action dated December 31, 2013, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	
	30	USPTO; Office Action dated January 15, 2014, from related application U.S. Patent Application No. 13/477,568 (filed May 22, 2012).	
	31	USPTO; Office Action dated January 21, 2014, from related application U.S. Patent Application No. 13/477,633 (filed May 22, 2012).	
	32	USPTO; Office Action dated February 28, 2014, from related application U.S. Patent Application No. 13/477,599 (filed May 22, 2012).	
	33	USPTO; Office Action dated May 7, 2014, from related application U.S. Patent Application No. 13/477,601 (filed May 22, 2012).	

16999441
050514

Examiner Signature	/Christian Dorman/	Date Considered	06/16/2014

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	2112

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
714	773	12/16/2013, 12/18/2013, 12/19/2013, 6/02/2014	CD

SEARCH NOTES		
Search Notes	Date	Examiner
EAST US_PGPUB, USPAT, USOCR, FPRS, EPO, JPO, DERWENT, IBM_TDB - search results are attached	12/16/2013, 12/18/2013, 12/19/2013, 6/02/2014, 6/03/2014, 6/05/2014	CD
Inventor Name Search, PALM	12/16/2013, 12/19/2013	CD
Double Patenting Search, PALM	12/16/2013, 12/19/2013	CD
IEEE NPL Search	12/18/2013	CD

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

Doc code: RCEX
Doc description: Request for Continued Examination (RCE)

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
(Submitted Only via EFS-Web)

Application Number	13477600	Filing Date	2012-05-22	Docket Number (if applicable)	DATUMTE.076A1	Art Unit	4181
First Named Inventor	Philip L. Northcott			Examiner Name	Dorman, Christian M.		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

☐ Other _____

☒ Enclosed

☒ Amendment/Reply

☒ Information Disclosure Statement (IDS)

☐ Affidavit(s)/ Declaration(s)

☐ Other _____

MISCELLANEOUS

☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months _____
(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

☐ Other _____

FEES

The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.

☒ The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 111410

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature

☐ Applicant Signature

Doc code: RCEX
Doc description: Request for Continued Examination (RCE)

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Signature of Registered U.S. Patent Practitioner			
Signature	Michael Okamoto	Date (YYYY-MM-DD)	2014-11-17
Name	Michael S. Okamoto	Registration Number	47831

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor	:	Philip L. Northcott
App. No.	:	13/477,600
Filed	:	May 22, 2012
Examiner	:	Dorman, Christian M.
Art Unit	:	2112
Conf. No.	:	2672

Submission under 37 C.F.R. § 1.114

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant requests continued examination and accordingly requests the Examiner to withdraw the finality of the previous Office Action in accordance with 37 C.F.R. § 1.114. Applicant respectfully responds to the Final Office Action of July 16, 2014 as set forth below.

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Summary of Interview begins on page 9 of this paper.

Remarks/Arguments begin on page 10 of this paper.

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method of selecting an error correction coding (ECC) scheme, the method comprising:

determining a bit error rate associated with a region ~~associated with a plurality of~~ comprising a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region;

comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, ~~wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity symbols alone and one or more higher-latency ECC decode options with secondary parity symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower-latency ECC decode option and the higher-latency ECC decode option are different for the first gear and the second gear,~~ wherein the first gear has a different data payload size and correction capability than the second gear, wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear; and

selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds;

wherein determining, comparing, and selecting are performed by an integrated circuit.

2. (Original) The method of Claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds such that the bit error rate is actively monitored.

3. (Original) The method of Claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds at least once every 4 program/erase cycles.

4. (Previously presented) The method of Claim 1, wherein determining, comparing, and selecting are performed in connection with at least one of flash initialization, read, or write operations for an end user.

5. (Original) The method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

6. (Original) The method of Claim 1, wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

7. (Original) The method of Claim 6, wherein the aUBER is 1E-16.

8. (Original) The method of Claim 1, wherein selecting further comprises selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

9. (Original) The method of Claim 1, further comprising:

receiving a command to perform formatting;

performing a memory test to determine raw bit error rates among the regions of the flash drive; and

allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

10. (Original) The method of Claim 1, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

11. (Original) The method of Claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor *SFx* is in a range from 1.5 to 4.

12. (Original) The method of Claim 10, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx = ECC\_CAPx / (Ex * RETx).$$

13. (Original) The method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

14. (Currently amended) The method of Claim 1, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

15. (Currently amended) An apparatus comprising:

a first circuit configured to determine a bit error rate associated with a region ~~associated with a plurality of~~ comprising a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region and configured to compare the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes, ~~wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity~~

~~symbols alone and one or more higher-latency ECC decode options with secondary parity symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower-latency ECC decode option and the higher-latency ECC decode option are different for the first gear and the second gear,~~ wherein the first gear has a different data payload size and correction capability than the second gear, wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear; and

a second circuit configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

16. (Previously presented) The apparatus of Claim 15, wherein the ECC encoder is configured to determine the bit error rate and the processor is configured to compare and to select in connection with at least one of flash initialization, read, or write operations for an end user.

17. (Original) The apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region.

18. (Original) The apparatus of Claim 15, wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

19. (Original) The apparatus of Claim 15, wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

20. (Original) The apparatus of Claim 15, wherein the processor is further configured to:

receive a command to perform formatting;

perform a memory test to determine raw bit error rates among the regions of the flash drive; and

allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

21. (Original) The apparatus of Claim 15, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

22. (Currently amended) The apparatus of Claim 15, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performance of ECC decoding with the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

23. (Original) The apparatus of Claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

24. (Currently amended) An apparatus for selecting an error correction coding (ECC) scheme, the apparatus comprising:

a means for determining a bit error rate associated with a region ~~associated with a plurality of~~ comprising a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region;

a means for comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, ~~wherein each gear of the set of predefined gears includes ECC parameters for a lower-latency ECC decode option with primary parity symbols alone and one or more higher-latency ECC decode options with secondary parity symbols and/or tertiary parity symbols, wherein at least ECC parameters for both the lower-latency ECC decode option and the higher-latency ECC decode option are different for the first gear and the second~~

~~gear,~~ wherein the first gear has a different data payload size and correction capability than the second gear, wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear; and

a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

25. (Original) The apparatus of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

26. (Original) The apparatus of Claim 24, wherein the selecting means further comprises means for selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

27. (Original) The apparatus of Claim 24, further comprising:

means for receiving a command to perform formatting;

means for performing a memory test to determine raw bit error rates among the regions of the flash drive; and

means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

28. (Original) The apparatus of Claim 24, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

29. (Currently amended) The apparatus of Claim 24, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein

the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

30. (Original) The apparatus of Claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

31. (New) The method of Claim 1, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region.

32. (New) The apparatus of Claim 15, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region.

33. (New) The apparatus of Claim 24, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region.

Application No.: **13/477,600**
Filing Date: **May 22, 2012**

SUMMARY OF INTERVIEW

Attendees, Date and Type of Interview

An interview was held on November 14, 2014 by telephone with participants Examiner Christian Dorman, inventor Phillip Northcott, and Applicant's representative Michael S. Okamoto.

Exhibits and/or Demonstrations

None.

Identification of Claims Discussed

1 and proposed new claim 31.

Identification of Prior Art Discussed

Diggs.

Proposed Amendments

Applicant proposed claim amendments to Claim 1 to distinguish over the art and proposed a new dependent claim.

Principal Arguments and Other Matters

The proposed amendment to Claim 1 recites that the region has a fixed number of two or more flash memory pages (*i.e.*, the region is of fixed size), and recites how the memory space is allocated within the region when changing ECC gears.

Results of Interview

The proposed amendments appear to distinguish Claim 1 over the currently cited references.

REMARKS

The July 16, 2014 Final Office Action rejects Claims 1-30 under 35 U.S.C. § 103(a). Applicant requests reconsideration of the rejections in view of the foregoing amendments and the following comments.

Rejections of Claims 1, 15, and 24 under 35 U.S.C. § 103(a)

The Office Action rejects Claims 1, 15, and 24 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2009/0070651 by Diggs ("Diggs") in view of U.S. Patent Application Publication No. 2013/0246891 by Manning ("Manning") further in view of U.S. Patent Application Publication No. 2013/0179754 by Cherubini, further in view of U.S. Patent No. 8,495,465 to Anholt ("Anholt"). Applicant has amended Claims 1, 15, and 24 to clarify distinctions over the cited references.

Initially, Applicant notes the subject matter of a prior amendment to Claims 1, 15, and 24 has been backed out and moved back to dependent Claims 14, 22, and 29.

Claims 1, 15, and 24 have been amended to recite that the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region" and "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

These amendments are supported by, for example, Tables II-IV of the application as filed. For example, the region can correspond to a page stripe and Table II illustrates an example with 4 pages per page stripe (and 4 page strips per page grid). Table III illustrates an example of the changing capacity of primary ECC with respect to gear. Table IV illustrates an example of the changing capacity of secondary ECC with respect to gear.

The Combination of References Does Not Teach All Elements

Host computers transfer data in fixed sizes (called allocation blocks or sectors). Once it is determined what the size of the allocation block is, the size does not change. Thus, when the data payload size changes within a region, this presents a difficult problem to handle. Rather than reallocation, in background paragraph **[0006]**, Diggs states:

> **[0006]** The ability of the storage subsystem to detect and correct errors can be increased by increasing the quantity of ECC bits stored per unit of data. However, such increases in the quantity of ECC used generally decreases the performance of the storage subsystem and may require additional (i.e., off-sector) storage space when the quantity of ECC data is larger than the amount of management data provided for each sector.

Applicant respectfully submits that the combination of Diggs, Manning, Cherubini, and Anholt does not teach all the claimed elements, in particular, the recited allocation of memory space. As stated in M.P.E.P. at § 2143.02, a "rationale to support a conclusion that a claim would have been obvious is that ***all the claimed elements were known in the prior art*** and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded nothing more than predictable results to one of ordinary skill in the art." M.P.E.P. § 2143.02; *see KSR v. Teleflex*, 550 U.S. at 398, 82 USPQ2d at 1395.

Proceeding Contrary to Accepted Wisdom

In addition, Applicant respectfully submits that to perform the dynamic reallocation of memory space within the region is proceeding contrary to accepted wisdom, which is evidence of nonobviousness. The totality of the prior art must be considered, and proceeding contrary to accepted wisdom in the art is evidence of nonobviousness. In re Hedges, 783 F.2d 1038, 228 USPQ 685 (Fed. Cir. 1986) "The totality of the prior art must be considered, and proceeding contrary to accepted wisdom in the art is evidence of nonobviousness." *In re Hedges*, 783 F.2d 1038, 228 USPQ 685 (Fed. Cir. 1986), M.P.E.P. § 2145(D)(3).

Accordingly, Applicant requests the Examiner to withdraw the rejections of Claims 1, 15, and 24, and to allow the same.

Rejection of Dependent Claims 2-14, 16-23, and 25-30

Dependent Claims 2-14, 16-23, and 25-30 depend from and further define Claims 1, 15, and 24, respectively. The dependent claims recite numerous additional distinctions over the cited references.

Applicant further notes that Claims 14, 22, and 29 were amended to revert to the original versions of these claims.

Claims 2-14, 16-23, and 25-30 add additional limitations that Applicant believes are not taught or suggested by the prior art. In addition, Applicant respectfully submits that dependent Claims 2-14, 16-23, and 25-30 are patentable for at least the reasons described for Claims 1, 15, and 24, respectively, and Applicant accordingly requests allowance of Claims 2-14, 16-23, and 25-30.

New Claims 31-33

Applicant has added new Claims 31, 32, and 33, which depends from Claims 1, 15, and 24, respectively, to further define the invention. Claims 31-33 each recite that "the secondary ECC parity symbols of each secondary codeword are spread out among each page of the region." This amendment is supported by, for example, original Claim 12 of related U.S. Patent Application No. 13/477,598, filed May 22, 2012, (Attorney Docket: DATUMTE.076A4) which had been incorporated by reference and is thus a part of this application's specification. Claim 12 in its original form recited the limitations of Claim 1 and "The method of Claim 1, wherein each primary codeword is programmed within a single page and each secondary codeword is programmed across the group of pages, and wherein the group of pages comprises a page stripe."

Applicant respectfully submits that the cited references do not teach or disclose new Claims 31-33, and that dependent Claims 31-33 are patentable for at least the reasons described for Claims 1, 15, and 24. Applicant accordingly requests allowance of Claims 31-33.

No Disclaimers or Disavowals

Although the present communication may include alterations to the application or claims, or characterizations of claim scope or referenced art, Applicant is not conceding in this application that previously pending claims are not patentable over the cited references. Rather, any alterations or characterizations are being made to facilitate expeditious prosecution of this

application. Applicant reserves the right to pursue at a later date any previously pending or other broader or narrower claims that capture any subject matter supported by the present disclosure, including subject matter found to be specifically disclaimed herein or by any prior prosecution. Accordingly, reviewers of this or any parent, child or related prosecution history shall not reasonably infer that Applicant has made any disclaimers or disavowals of any subject matter supported by the present application.

Co-Pending Applications of Assignee

Applicant wishes to draw the Examiner's attention to the following co-pending applications of the present application's assignee.

Docket No.	Serial No. / Patent No.	Title (as filed)	Filed
DATUMTE.076A2	13/477,633	SYSTEMS AND METHODS FOR INITIALIZING REGIONS OF A FLASH DRIVE HAVING DIVERSE ERROR CORRECTION CODING (ECC) SCHEMES	May 22, 2012
DATUMTE.076A3	13/477,629	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING AMONG DIFFERENT ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A4	13/477,598	SYSTEMS AND METHODS FOR REDUNDANTLY STORING ERROR CORRECTION CODES IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A5	13/477,595 8,788,910	SYSTEMS AND METHODS FOR LOW LATENCY, HIGH RELIABILITY ERROR CORRECTION IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A6	13/477,568	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A7	13/477,599	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A8	13/477,845 8,793,556	SYSTEMS AND METHODS FOR RECLAIMING FLASH BLOCKS OF A FLASH DRIVE	May 22, 2012

Application No.: **13/477,600**
Filing Date: **May 22, 2012**

Docket No.	**Serial No. / Patent No.**	**Title (as filed)**	**Filed**
DATUMTE.076A9	13/477,635	SYSTEMS AND METHODS FOR REDUNDANTLY STORING DATA IN A FLASH DRIVE SUCH THAT A FAILED PAGE CAN BE TOLERATED	May 22, 2012
DATUMTE.076A10	13/477,601	SYSTEMS AND METHODS FOR RECOVERING DATA FROM FAILED PORTIONS OF A FLASH DRIVE	May 22, 2012

Please charge any additional fees, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: November 17, 2014

By: _____

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

19134241
111714

INFORMATION DISCLOSURE STATEMENT

Inventor	:	Philip L. Northcott
App. No.	:	13/477600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY
Examiner	:	Dorman, Christian M.
Art Unit	:	2112
Conf. No.	:	2672

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. Copies of any listed foreign and non-patent literature references are being submitted.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed with an RCE and no fee is believed to be required.

The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment, to Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: November 17, 2014

By: _____

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

IDS

19135391
101614

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 2	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,279,830	08-28-2001	Ishibashi, Toshiharu	
	2	7,415,651	08-19-2008	Argon, Cenk	
	3	7,545,689	06-09-2009	Ilkbahar, et al.	
	4	8,255,620	08-25-2012	Frost, et al.	
	5	8,402,217	03-19-2013	Burd, Gregory	
	6	8,464,095	06-11-2013	Bonwick, Jeffrey S.	
	7	8,560,881	10-15-2013	Frost, et al.	
	8	8,689,082	04-01-2014	Oh, et al.	
	9	8,713,245	04-29-2014	Frost, et al.	
	10	8,713,411	04-29-2014	Kong, et al.	
	11	8,725,944	03-13-2014	Burd, Gregory	
	12	8,750,042	06-10-2014	Sharon, et al.	
	13	2011/0038203	02-17-2001	Camp, et al.	
	14	2011/0072196	03-24-2011	Forhan, et al.	
	15	2012/0110417	05-03-2012	D'Abreu, et al.	
	16	2012/0311406	12-06-2012	Ratnam, et al.	
	17	2013/0060565	03-07-2013	Nair, et al.	
	18	2013/0173955	07-04-2013	Hallak, et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T^1

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T^1
	19	USPTO; Office Action dated May 27, 2014, from related application U.S. Patent Application No. 13/477,629 (filed May 22, 2012).	
	20	USPTO; Office Action dated June 5, 2014, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T^1 - Place a check mark in this area when an English language Translation is attached.

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 2 OF 2	Attorney Docket No.	DATUMTE.076A1

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	21	USPTO; Office Action dated June 9, 2014, from related application U.S. Patent Application No. 13/477,845 (filed May 22, 2012).	
	22	USPTO; Office Action dated June 20, 2014, from related application U.S. Patent Application No. 13/477,595 (filed May 22, 2012).	
	23	USPTO; Office Action dated July 31, 2014, from related application U.S. Patent Application No. 13/477,568 (filed May 22, 2012).	
	24	USPTO; Office Action dated August 1, 2014, from related application U.S. Patent Application No. 13/477,633 (filed May 22, 2012).	
	25	USPTO; Office Action dated August 27, 2014, from related application U.S. Patent Application No. 13/844,448 (filed March 15, 2013).	
	26	USPTO; Office Action dated September 2, 2014, from related application U.S. Patent Application No. 13/844,499 (filed March 15, 2013).	
	27	USPTO; Office Action dated October 22, 2014, from related application U.S. Patent Application No. 13/477,599 (filed May 22, 2012).	
	28	USPTO; Office Action dated November 7, 2014, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	
	29	USPTO; Office Action dated November 7, 2014, from related application U.S. Patent Application No. 13/844,499 (filed March 15, 2013).	
	30	USPTO; Office Action dated November 13, 2014, from related application U.S. Patent Application No. 13/477,601 (filed May 22, 2012).	
	31	AHN, JONG-SUK, and HEIDMANN, JOHN, An Adaptive FEC Algorithm for Mobile Wireless Networks [online], March 25, 2002 [retrieved on November 13, 2014]. Retrieved from the Internet: < URL: http://www.isi.edu/~johnh/PAPERS/Ahn02a.pdf>.	
	32	3rd GENERATION PARTNERSHIP PROJECT, Technical Report, Technical Specification Group Radio Access Network, UTRA High Speed Downlink Packet Access (Release 4), 3GPP TR 25.950 v4.0.1, July 2005 [retrieved on November 13, 2014]. Retrieved from the Internet: < URL: http://www.arib.or.jp/IMT-2000/V900Jul11/5_Appendix/Rel4/25/25950-401.pdf>.	
	33	WIKIPEDIA, Asynchronous Transfer Mode, date unknown, [retrieved on November 17, 2014]. Retrieved from the Internet: < URL: http://en.wikipedia.org/w/index.php?title=Asynchronous_Transfer_Mode&printable=yes>.	

19133616
111714

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

Electronic Patent Application Fee Transmittal

Application Number:	13477600
Filing Date:	22-May-2012
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Filer:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Claims in Excess of 20	1202	3	80	240
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension - 1 month with $0 paid	1251	1	200	200
Miscellaneous:				
Request for Continued Examination	1801	1	1200	1200
		Total in USD ($)		**1640**

Electronic Acknowledgement Receipt

EFS ID:	20720068
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/Tony Do
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	17-NOV-2014
Filing Date:	22-MAY-2012
Time Stamp:	18:31:49
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$1640
RAM confirmation Number	5706
Deposit Account	111410
Authorized User	KNOBBE MARTENS OLSON AND BEAR

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	RCE_DATUMTE076A1.pdf	344718 49afb346e3b9c29fc06f94e5037c4f1d2ba6ed59	no	3
Warnings:					
This is not a USPTO supplied RCE SB30 form.					
Information:					
2		OAR_DATUMTE076A1.pdf	1447252 b2818bf50f52448ffe41ec1bdd220ae11ff3833c	yes	14

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment Submitted/Entered with Filing of CPA/RCE	1	1
Claims	2	8
Applicant summary of interview with examiner	9	9
Applicant Arguments/Remarks Made in an Amendment	10	14

Warnings:					
Information:					
3		IDS_DATUMTE076A1.pdf	456448 5926ad4456f3ef69195ab2041046f59f82641212	yes	4

Multipart Description/PDF files in .zip description

Document Description	Start	End
Transmittal Letter	1	2
Information Disclosure Statement (IDS) Form (SB08)	3	4

Warnings:					
Information:					
4	Non Patent Literature	NPL01_DATUMTE076A1.PDF	1974239 5edfb2e1e29e8824716da9e03fd6747995280902	no	47
Warnings:					
Information:					
5	Non Patent Literature	NPL02_DATUMTE076A1.PDF	1262248 e0e442a8fcfe1cc555c28a23cad68fcaaa1737da	no	28

Warnings:					
Information:					
6	Non Patent Literature	NPL03_DATUMTE076A1.PDF	257464 ec306a2d3e3a6e017751cfb502e1c7cd461e6446	no	6
Warnings:					
Information:					
7	Non Patent Literature	NPL04_DATUMTE076A1.PDF	280159 a7a24242fa3769867dd7a91395cdab35e1131570	no	7
Warnings:					
Information:					
8	Non Patent Literature	NPL05_DATUMTE076A1.PDF	1024091 073f168d30203d531f9d269867b381f5bf9d54f5	no	25
Warnings:					
Information:					
9	Non Patent Literature	NPL06_DATUMTE076A1.PDF	806962 c284c0f9d176f85b1745159312 6b75c4dee3763d	no	20
Warnings:					
Information:					
10	Non Patent Literature	NPL07_DATUMTE076A1.PDF	736630 81e0995ef9e0e9ecd0c8c93a4e550385c6caebd5	no	15
Warnings:					
Information:					
11	Non Patent Literature	NPL08_DATUMTE076A1.PDF	1099029 e7d755c885baf89c4509ff18d3d010e0faee4609	no	20
Warnings:					
Information:					
12	Non Patent Literature	NPL09_DATUMTE076A1.PDF	435351 3ead6d0febdcefaacad251d0092d21e08fb10974	no	6
Warnings:					
Information:					
13	Non Patent Literature	NPL10_DATUMTE076A1.PDF	945362 8e62cc0423d0c4b247986ba4c28cff651319bb43	no	24
Warnings:					
Information:					
14	Non Patent Literature	NPL11_DATUMTE076A1.PDF	195382 19c7b84109ec06e51260d7ea09b4a80a6063c9e9	no	5

Warnings:					
Information:					
15	Non Patent Literature	NPL12_DATUMTE076A1.PDF	724530 05cfd645188cd0fd8a1efbf9970d39d5efc0902d	no	14
Warnings:					
Information:					
16	Non Patent Literature	NPL13_DATUMTE076A1.pdf	331889 1b4321fa970fa310a838a789da520bf5ff5363ce	no	10
Warnings:					
Information:					
17	Non Patent Literature	NPL15_DATUMTE076A1.pdf	652771 3698e9c5bd4b1f1a99fcdcdce92ca2e42153b938	no	8
Warnings:					
Information:					
18	Non Patent Literature	NPL14_DATUMTE076A1.pdf	344180 1e0d0bfb930ed9b769147e24a0f3206803522f70	no	27
Warnings:					
Information:					
19	Fee Worksheet (SB06)	fee-info.pdf	34248 fda0fe260bdfb5785d520793a395335dcd1e08a8	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	13352953		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/477,600	Filing Date 05/22/2012	☐ To be Mailed

ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)		
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)		
	11/17/2014	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	* 33	Minus	** 30	= 3	X $80 =	240
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $420 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	**240**

		(Column 1)		(Column 2)	(Column 3)		
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/TAMMY d. MCBETH BROWN/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

20995 7590 02/02/2015

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
DORMAN, CHRISTIAN M	

ART UNIT	PAPER NUMBER
2112	

NOTIFICATION DATE	DELIVERY MODE
02/02/2015	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

jayna.cartee@knobbe.com
efiling@knobbe.com

Office Action Summary	Application No. 13/477,600	Applicant(s) NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 2112	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11/17/2014*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *1-33* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-33* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☒ The drawing(s) filed on *5/22/2012* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date *11/17/2014*.

3) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *11/14/2014* .

4) ☐ Other: _____.

DETAILED ACTION

1. This office action is sent in response to Applicant's Communication received on November 17th, 2014 for application number 13/477600. The office hereby acknowledges receipt of the following and placed of record in file: Claims and Applicant Arguments/Remarks made in an Amendment.

2. Claims 1-30 are presented for examination.

Notice of Pre-AIA or AIA Status

3. The present application is being examined under the pre-AIA first to invent provisions.

Information Disclosure Statement

4. The information disclosure statement (IDS) submitted on 11/17/2014 was filed before the mailing date of this office action. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

Response to Amendments and Arguments

5. The applicant's amendments made to claims 1, 15, and 24, filed 11/17/2014, have been fully considered and have been rejected upon the application of further prior art. Claims 1, 15, and 24 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651), in view of Manning (U.S. Patent Application Publication 2013/0246891), further in view of Agarwal [U.S. Patent 8,433,981]. The applicant's amendments made to claims 1, 15, and 24 consisted of moving previously amended language moved up from claims 14, 22, and

29, respectively, back into said claims. The applicant further amended claims 1, 15, and 24 to recite wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region" and "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear." Regarding the first part of the amendment (regarding the fixed number of pages), while Diggs [U.S. Patent Application Publication 2009/0070651] teaches that data can be stored in pages [paragraph 0063] and Manning [U.S. Patent Application Publication 2013/0246891] teaches wherein a fixed number of codewords are stored in each page of data [paragraph 0023], these references are not seen to explicitly teach wherein a fixed number of two or more flash memory pages or integer fractions thereof are used. Thereby, said component of the amendment is rejected upon the application of further prior art. Specifically, Agarwal [U.S. Patent 8,433,981] teaches wherein blocks of flash memory comprise fixed numbers of pages [column 2, lines 38-48]. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Agarwal to the method of Diggs and Manning wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region." Motivation to do so would have been to allow the flash memory used to store data organized into specific

groups for reading and writing [Agarwal, column 2, lines 38-48]. Regarding the second part of the amendment (payload size changes with gears), while Diggs fails to explicitly teach said amendment, upon further examination the examiner found that Manning cures this deficiency. In the applicant initiated interview on 11/14/2014 the applicant argued and the examiner agreed that Diggs [U.S. Patent Application Publication 2009/0070651] teaches wherein the amount of parity information stored in a region doesn't affect the data payload of said region [Diggs, paragraph 0005]. The examiner apologizes for not identifying that the other prior art used in the previous rejection could be used to reject this part of the amended claim language. Specifically, Manning teaches wherein when codewords are used with different levels of error correction wherein each fixed size codeword stores data payload and ECC information, as the amount of ECC is increased the space for data payload in each codeword decreases [paragraph 0023]. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear." Motivation to do so would have been to allow for adjustable amount of error correction information to be used to protect the data payload [Manning, paragraph 0023].

6. The applicant's newly added claims 31, 32, and 33, filed 11/17/2014, have been fully considered and have been rejected upon the application of further prior art. Claims 31, 32, and 33 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over

Diggs (U.S. Patent Application Publication 2009/0070651), in view of Manning (U.S. Patent Application Publication 2013/0246891), in view of Agarwal [U.S. Patent 8,433,981], further in view of D'Abreu [U.S. Patent Application Publication 2012/0110417], wherein D'Abreu is the further prior art, teach the claimed limitations. Specifically, D'Abreu teaches wherein the secondary parity symbols are spread out across each page of the group of pages when programmed [figure 1, item 120, ECC page includes data blocks with their own error correction, items 122 and 123, etc., and an additional ECC Data, item 134; Paragraphs 003 and 0004, is two levels of error correction within each data page]. The examiner further notes that claims 31, 32, and 33 are rejected under 35 U.S.C. 112(b) because it is unclear what the secondary codewords are, as no primary codewords are defined in claims 31, 32, and 33 or the claims on which they are dependent.

7. The rejection is maintained, see below.

Claim Rejections - 35 USC § 112

8. Claims 31, 32, and 33 rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention. Specifically, claims 31, 32, and 33 are rejected under 35 U.S.C. 112(b) because it is unclear what the secondary codewords are, as no primary codewords are defined in claims 31, 32, and 33 or the claims on which they are dependent.

Claim Rejections - 35 USC § 103

9. The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-5, 8, 9, 14-16, 17, 19, 20, 22-27, 29, and 30 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651), in view of Manning (U.S. Patent Application Publication 2013/0246891), further in view of Agarwal [U.S. Patent 8,433,981].

Regarding claim 1, Diggs teaches a method of selecting an error correction coding (ECC) scheme, the method comprising: determining [by monitoring] a bit error rate associated with a region [paragraph 0029, lines 10-13 (the bit error rate of data is monitored)] associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]; comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes [paragraph 0025 (in one embodiment two different ECC

settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used)]; wherein determining, comparing, and selecting are performed by an integrated circuit [paragraph 0021, lines 8-10 (controller is implemented as an integrated circuit), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)].

Diggs is not seen to explicitly teach that integer fractions of pages can be used. Diggs further is not seen to explicitly teach wherein "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

Manning in an analogous art teaches wherein when codewords are used with different levels of error correction wherein each fixed size codeword stores data payload and ECC information, as the amount of ECC is increased the space for data payload in

each codeword decreases [paragraph 0023]. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear." Motivation to do so would have been to allow for adjustable amount of error correction information to be used to protect the data payload [Manning, paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used and "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

Diggs and Manning are not seen to explicitly teach wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein

the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region."

Agarwal in an analogous art teaches wherein blocks of flash memory comprise fixed numbers of pages [column 2, lines 38-48].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Agarwal to the method of Diggs and Manning wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region."

Motivation to do so would have been to allow the flash memory used to store data organized into specific groups for reading and writing [Agarwal, column 2, lines 38-48].

Regarding claim 2, Diggs discloses the method of claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds such that the bit error rate is actively monitored [paragraph 0031, lines 1-4 (the controller can be set up to repeatedly monitor the data and set the ECC level based on the reading), paragraph 0051 and paragraph 0056 (the data monitored can be the bit error rate monitored and can be compared to threshold values)].

Regarding claim 3, Diggs discloses the method of claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate

monitored and compared to threshold values)] at least once every 4 program/erase cycles [paragraph 0045, lines 5-11 (the block erase count can be monitored) and paragraph 0031, lines 1-4 (events can be used to monitor the data and set the ECC level periodically)].

Regarding claim 4, Diggs teaches the method of claim 1, wherein determining, comparing, and selecting are performed in connection with at least one of flash initialization, read, or write operations for an end user [figure 2, item 203, during write operations, the ECC setting is determined before encoding the data with the ECC level that is determined].

Regarding claim 5, Diggs discloses the method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

Regarding claim 8, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding iterations (paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level

means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs wherein selecting further comprises selecting a gear having a smaller payload size and a higher amount of correction capability based at least partly on the number of decoding iterations used to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

Regarding claim 9, Diggs discloses the method of claim 1, further comprising: receiving a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; performing a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to

determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

Regarding claim 15, Diggs teaches an apparatus comprising: a first circuit [monitoring circuit] configured to determine a bit error rate [paragraph 0029, lines 10-13 (the bit error rate of data is monitored) and paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] associated with a region associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)] and to compare the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037,

lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and a second circuit [controller] configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used) and paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)].

Diggs is not seen to explicitly teach that integer fractions of pages can be used. Diggs further is not seen to explicitly teach wherein "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

Manning in an analogous art teaches wherein when codewords are used with different levels of error correction wherein each fixed size codeword stores data payload and ECC information, as the amount of ECC is increased the space for data payload in each codeword decreases [paragraph 0023]. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the

second gear." Motivation to do so would have been to allow for adjustable amount of error correction information to be used to protect the data payload [Manning, paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the apparatus of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used and "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

Diggs and Manning are not seen to explicitly teach wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region."

Agarwal in an analogous art teaches wherein blocks of flash memory comprise fixed numbers of pages [column 2, lines 38-48].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Agarwal to the method of Diggs and Manning

wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region."

Motivation to do so would have been to allow the flash memory used to store data organized into specific groups for reading and writing [Agarwal, column 2, lines 38-48].

Regarding claim 16, Diggs teaches the apparatus of claim 1, wherein the ECC encoder is configured to determine the bit error rate and the processor is configured to compare and to select in connection with at least one of flash initialization, read, or write operations for an end user [figure 2, item 203, during write operations, the ECC setting is determined before encoding the data with the ECC level that is determined].

Regarding claim 17, Diggs discloses the apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

Regarding claim 19, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding

iterations (paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the apparatus of Diggs wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

Regarding claim 20, Diggs discloses the apparatus of Claim 15, wherein the processor is further configured to: receive a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; perform a memory test to determine raw bit error rates among the regions

of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

Regarding claim 23, Diggs discloses the apparatus of claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive [paragraphs 0061 and 0062 (show how the apparatus could be embodied in this way) and claim 14].

Regarding claim 24, Diggs teaches an apparatus [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for selecting an error correction coding (ECC) scheme, the apparatus comprising: a means for determining a bit error rate [paragraph 0029, lines 10-13 (the bit error rate of data is monitored)] associated with a region associated with a plurality of flash memory pages or integer fractions thereof [paragraph 0016, lines 3-11 (blocks of data are monitored at a time), paragraph 0037, lines 17-18 (data blocks having different formats and sizes may be utilized), paragraph 0043, lines 1-2, and paragraph 0063, lines 11-13 (ECC levels can be specified at any data level where the storage device can be divided into pages)]; a means for comparing the determined bit error rate to one or more predetermined thresholds [paragraph 0051 and paragraph 0056 (bit error rate monitored and compared to threshold values)] corresponding to a

set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes [paragraph 0025 (in one embodiment two different ECC settings are used, wherein both of these settings have different ECC schemes)], wherein the first gear has a different data payload size and correction capability than the second gear [paragraph 0024 (different ECC settings correspond to different ratios of ECC bits to data), paragraph 0025 (the gears have different amounts of ECC information encoded, corresponding to different correction capability), paragraph 0037, lines 7-18 (different formats and size of data blocks can be used, could have different data payload sizes)]; and a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds [paragraph 0030 (the ECC level is set by analyzing the monitor data and setting the ECC settings based on that) and paragraph 0051 and paragraph 0056 (threshold values are used)].

Diggs is not seen to explicitly teach that integer fractions of pages can be used. Diggs further is not seen to explicitly teach wherein "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

Manning in an analogous art teaches wherein when codewords are used with different levels of error correction wherein each fixed size codeword stores data payload and ECC information, as the amount of ECC is increased the space for data payload in each codeword decreases [paragraph 0023]. It would have been obvious to one of

ordinary skill in the art at the time of the invention to apply the teachings of Manning to the method of Diggs "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear." Motivation to do so would have been to allow for adjustable amount of error correction information to be used to protect the data payload [Manning, paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the apparatus of Diggs and have it where integer fractions of pages are used and where the data payload size is different for the level of ECC used and "wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear."

Motivation to do so would be to have the preferred page stripe layout [paragraph 0032, lines 3-6] for using integer fractions of pages and to optimize the memory density of the data [paragraph 0023, lines 19-27] for each gear having a different data payload size.

Diggs and Manning are not seen to explicitly teach wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region."

Agarwal in an analogous art teaches wherein blocks of flash memory comprise fixed numbers of pages [column 2, lines 38-48].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Agarwal to the method of Diggs and Manning wherein the region comprises "a fixed number of two or more flash memory pages or integer fractions thereof, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region."

Motivation to do so would have been to allow the flash memory used to store data organized into specific groups for reading and writing [Agarwal, column 2, lines 38-48].

Regarding claim 25, Diggs discloses the apparatus [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region [paragraph 0051 (a high monitored bit error rate is an indicator of a high future probability of a data error occurring. Because of this, the monitored bit error rate can be estimated)].

Regarding claim 26, Diggs teaches wherein the wear level can be used as a measure of the level of error correction needed for the memory. The wear level can indicate how many times the memory has been erased, which corresponds to decoding

iterations [paragraph 0045, lines 4-11]. Diggs also teaches wherein a higher amount of correction capability is used based on an increased wear level [paragraph 0008].

Diggs fails to explicitly teach wherein the payload size decreases as the correction capability increases [paragraph 0024 teaches wherein a higher ECC level means more ECC bits per unit of data and paragraph 0037 teaches wherein the blocks monitored can be of different formats and sizes].

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the smaller the data payload size, the worse the memory density (ratio of data to ECC bits) and the better the error correcting ability (because of the lower data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one with ordinary skill in the art at the time of the invention to apply the teachings of Manning to the apparatus of Diggs wherein the selecting means further comprises means [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

Motivation to do so would have been to optimize the memory density of the data for each gear having a different data payload size [Manning, paragraph 0023, lines 19-27].

Regarding claim 27, Diggs discloses the apparatus of Claim 24, further comprising: means [paragraph 0021, lines 8-10 (controller can be implemented as multiple integrated circuits), paragraph 0029, lines 10-13 (controller determines monitor data), paragraph 0030, lines 1-3 (controller comparing and selecting)] for receiving a command to perform formatting [paragraph 0031, lines 1-4 (can perform monitoring and set ECC level in response to events, like formatting)]; means for performing a memory test to determine raw bit error rates among the regions of the flash drive [paragraph 0029, lines 10-14 (bit error rate can be determined for the memory regions)]; and means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region [paragraph 0031 (information used to determine the level of error correction used for said region), paragraph 0025 (an embodiment where two levels of ECC code are used)].

Regarding claim 30, Diggs discloses the apparatus of claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive [paragraphs 0061 and 0062 (show how the apparatus could be embodied in this way) and claim 14].

11. Claims 6, 7, 10-13, 18, 21, and 28 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651) in view of Manning (U.S. Patent Application Publication 2013/0246891), in view of Agarwal [U.S. Patent 8,433,981] as applied to claims 1, 15, and 24 above, further in view of Bennett (U.S. Patent Application Publication 2013/0080862).

Regarding claim 6, Diggs teaches wherein a level of ECC is encoded into a given set of data dependent on comparing information monitored from said data with certain threshold values, where the threshold can be the bit error rate. [paragraph 0008, lines11-15 and paragraph 0029, lines 12-13]. Diggs, however, does not explicitly teach wherein the highest data payload size is used that still has sufficient correction ability for the determined bit error rate.

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the higher the data payload size, the better the memory density (ratio of data to ECC bits) and the worse the error correcting ability (because of the higher data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs so that selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate.

Motivation to do so would have been to achieve the best memory density possible [paragraph 0023, lines 19-23].

Neither Diggs nor Manning teach wherein there is a predetermined acceptable uncorrectable bit error rate.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Bennets to the modified method of Diggs with

Manning wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

Regarding claim 7, Diggs and Manning fail to teach where the aUBER is 1E-16.

Bennett teaches wherein the aUBER can be 1E-18 and furthermore can be set to required level [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Bennett to the method of Diggs and Manning such that the aUBER is 1E-16.

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

Regarding claim 10, Diggs further teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be

combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs and Manning fail to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the method of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Regarding claim 11, Diggs further teaches wherein the threshold values used can be modified by different functions and set to any value necessary to achieve the desired level of error correction [paragraphs 0050 and 0056]. Thereby, it could be had the method of claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor SFx is in a range from 1.5 to 4.

Regarding claim 12, Diggs further teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056]. Furthermore Diggs teaches that a high monitored bit error rate anticipates a high probability of future errors occurring and because of this the ECC setting is adjusted [paragraph 0051]. This information corresponds to the excursion factor times the retention factor, which is the safety factor, because they both anticipate the increased probability of future errors occuring.

Diggs and Manning fail to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by

multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate that is set to a required value [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs and Manning with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction and have this value be divided by a safety factor, where the safety factor is found by the given expression as a product of the excursion factor and the retention factor. Diggs teaches where these threshold values can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system and the safety factor is itself found as a product of the excursion factor and the retention factor. Thereby, the method of Diggs would be modified by the teachings of Bennett wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor, where the safety factor is equal to the excursion factor times the retention factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Regarding claim 13, Diggs teaches wherein the threshold values used can be modified by different functions and set to any value necessary to achieve the desired

level of error correction [paragraphs 0050 and 0056]. Thereby, it can be had the method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

Regarding claim 18, Diggs teaches wherein a level of ECC is encoded into a given set of data dependent on comparing information monitored from said data with certain threshold values, where the threshold can be the bit error rate. [paragraph 0008, lines11-15 and paragraph 0029, lines 12-13]. Diggs, however, does not explicitly teach wherein the highest data payload size is used that still has sufficient correction ability for the determined bit error rate.

Manning teaches wherein there is a tradeoff between data payload size and error correcting ability in codewords. It is taught that the higher the data payload size, the better the memory density (ratio of data to ECC bits) and the worse the error correcting ability (because of the higher data to ECC bit ratio) [paragraph 0023].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teachings of Manning to the method of Diggs so that selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate.

Motivation to do so would have been to achieve the best memory density possible [paragraph 0023, lines 19-23].

Diggs and Manning fail to teach wherein there is a predetermined acceptable uncorrectable bit error rate.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Manning and Bennets to the apparatus of Diggs wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

The motivation to apply the teachings of Bennett would be to operate at the maximum error rate while still guaranteeing the required overall uncorrected bit error rate [paragraph 0039].

Regarding claim 21, Diggs teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs 0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs fails to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the apparatus of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

Regarding claim 28, Diggs teaches wherein there are thresholds based on weighted data monitored from the memory system wherein these thresholds can be based on a maximum allowable bit error rate and the ECC setting is set to provide sufficient error correcting ability based on the threshold being reached [paragraphs

0008, lines 11-15 and paragraph 0031]. Diggs also teaches that monitor data can be combined according to different functions to come to the desired threshold values that allow for a desired error correction setting [paragraphs 0050 and 0056].

Diggs fails to explicitly teach wherein there is a maximum allowable uncorrectable bit error rate that is adjusted by a safety factor which is found by multiplying the excursion factor (a bit error rate compensating for variability in the system) and a safety factor.

Bennett teaches wherein there is a predetermined acceptable uncorrectable bit error rate [paragraphs 0032 and 0039].

It would have been obvious to one with ordinary skill in the art at the time of the invention to modify the method of Diggs with the teachings of Bennett to allow for the uncorrectable bit error rate to be a threshold value used in determining the level of error correction. Diggs teaches where this value can be a product of a function in order to achieve a required level, like where the uncorrectable bit error rate is divided by the safety factor to anticipate changes in the memory system. Thereby, the apparatus of Diggs would be modified by the teachings of Bennett wherein a threshold Gx of the one or more predetermined thresholds is less than or equal to a raw bit error rate ECC_CAP for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor SFx, wherein the safety factor SFx is a product of an excursion factor Ex and a retention factor RETx, such that

the threshold is less than or equal to the bit error rate threshold divided by the safety factor.

Motivation to do so would be to compensate for a future probability of data errors occurring [Diggs paragraph 0051].

12. Claims 14, 22, and 29 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651), in view of Manning (U.S. Patent Application Publication 2013/0246891), in view of Agarwal [U.S. Patent 8,433,981] as applied to claims 1, 15, and 24 above, further in view of Cherubini (U.S. Patent Application Publication 2013/0179754).

Regarding claim 14, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the method of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using

the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

Regarding claim 22, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

Regarding claim 29, Diggs further teaches wherein different decoding schemes are used with varying levels of error correcting ability where there is a tradeoff between error correcting ability and the data bits to ECC bits ratio [paragraphs 0008 and 0024].

However, Diggs and Manning fail to teach wherein the parameter that is adjusted for decoding the data is the latency of the decoding scheme.

Cherubini teaches that latency is a critical parameter in the decoding process wherein a higher latency allows for a reduced complexity of implementation, so it is desirable to have the reduced complexity while staying within the acceptable limits of the latency seen in the decoding process [paragraph 0084].

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Cherubini to the apparatus of Diggs wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

Motivation to do so would have been to have reduced complexity of implementation while still staying within the acceptable limits of the latency for operation of the memory system [Cherubini paragraph 0084].

13. Claims 31-33 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Diggs (U.S. Patent Application Publication 2009/0070651), in view of Manning (U.S. Patent Application Publication 2013/0246891), in view of Agarwal [U.S. Patent 8,433,981] as applied to claims 1, 15, and 24 above, further in view of D'Abreu [U.S. Patent Application Publication 2012/0110417].

Regarding claim 31, Diggs, Manning, and Aggarwal are not seen to explicitly teach "the method of claim 1, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region."

D'Abreu in an analogous art teaches wherein the secondary parity symbols are spread out across each page of the group of pages when programmed [figure 1, item 120, ECC page includes data blocks with their own error correction, items 122 and 123, etc., and an additional ECC Data, item 134; Paragraphs 003 and 0004, is two levels of error correction within each data page].

D'Abreu in an analogous art teaches wherein the secondary parity symbols are arranged for distribution among the group of pages such that the secondary parity symbols are spread out across each page of the group of pages when programmed [figure 1, item 120, data page is programmed with primary and secondary parity for the sub-blocks and data block as a whole]. Thereby, D'Abreu teaches wherein the secondary parity data as taught by Frost and Jeddeloh could be incorporated into the pages of the data rather than stored on separate pages.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the method of Diggs, Manning, and Agarwal and apply the teachings of D'Abreu wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region.

Motivation to do so would have been to be able to correct uncorrectable errors in a page of data by using additional parity information [D'Abreu, abstract].

Regarding claim 32, claim 32 is analogous to claim 31 and is thereby rejected for the same reasons as those given above for claim 31.

Regarding claim 33, claim 33 is analogous to claim 31 and is thereby rejected for the same reasons as those given above for claim 31.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTIAN DORMAN whose telephone number is (571)272-7657. The examiner can normally be reached on Monday-Friday 9:30am-3pm, Friday alternating.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert Decady can be reached on 571-272-3819. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHRISTIAN DORMAN/
Examiner, Art Unit 2112

/ALBERT DECADY/
Supervisory Patent Examiner, Art Unit 2112

Applicant-Initiated Interview Summary	**Application No.** 13/477,600	**Applicant(s)** NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	**Art Unit** 2112	

All participants (applicant, applicant's representative, PTO personnel):

(1) *CHRISTIAN DORMAN*. (3) *Michael S. Okamoto*.

(2) *Phillip Northcott*. (4)_____.

Date of Interview: *14 November 2014*.

Type: ☒ Telephonic ☐ Video Conference
☐ Personal [copy given to: ☐ applicant ☐ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☒ No.
If Yes, brief description: _____.

Issues Discussed ☐101 ☐112 ☐102 ☒103 ☐Others
(For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: *1 and 31*.

Identification of prior art discussed: *Diggs [U.S. Patent Application Publication 2009/0070651]*.

Substance of Interview
(For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

During the interview the applicant explained how for their invention it is a fixed number of of pages and the size allocated for information and parity is also fixed so that the data payload decreases as more parity is added. The examiner agreed that amending this idea into claim 1 overcomes the Diggs reference, wherein data payload size changes with gear setting.The applicant further explained how they define data payload to refer to what the upper layer (host and journaling engine) consider the data.

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

☐ Attachment

/CHRISTIAN DORMAN/ Examiner, Art Unit 2112	

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

———

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Notice of References Cited	Application/Control No. 13/477,600	Applicant(s)/Patent Under Reexamination NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 2112	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2012/0110417	05-2012	D' Abreu et al.	714/773
*	B	US-8,433,981	04-2013	Agarwal et al.	714/773
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	61	("20110038203" | "20110072196" | "20120110417" | "20120311406" | "20130060565" | "20130173955" | "6279830" | "7415651" | "7545689" | "8255620" | "8402217" | "8464095" | "8560881" | "8689082" | "8713245" | "8713411" | "8725944" | "8750042").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:35
S2	25071	G06F11/1004.cpc. G06F11/1008.cpc. G06F11/1044.cpc. G06F11/1048.cpc. G06F11/1068.cpc. G06F11/1072.cpc. 714/763.ccls. 714/773.ccls. 714/774.ccls. 714/746.ccls. 714/752.ccls. 714/769.ccls. 714/779.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:37
S3	17	("20090070651" | "20090327840" | "20110072196" | "20110213920" | "20130173955" | "20130246891" | "8612680").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:40
S4	12	("20090070651" | "20130080862" | "20130179754" | "20130246891" | "8495465").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:40
S5	2	S4 and payload	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:44
S6	30	fixed adj number with page with flash	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:30
S7	25116	G06F11/1004.cpc. G06F11/1008.cpc. G06F11/1044.cpc. G06F11/1048.cpc. G06F11/1068.cpc. G06F11/1072.cpc. 714/763.ccls. 714/773.ccls. 714/774.ccls. 714/746.ccls. 714/752.ccls. 714/769.ccls. 714/779.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:30

S8	13	S6 and S7	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:30
S9	7	"20090070651".pn. "20130179754".pn. "20130246891".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:32
S10	2	"8560881".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 19:15

1/24/2015 7:34:12 PM
C:\Users\cdorman\Documents\EAST\Workspaces\13477600_RCE.wsp

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	58	"6279830".pn. "7415651".pn. "7545689".pn. "8255620".pn. "8402217".pn. "8464095".pn. "8621318".pn. "8689082".pn. "8713411".pn. "8725944".pn. "8750042".pn. "20100023800".pn. "20110038203".pn. "20120110417".pn. "20120311406".pn. "20130060565".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 07:19
S2	21	(US-20110258380-$ or US-20130246891-$ or US-20130173955-$ or US-20100268876-$ or US-20110213920-$ or US-20090327840-$ or US-20090070651-$ or US-20110072196-$).did. or (US-8560881-$ or US-4837447-$ or US-8612680-$).did.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 07:19
S3	0	S1 and (payload pay$1load)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:12
S4	17840	G06F11/1068.cpc. G06F11/1072.cpc. G06F11/1076.cpc. G06F11/1044.cpc. G06F11/1004.cpc. G06F11/1008.cpc. 714/773.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:12
S5	1766	(payload pay$1load packet) with (ECC (error adj (correct$4 detect$4)) parity protect$3 EDC) near5 (variable var$3 varying depend$4 chang$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:15
S6	17	("20090070651" | "20090327840" | "20110072196" | "20110213920" | "20130173955" | "20130246891" | "8612680").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:18
S7	48	S5 and S4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:18

S8	4127	((flash with memory) SSD (solid adj state) solid$1state non$1volatile (non adj volatile)) and (payload near2 size)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:34
S9	472	((flash with memory) SSD (solid adj state) solid$1state non$1volatile (non adj volatile)) and (payload near2 size) with (chang$3 variable var$4 modif$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:35
S10	0	"714".ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:36
S11	144878	"714".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:36
S12	30	S11 and S9	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/12 08:36
S13	112	("20020120820" | "20020159285" | "20030088821" | "20030112879" | "20030156454" | "20040268065" | "20050114587" | "20060221752" | "20070061689" | "20070124647" | "20070171714" | "20070171730" | "20070233752" | "20070266295" | "20070266296" | "20070277066" | "20080010582" | "20080034272" | "20080137414" | "20080163023" | "20080168319" | "20080172589" | "20080294960" | "20080320361" | "20080320373" | "20090013233" | "20090013234" | "20090037627" | "20090070651" | "20090100307" | "20090132889" | "20090164836" | "20090182939" | "20090327589" | "20090327840" | "20100008214" | "20100023800" | "20100115376" | "20100199153" | "20100281341" | "20100332894" | "20100332922" | "20110019475" | "20110029716" | "20110038203" | "20110040926" | "20110060969" | "20110167199" | "20110167319" | "20110213920" | "20120072680" | "20120089767" | "20120260146" | "20120311406"	US-PGPUB; USPAT; USOCR	OR	ON	2015/01/12 08:48

		"20130060565" | "20130073895" | "20130080862" | "20130151914" | "20130173955" | "20130179754" | "20130246891" | "4112502" | "5404485" | "5592497" | "5942004" | "5956743" | "5974544" | "6023781" | "6279133" | "6279830" | "6574774" | "6591394" | "6628723" | "6651212" | "6675318" | "6731538" | "6732322" | "6732325" | "6751766" | "6839870" | "6871303" | "6895543" | "7061804" | "7184356" | "7389465" | "7415651" | "7451383" | "7478314" | "7643342" | "7739576" | "7783955" | "7809900" | "7844879" | "7860200" | "7904619" | "7904672" | "7904780" | "7944748" | "7962831" | "8046542" | "8051358" | "8065583" | "8090980" | "8145855").PN. OR ("8335951" | "8347138" | "8392791" | "8433979" | "8495465" | "8560881" | "8612680" | "8621318").PN. OR ("8793556").URPN.				

1/24/2015 7:34:38 PM
C:\Users\cdorman\Documents\EAST\Workspaces\13477568_RCE.wsp

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 2	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,279,830	08-28-2001	Ishibashi, Toshiharu	
	2	7,415,651	08-19-2008	Argon, Cenk	
	3	7,545,689	06-09-2009	Ilkbahar, et al.	
	4	8,255,620	08-25-2012	Frost, et al.	
	5	8,402,217	03-19-2013	Burd, Gregory	
	6	8,464,095	06-11-2013	Bonwick, Jeffrey S.	
	7	8,560,881	10-15-2013	Frost, et al.	
	8	8,689,082	04-01-2014	Oh, et al.	
	9	8,713,245	04-29-2014	Frost, et al.	
	10	8,713,411	04-29-2014	Kong, et al.	
	11	8,725,944	03-13-2014	Burd, Gregory	
	12	8,750,042	06-10-2014	Sharon, et al.	
	13	2011/0038203	02-17-2001	Camp, et al.	
	14	2011/0072196	03-24-2011	Forhan, et al.	
	15	2012/0110417	05-03-2012	D'Abreu, et al.	
	16	2012/0311406	12-06-2012	Ratnam, et al.	
	17	2013/0060565	03-07-2013	Nair, et al.	
	18	2013/0173955	07-04-2013	Hallak, et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	19	USPTO; Office Action dated May 27, 2014, from related application U.S. Patent Application No. 13/477,629 (filed May 22, 2012).	
	20	USPTO; Office Action dated June 5, 2014, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 2 OF 2	Attorney Docket No.	DATUMTE.076A1

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	21	USPTO; Office Action dated June 9, 2014, from related application U.S. Patent Application No. 13/477,845 (filed May 22, 2012).	
	22	USPTO; Office Action dated June 20, 2014, from related application U.S. Patent Application No. 13/477,595 (filed May 22, 2012).	
	23	USPTO; Office Action dated July 31, 2014, from related application U.S. Patent Application No. 13/477,568 (filed May 22, 2012).	
	24	USPTO; Office Action dated August 1, 2014, from related application U.S. Patent Application No. 13/477,633 (filed May 22, 2012).	
	25	USPTO; Office Action dated August 27, 2014, from related application U.S. Patent Application No. 13/844,448 (filed March 15, 2013).	
	26	USPTO; Office Action dated September 2, 2014, from related application U.S. Patent Application No. 13/844,499 (filed March 15, 2013).	
	27	USPTO; Office Action dated October 22, 2014, from related application U.S. Patent Application No. 13/477,599 (filed May 22, 2012).	
	28	USPTO; Office Action dated November 7, 2014, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	
	29	USPTO; Office Action dated November 7, 2014, from related application U.S. Patent Application No. 13/844,499 (filed March 15, 2013).	
	30	USPTO; Office Action dated November 13, 2014, from related application U.S. Patent Application No. 13/477,601 (filed May 22, 2012).	
	31	AHN, JONG-SUK, and HEIDMANN, JOHN, An Adaptive FEC Algorithm for Mobile Wireless Networks [online], March 25, 2002 [retrieved on November 13, 2014]. Retrieved from the Internet: < URL: http://www.isi.edu/~johnh/PAPERS/Ahn02a.pdf>.	
	32	3rd GENERATION PARTNERSHIP PROJECT, Technical Report, Technical Specification Group Radio Access Network, UTRA High Speed Downlink Packet Access (Release 4), 3GPP TR 25.950 v4.0.1, July 2005 [retrieved on November 13, 2014]. Retrieved from the Internet: < URL: http://www.arib.or.jp/IMT-2000/V900Jul11/5_Appendix/Rel4/25/25950-401.pdf>.	
	33	WIKIPEDIA, Asynchronous Transfer Mode, date unknown, [retrieved on November 17, 2014]. Retrieved from the Internet: < URL: http://en.wikipedia.org/w/index.php?title=Asynchronous_Transfer_Mode&printable=yes>.	

19133616
111714

Examiner Signature	/Christian Dorman/	Date Considered	01/20/2015

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	2112

CPC- SEARCHED		
Symbol	Date	Examiner
G06F11/1004,1008,1044,1048,1068,1072	1/12,20,24/2015	CD

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
714	773	12/16/2013, 12/18/2013, 12/19/2013, 6/02/2014	CD
714	763,769,773,774,779	1/12,20,24/2015	CD

SEARCH NOTES		
Search Notes	Date	Examiner
EAST US_PGPUB, USPAT, USOCR, FPRS, EPO, JPO, DERWENT, IBM_TDB - search results are attached	12/16/2013, 12/18/2013, 12/19/2013, 6/02/2014, 6/03/2014, 6/05/2014, 1/12,20,24/2015	CD
Inventor Name Search, PALM	12/16/2013, 12/19/2013	CD
Double Patenting Search, PALM	12/16/2013, 12/19/2013	CD
IEEE NPL Search	12/18/2013	CD

INTERFERENCE SEARCH	

US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	2112

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/19/2013	06/16/2014	01/24/2015						
	1	✓	✓	✓						
	2	✓	✓	✓						
	3	✓	✓	✓						
	4	✓	✓	✓						
	5	✓	✓	✓						
	6	✓	✓	✓						
	7	✓	✓	✓						
	8	✓	✓	✓						
	9	✓	✓	✓						
	10	✓	✓	✓						
	11	✓	✓	✓						
	12	✓	✓	✓						
	13	✓	✓	✓						
	14	✓	✓	✓						
	15	✓	✓	✓						
	16	✓	✓	✓						
	17	✓	✓	✓						
	18	✓	✓	✓						
	19	✓	✓	✓						
	20	✓	✓	✓						
	21	✓	✓	✓						
	22	✓	✓	✓						
	23	✓	✓	✓						
	24	✓	✓	✓						
	25	✓	✓	✓						
	26	✓	✓	✓						
	27	✓	✓	✓						
	28	✓	✓	✓						
	29	✓	✓	✓						
	30	✓	✓	✓						
	31			✓						
	32			✓						
	33			✓						

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor	:	Philip L. Northcott
App. No.	:	13/477,600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY
Examiner	:	Dorman, Christian M.
Art Unit	:	2112
Conf. No.	:	2672

Amendment and Response to Office Action of February 2, 2015

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully responds to the Office Action of February 2, 2015 as set forth below.

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Remarks/Arguments begin on page 9 of this paper.

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method of selecting an error correction coding (ECC) scheme, the method comprising:

determining a bit error rate associated with a region comprising a fixed number of two or more flash memory pages or integer fractions thereof, wherein the two or more flash memory pages of a region can be read simultaneously, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region;

comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein the first gear has a different data payload size and correction capability than the second gear, wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear; and

selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds;

wherein determining, comparing, and selecting are performed by an integrated circuit.

2. (Original) The method of Claim 1, further comprising repeatedly determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds such that the bit error rate is actively monitored.

3. (Original) The method of Claim 1, further comprising determining the bit error rate and comparing the determined bit error rate to one or more predetermined thresholds at least once every 4 program/erase cycles.

4. (Previously presented) The method of Claim 1, wherein determining, comparing, and selecting are performed in connection with at least one of flash initialization, read, or write operations for an end user.

5. (Original) The method of Claim 1, further comprising reading data from the region of flash memory, wherein determining the bit error rate further comprises estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

6. (Original) The method of Claim 1, wherein selecting further comprises selecting the gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

7. (Original) The method of Claim 6, wherein the aUBER is 1E-16.

8. (Original) The method of Claim 1, wherein selecting further comprises selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

9. (Original) The method of Claim 1, further comprising:

receiving a command to perform formatting;

performing a memory test to determine raw bit error rates among the regions of the flash drive; and

allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

10. (Original) The method of Claim 1, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

11. (Original) The method of Claim 10, wherein the aUBER corresponds to about 1E-16 and the safety factor *SFx* is in a range from 1.5 to 4.

12. (Original) The method of Claim 10, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx = ECC\_CAPx / (Ex * RETx).$$

13. (Original) The method of Claim 12, wherein the aUBER corresponds to about 1E-16 and the excursion factor is in a range of about 1-4, and a retention factor is in a range of 1.2 to 2.2.

14. (Previously presented) The method of Claim 1, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

15. (Currently amended) An apparatus comprising:

a first circuit configured to determine a bit error rate associated with a region comprising a fixed number of two or more flash memory pages or integer fractions thereof, wherein the two or more flash memory pages of a region can be read simultaneously, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region and configured to compare the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different error correction coding (ECC) schemes, wherein the first gear has a different data payload size and correction capability than the second gear, wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear; and

a second circuit configured to select a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

16. (Previously presented) The apparatus of Claim 15, wherein the ECC encoder is configured to determine the bit error rate and the processor is configured to compare and to select in connection with at least one of flash initialization, read, or write operations for an end user.

17. (Original) The apparatus of Claim 15, wherein the first circuit is further configured to read data from the region of flash memory, wherein the ECC decoder is further configured to determine the bit error rate based at least partly on a number of errors corrected by ECC in connection with a read of data from the region.

18. (Original) The apparatus of Claim 15, wherein the processor is further configured to select a gear that has the highest data payload size and sufficient correction capability for the determined bit error rate such that memory reads from the region can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER).

19. (Original) The apparatus of Claim 15, wherein the processor is further configured to select a gear that has a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations performed to correct the requested user data.

20. (Original) The apparatus of Claim 15, wherein the processor is further configured to:

receive a command to perform formatting;

perform a memory test to determine raw bit error rates among the regions of the flash drive; and

allocate a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

21. (Original) The apparatus of Claim 15, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

22. (Previously presented) The apparatus of Claim 15, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein

the gear is selected such that performance of ECC decoding with the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

23. (Original) The apparatus of Claim 15, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

24. (Currently amended) An apparatus for selecting an error correction coding (ECC) scheme, the apparatus comprising:

a means for determining a bit error rate associated with a region comprising a fixed number of two or more flash memory pages or integer fractions thereof, wherein the two or more flash memory pages of a region can be read simultaneously, wherein the region stores at least data payload and primary and secondary ECC parity symbols corresponding to the data payload of the region;

a means for comparing the determined bit error rate to one or more predetermined thresholds corresponding to a set of predefined gears comprising at least a first gear and a second gear, wherein the predefined gears correspond to different predefined ECC schemes, wherein the first gear has a different data payload size and correction capability than the second gear, wherein the amount of memory space allocated for the storage of data payload within the region varies between the first gear and the second gear to accommodate a varying number of parity symbols between the first gear and the second gear; and

a means for selecting a gear from the set for the region based at least partly on the comparisons to the one or more predetermined thresholds.

25. (Original) The apparatus of Claim 24, further comprising means for reading data from the region of flash memory, wherein the determining means further comprises means for estimating the bit error rate based at least partly on a number of errors corrected by ECC in connection with reading the data from the region.

26. (Original) The apparatus of Claim 24, wherein the selecting means further comprises means for selecting a gear having a smaller data payload size and a higher amount of correction capability based at least partly a number of decoding iterations used to correct the requested user data.

27. (Original) The apparatus of Claim 24, further comprising:

means for receiving a command to perform formatting;

means for performing a memory test to determine raw bit error rates among the regions of the flash drive; and

means for allocating a region to one of at least the first gear or the second gear based at least partly on the raw bit error rate for the region.

28. (Original) The apparatus of Claim 24, wherein a threshold *Gx* of the one or more predetermined thresholds is less than or equal to a raw bit error rate *ECC_CAP* for which the corresponding strength of ECC provides sufficient error correction such that memory reads can be corrected with no more than a predetermined acceptable uncorrectable bit error rate (aUBER) divided by a safety factor *SFx*, wherein the safety factor *SFx* is a product of an excursion factor *Ex* and a retention factor *RETx*, such that the following expression holds:

$$Gx \leq ECC\_CAPx / SFx.$$

29. (Previously presented) The apparatus of Claim 24, wherein the gear includes a lower-latency ECC decode option and one or more higher-latency ECC decode options, wherein the gear is selected such that performing ECC decoding using the lower-latency ECC decode option alone is sufficient to provide corrected data at least 90% of the time.

30. (Original) The apparatus of Claim 24, further comprising a plurality of flash memory devices, wherein the apparatus comprises a flash drive.

31. (Currently amended) The method of Claim 1, wherein the data payload and corresponding primary ECC parity symbols comprise primary codewords, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region, wherein each of the two or more flash memory pages of a region reside on separate dies.

32. (Currently amended) The apparatus of Claim 15, wherein the data payload and corresponding primary ECC parity symbols comprise primary codewords, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across

the region, wherein each of the two or more flash memory pages of a region reside on separate dies.

33. (Currently amended) The apparatus of Claim 24, wherein the data payload and corresponding primary ECC parity symbols comprise primary codewords, wherein the data payload and corresponding secondary ECC parity symbols comprise secondary codewords, wherein the secondary ECC parity symbols of each secondary codeword are programmed across the region, wherein each of the two or more flash memory pages of a region reside on separate dies.

REMARKS

The February 2, 2015 Office Action rejects Claims 1-33 under 35 U.S.C. § 103(a). Applicant requests reconsideration of the rejections in view of the foregoing amendments and the following comments.

Rejections of Claims 31-33 under 35 U.S.C. § 112(b) or 112, second paragraph

The Office Action rejects Claims 31-33 for defining secondary codewords without defining primary codewords. Applicant has amended Claims 31-33 to recite primary codewords and respectfully request the Examiner to withdraw the rejections under 35 U.S.C. § 112(b) or 112, second paragraph.

In addition, Applicant has amended Claims 31-33 to add "wherein each of the two or more flash memory pages of a region reside on separate dies." These amendments are supported by, for example, paragraph **[0081]** of the specification as originally filed.

Rejections of Claims 1, 15, and 24 under 35 U.S.C. § 103(a)

The Office Action rejects Claims 1, 15, and 24 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2009/0070651 by Diggs ("Diggs") in view of U.S. Patent Application Publication No. 2013/0246891 by Manning ("Manning") further in view of U.S. Patent No. 8,433,981 by Agarwal ("Agarwal"). Applicant has amended Claims 1, 15, and 24 to clarify distinctions over the cited references.

In rejecting Claims 1, 15, and 24, the Office Action interprets "region" as reading on both Manning's codewords and Agarwal's blocks. Applicant has clarified that "the two or more flash memory pages of a region can be read simultaneously," which is supported by, for example, paragraph **[0081]** of the specification as originally filed.

Manning does not teach or disclose that a region can comprise a fixed number of two or more flash memory pages or integer fractions thereof. Further, a block as taught by Agarwal corresponds to a region that can be erased rather than to a region that can be read simultaneously. Thus, the "region" as recited in the claims is not taught or disclosed in the cited references. Applicant further respectfully submits that it is not reasonable to interpret "region" in more than one way in rejecting a claim. While Applicant acknowledges that the Examiner is permitted to interpret claim language broadly, such interpretation must nonetheless still be reasonable. See M.P.E.P. 2111.

The Combination of References Does Not Teach All Elements

Host computers transfer data in fixed sizes (called allocation blocks or sectors). Once it is determined what the size of the allocation block is, the size does not change. Thus, when the data payload size changes within a region, this presents a difficult problem to handle. Rather than reallocation, in background paragraph **[0006]**, Diggs states:

> **[0006]** The ability of the storage subsystem to detect and correct errors can be increased by increasing the quantity of ECC bits stored per unit of data. However, such increases in the quantity of ECC used generally decreases the performance of the storage subsystem and may require additional (i.e., off-sector) storage space when the quantity of ECC data is larger than the amount of management data provided for each sector.

Applicant respectfully submits that the combination of Diggs, Manning, and Agarwal does not teach all the claimed elements, in particular, the recited allocation of memory space. As stated in M.P.E.P. at § 2143.02, a "rationale to support a conclusion that a claim would have been obvious is that ***all the claimed elements were known in the prior art*** and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded nothing more than predictable results to one of ordinary skill in the art." M.P.E.P. § 2143.02; *see KSR v. Teleflex*, 550 U.S. at 398, 82 USPQ2d at 1395.

Proceeding Contrary to Accepted Wisdom

In addition, Applicant respectfully submits that to perform the dynamic reallocation of memory space within the region is proceeding contrary to accepted wisdom, which is evidence of nonobviousness. The totality of the prior art must be considered, and proceeding contrary to accepted wisdom in the art is evidence of nonobviousness. *In re Hedges*, 783 F.2d 1038, 228 USPQ 685 (Fed. Cir. 1986) "The totality of the prior art must be considered, and proceeding contrary to accepted wisdom in the art is evidence of nonobviousness." *In re Hedges*, 783 F.2d 1038, 228 USPQ 685 (Fed. Cir. 1986), M.P.E.P. § 2145(D)(3).

Accordingly, Applicant requests the Examiner to withdraw the rejections of Claims 1, 15, and 24, and to allow the same.

Rejection of Dependent Claims 2-14, 16-23, and 25-33

Dependent Claims 2-14 and 31, 16-23, and 32, and 25-30 and 33 depend from and further define Claims 1, 15, and 24, respectively. The dependent claims recite numerous additional distinctions over the cited references.

With respect to Claims 31-33, Applicant respectfully submits that if the term "region" is interpreted to read on a codeword, then the recitation of "parity symbols of each secondary codeword are programmed across the region" would not make sense and thus could not be reasonable.

Claims 2-14, 16-23, and 25-33 add additional limitations that Applicant believes are not taught or suggested by the prior art. In addition, Applicant respectfully submits that dependent Claims 2-14, 16-23, and 25-33 are patentable for at least the reasons described for Claims 1, 15, and 24, respectively, and Applicant accordingly requests allowance of Claims 2-14, 16-23, and 25-33.

No Disclaimers or Disavowals

Although the present communication may include alterations to the application or claims, or characterizations of claim scope or referenced art, Applicant is not conceding in this application that previously pending claims are not patentable over the cited references. Rather, any alterations or characterizations are being made to facilitate expeditious prosecution of this application. Applicant reserves the right to pursue at a later date any previously pending or other broader or narrower claims that capture any subject matter supported by the present disclosure, including subject matter found to be specifically disclaimed herein or by any prior prosecution. Accordingly, reviewers of this or any parent, child or related prosecution history shall not reasonably infer that Applicant has made any disclaimers or disavowals of any subject matter supported by the present application.

Co-Pending Applications of Assignee

Applicant wishes to draw the Examiner's attention to the following co-pending applications of the present application's assignee.

Docket No.	Serial No. / Patent No.	Title (as filed)	Filed

Application No.: 13/477,600
Filing Date: May 22, 2012

Docket No.	Serial No. / Patent No.	Title (as filed)	Filed
DATUMTE.076A2	13/477,633 8,996,957	SYSTEMS AND METHODS FOR INITIALIZING REGIONS OF A FLASH DRIVE HAVING DIVERSE ERROR CORRECTION CODING (ECC) SCHEMES	May 22, 2012
DATUMTE.076A3	13/477,629 9,021,337	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING AMONG DIFFERENT ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A4	13/477,598 9,021,336	SYSTEMS AND METHODS FOR REDUNDANTLY STORING ERROR CORRECTION CODES IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A5	13/477,595 8,788,910	SYSTEMS AND METHODS FOR LOW LATENCY, HIGH RELIABILITY ERROR CORRECTION IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A6	13/477,568	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A7	13/477,599 8,972,824	SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	May 22, 2012
DATUMTE.076A8	13/477,845 8,793,556	SYSTEMS AND METHODS FOR RECLAIMING FLASH BLOCKS OF A FLASH DRIVE	May 22, 2012
DATUMTE.076A9	13/477,635 9,047,214	SYSTEMS AND METHODS FOR REDUNDANTLY STORING DATA IN A FLASH DRIVE SUCH THAT A FAILED PAGE CAN BE TOLERATED	May 22, 2012
DATUMTE.076A10	13/477,601 9,021,333	SYSTEMS AND METHODS FOR RECOVERING DATA FROM FAILED PORTIONS OF A FLASH DRIVE	May 22, 2012

Please charge any additional fees, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: June 2, 2015

By: _____

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

20788700
060215

INFORMATION DISCLOSURE STATEMENT

Inventor	:	Philip L. Northcott
App. No.	:	13/477600
Filed	:	May 22, 2012
For	:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY
Examiner	:	Dorman, Christian M.
Art Unit	:	2112
Conf. No.	:	2672

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

References and Listing

Pursuant to 37 CFR 1.56, an Information Disclosure Statement listing references is provided herewith. Copies of any listed foreign and non-patent literature references are being submitted.

No Disclaimers

To the extent that anything in the Information Disclosure Statement or the listed references could be construed as a disclaimer of any subject matter supported by the present application, Applicant hereby rescinds and retracts such disclaimer.

Timing of Disclosure

This Information Disclosure Statement is being filed after receipt of a First Office Action, but before the mailing date of a Final Action and before the mailing date of a Notice of Allowance. This Statement is accompanied by the fees set forth in 37 CFR 1.17(p). The

Application No.: **13/477600**
Filing Date: **May 22, 2012**

Commissioner is hereby authorized to charge any additional fees which may be required or to credit any overpayment to Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: June 2, 2015

By: _____

Michael S. Okamoto
Registration No. 47,831
Attorney of Record
Customer No. 20995
(310) 551-3450

20809747
060215

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 1	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	8,788,910	07-22-2014	Northcott, Philip L.	
	2	8,793,556	07-29-2014	Northcott, et al.	
	3	2010/0142756 A1	06-10-2010	Fields, et al.	
	4	2013/0031563 A1	01-31-2013	Skowron, et al.	
	5	2014/0136927 A1	05-15-2014	Li, et al.	

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T^1
	6	USPTO; Office Action dated December 22, 2014, from related application U.S. Patent Application No. 13/844474 (filed March 15, 2013).	
	7	USPTO; Office Action dated February 27, 2015, from related application U.S. Patent Application No. 13/477,568 (filed May 22, 2012).	
	8	USPTO; Office Action dated April 10, 2015, from related application U.S. Patent Application No. 13/844,486 (filed March 15, 2013).	

20809095
060215

Examiner Signature	Date Considered
***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

T^1 - Place a check mark in this area when an English language Translation is attached.

Electronic Patent Application Fee Transmittal

Application Number:	13477600
Filing Date:	22-May-2012
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Filer:	Michael S. Okamoto/Kenneth Watanabe
Attorney Docket Number:	DATUMTE.076A1

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension - 1 month with $0 paid	1251	1	200	200
Miscellaneous:				
		Total in USD ($)		**200**

Electronic Acknowledgement Receipt

EFS ID:	22517013
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/Jennifer Neat
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	02-JUN-2015
Filing Date:	22-MAY-2012
Time Stamp:	19:38:03
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$200
RAM confirmation Number	6366
Deposit Account	111410
Authorized User	KNOBBE MARTENS OLSON AND BEAR

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		20150602_Amendment_DATUMTE076A1.pdf	598178 be1643ca127223f6ef6e9c3da2775388b6d94eee	yes	13

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	8
Applicant Arguments/Remarks Made in an Amendment	9	13

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
2		20150602_IDS_DATUMTE076A1.pdf	96540 9ac83160197ae79a69b25ddd1238eed63ee199ac	yes	3

Multipart Description/PDF files in .zip description

Document Description	Start	End
Transmittal Letter	1	2
Information Disclosure Statement (IDS) Form (SB08)	3	3

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
3	Non Patent Literature	NPL1.PDF	925971 22f53e3803cbceb759836f3c94e997eb010e6193	no	18

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
4	Non Patent Literature	NPL2.PDF	1112417 471872d0f157cbf481a00cf880cfd8f2782485578	no	23

Warnings:

Information:

5	Non Patent Literature	NPL3.PDF	1098260 9240c5c3a293eaec5d8a084fbb26452ca24a476e	no	20
Warnings:					
Information:					
6	Fee Worksheet (SB06)	fee-info.pdf	31065 36d7ad05eaa1616b60ae8b9d2832382462 3f1993	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3862431		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 13/477,600	Filing Date 05/22/2012	☐ To be Mailed

ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)		
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	06/02/2015	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 33	Minus	** 33	= 0	X $80 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $420 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
/ANNIE SINGLETON/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

20995 7590 07/06/2015

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER
DORMAN, CHRISTIAN M

ART UNIT	PAPER NUMBER
2112	

DATE MAILED: 07/06/2015

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

TITLE OF INVENTION: SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$960	$0	$0	$960	10/06/2015

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 02/11)

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

20995 7590 07/06/2015
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

TITLE OF INVENTION: SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMOUNG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$960	$0	$0	$960	10/06/2015

EXAMINER	ART UNIT	CLASS-SUBCLASS
DORMAN, CHRISTIAN M	2112	714-773000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): (**Please first reapply any previously paid issue fee shown above**)
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ Applicant certifying micro entity status. See 37 CFR 1.29
  NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
- ❑ Applicant asserting small entity status. See 37 CFR 1.27
  NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
- ❑ Applicant changing to regular undiscounted fee status.
  NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Ex.1002 / Page 716 of 744
Sandisk Technologies, Inc. et al

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

20995 7590 07/06/2015

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
DORMAN, CHRISTIAN M	
ART UNIT	**PAPER NUMBER**
2112	

DATE MAILED: 07/06/2015

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 13/477,600	Applicant(s) NORTHCOTT, PHILIP L.	
	Examiner CHRISTIAN DORMAN	Art Unit 2112	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the communication received 6/02/2015*.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on_____.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are *1-33*. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov .
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some *c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 6/02/2015
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
5. ☐ Examiner's Amendment/Comment
6. ☐ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/CHRISTIAN DORMAN/ Examiner, Art Unit 2112	/ALBERT DECADY/ Supervisory Patent Examiner, Art Unit 2112

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	2112

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/19/2013	06/16/2014	01/24/2015	06/25/2015					
	1	✓	✓	✓	=					
	2	✓	✓	✓	=					
	3	✓	✓	✓	=					
	4	✓	✓	✓	=					
	5	✓	✓	✓	=					
	6	✓	✓	✓	=					
	7	✓	✓	✓	=					
	8	✓	✓	✓	=					
	9	✓	✓	✓	=					
	10	✓	✓	✓	=					
	11	✓	✓	✓	=					
	12	✓	✓	✓	=					
	13	✓	✓	✓	=					
	14	✓	✓	✓	=					
	15	✓	✓	✓	=					
	16	✓	✓	✓	=					
	17	✓	✓	✓	=					
	18	✓	✓	✓	=					
	19	✓	✓	✓	=					
	20	✓	✓	✓	=					
	21	✓	✓	✓	=					
	22	✓	✓	✓	=					
	23	✓	✓	✓	=					
	24	✓	✓	✓	=					
	25	✓	✓	✓	=					
	26	✓	✓	✓	=					
	27	✓	✓	✓	=					
	28	✓	✓	✓	=					
	29	✓	✓	✓	=					
	30	✓	✓	✓	=					
	31			✓	=					
	32			✓	=					
	33			✓	=					

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	**Art Unit**
	CHRISTIAN DORMAN	2112

CPC							
Symbol						**Type**	**Version**
G06F		11	/	1068		F	2013-01-01
G06F		11	/	1044		A	2013-01-01
G06F		11	/	1008		A	2013-01-01
G06F		11	/	1048		I	2013-01-01
			/				
			/				
			/				
			/				
			/				
			/				
			/				
			/				
			/				
			/				
			/				

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
			/					
			/					

/CHRISTIAN DORMAN/ Examiner.Art Unit 2112	06/25/2015	**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	33	
/ALBERT DECADY/ Supervisory Patent Examiner.Art Unit 2112	06/29/2015	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	6

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	**Art Unit**
	CHRISTIAN DORMAN	2112

US ORIGINAL CLASSIFICATION		INTERNATIONAL CLASSIFICATION	
CLASS	SUBCLASS	CLAIMED	NON-CLAIMED
714	773	G 0 6 F 11 / 08 (2006.0)	
		G 0 6 F 11 / 10 (2006.0)	

CROSS REFERENCE(S)					
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
714	774	779			

/CHRISTIAN DORMAN/ Examiner.Art Unit 2112	06/25/2015	**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	33	
/ALBERT DECADY/ Supervisory Patent Examiner.Art Unit 2112	06/29/2015	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	6

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	**Art Unit**
	CHRISTIAN DORMAN	2112

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	17	17	33	33										
2	2	18	18												
3	3	19	19												
4	4	20	20												
5	5	21	21												
6	6	22	22												
7	7	23	23												
8	8	24	24												
9	9	25	25												
10	10	26	26												
11	11	27	27												
12	12	28	28												
13	13	29	29												
14	14	30	30												
15	15	31	31												
16	16	32	32												

/CHRISTIAN DORMAN/ Examiner.Art Unit 2112	06/25/2015	**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	33	
/ALBERT DECADY/ Supervisory Patent Examiner.Art Unit 2112	06/29/2015	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	6

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13477600	NORTHCOTT, PHILIP L.
	Examiner	Art Unit
	CHRISTIAN DORMAN	2112

CPC- SEARCHED		
Symbol	Date	Examiner
G06F11/1004,1008,1044,1048,1068,1072	1/12,20,24/2015, 6/25/2015	CD

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
714	773	12/16/2013, 12/18/2013, 12/19/2013, 6/02/2014	CD
714	763,769,773,774,779	1/12,20,24/2015, 6/25/2015	CD

SEARCH NOTES		
Search Notes	Date	Examiner
EAST US_PGPUB, USPAT, USOCR, FPRS, EPO, JPO, DERWENT, IBM_TDB - search results are attached	12/16,18,19/2013, 6/02/2014, 6/03/2014, 6/05/2014, 1/12,20,24/2015, 6/25/2015	CD
Inventor Name Search, PALM	12/16/2013, 12/19/2013	CD
Double Patenting Search, PALM	12/16/2013, 12/19/2013	CD
IEEE NPL Search	12/18/2013	CD
Consulted with Primary Examiner Samir Rizk	1/22/2015	CD

INTERFERENCE SEARCH			
US Class/ CPC Symbol	**US Subclass / CPC Group**	**Date**	**Examiner**
	Search for Interference	6/25/2015	CD

EAST Search History

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S19	8	memory with FLASH with ("ECC" (error adj correct$4)) with page with primary with secondary with payload	US-PGPUB; USPAT; UPAD	OR	ON	2015/06/25 11:10
S20	8	(FLASH memory) with (data adj payload) with ("ECC" (error adj correct$4)) with page with select$3	US-PGPUB; USPAT; UPAD	OR	ON	2015/06/25 11:12
S21	8	gear with ("ECC" (error adj correct$3)) with select$4 with region with threshold	US-PGPUB; USPAT; UPAD	OR	ON	2015/06/25 11:12
S22	8	gear with ("ECC" (error adj correct$3)) with select$4 with region	US-PGPUB; USPAT; UPAD	OR	ON	2015/06/25 11:13

6/ 25/ 2015 11:13:42 AM
C:\ Users\ cdorman\ Documents\ EAST\ Workspaces\ 13477600_RCE.wsp

Search for Interference
6/25/2015
/CMD/

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	61	("20110038203" | "20110072196" | "20120110417" | "20120311406" | "20130060565" | "20130173955" | "6279830" | "7415651" | "7545689" | "8255620" | "8402217" | "8464095" | "8560881" | "8689082" | "8713245" | "8713411" | "8725944" | "8750042").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:35
S2	25071	G06F11/1004.cpc. G06F11/1008.cpc. G06F11/1044.cpc. G06F11/1048.cpc. G06F11/1068.cpc. G06F11/1072.cpc. 714/763.ccls. 714/773.ccls. 714/774.ccls. 714/746.ccls. 714/752.ccls. 714/769.ccls. 714/779.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:37
S3	17	("20090070651" | "20090327840" | "20110072196" | "20110213920" | "20130173955" | "20130246891" | "8612680").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:40
S4	12	("20090070651" | "20130080862" | "20130179754" | "20130246891" | "8495465").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:40
S5	2	S4 and payload	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 19:44
S6	30	fixed adj number with page with flash	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:30
S7	25116	G06F11/1004.cpc. G06F11/1008.cpc. G06F11/1044.cpc. G06F11/1048.cpc. G06F11/1068.cpc. G06F11/1072.cpc. 714/763.ccls. 714/773.ccls. 714/774.ccls. 714/746.ccls. 714/752.ccls. 714/769.ccls. 714/779.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:30

S8	13	S6 and S7	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:30
S9	7	"20090070651".pn. "20130179754".pn. "20130246891".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 18:32
S10	2	"8560881".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/24 19:15
S11	12	("20100142756" | "20130031563" | "20140136927" | "8788910" | "8793556").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:10
S12	18	("20120110417" | "8433981").PN. ("20090070651" | "20130080862" | "20130179754" | "20130246891" | "8495465").PN.("20090070651" | "20130080862" | "20130179754" | "20130246891").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:10
S13	11	"13/477,600"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:14
S14	26635	G06F11/1004.cpc. G06F11/1008.cpc. G06F11/1044.cpc. G06F11/1048.cpc. G06F11/1068.cpc. G06F11/1072.cpc. 714/763.ccls. 714/773.ccls. 714/774.ccls. 714/746.ccls. 714/752.ccls. 714/769.ccls. 714/779.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:35
S15	141	(data near2 payload) with (chang$4 different modify$3) same ("ECC" (error adj correct$4)) with (different chang$4 modify$4 select$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:42
S16	26635	S14 and S14	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2015/06/25 10:47

			DERWENT; IBM_TDB			
S17	19	S14 and S15	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:47
S18	58	(data near2 payload) with (chang$4 different modify$3) same ("ECC" (error adj correct$4)) with (different chang$4 modify$4 select$4) same2 (memory FLASH)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 10:55

6/ 25/ 2015 11:13:33 AM
C:\ Users\ cdorman\ Documents\ EAST\ Workspaces\ 13477600_RCE.wsp

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 1	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	8,788,910	07-22-2014	Northcott, Philip L.	
	2	8,793,556	07-29-2014	Northcott, et al.	
	3	2010/0142756 A1	06-10-2010	Fields, et al.	
	4	2013/0031563 A1	01-31-2013	Skowron, et al.	
	5	2014/0136927 A1	05-15-2014	Li, et al.	

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	6	USPTO; Office Action dated December 22, 2014, from related application U.S. Patent Application No. 13/844474 (filed March 15, 2013).	
	7	USPTO; Office Action dated February 27, 2015, from related application U.S. Patent Application No. 13/477,568 (filed May 22, 2012).	
	8	USPTO; Office Action dated April 10, 2015, from related application U.S. Patent Application No. 13/844,486 (filed March 15, 2013).	

20809095
060215

Examiner Signature	/CHRISTIAN DORMAN/	Date Considered	06/25/2015

*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /CD/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/477,600	05/22/2012	2112	2220	DATUMTE.076A1	30	3

CONFIRMATION NO. 2672

CORRECTED FILING RECEIPT

20995
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

OC000000078311434

Date Mailed: 07/16/2015

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Philip L. Northcott, Coquitlam, CANADA;

Applicant(s)

Philip L. Northcott, Coquitlam, CANADA;

Assignment For Published Patent Application

PMC-Sierra, Inc., Santa Clara, CA

Power of Attorney: The patent practitioners associated with Customer Number 20995

Domestic Applications for which benefit is claimed - None.
A proper domestic benefit claim must be provided in an Application Data Sheet in order to constitute a claim for domestic benefit. See 37 CFR 1.76 and 1.78.

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

If Required, Foreign Filing License Granted: 06/01/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/477,600**

Projected Publication Date: Request for Non-Publication Acknowledged

Non-Publication Request: Yes

Early Publication Request: No

Title

SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY

Preliminary Class

714

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications:

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax **(571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

20695 7093 07/06/2015
KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	05/22/2012	Philip L. Northcott	DATUMTE.076A1	2672

TITLE OF INVENTION: SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FRO ROBUSTNESS AND LOW LATENCY

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEES DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$960	$0	$0	$960	10/06/2015

EXAMINER	ART UNIT	CLASS-SUBCLASS
DOBMAN, CHRISTIAN M	2112	714-773000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Knobbe Martens
Olson & Bear LLP
2
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: PMC-Sierra, Inc.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Sunnyvale, California

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☒ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies .....

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The director is hereby authorized to charge ~~the required fee(s),~~ any deficiency, or credits any overpayment, to Deposit Account Number 11-1410 (enclose an extra copy of this form).

5 **Change in Entity Status** (from status indicated above)
- ☐ Applicant certifying micro entity status. See 37 CFR 1.29 — NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
- ☐ Applicant asserting small entity status. See 37 CFR 1.27 — NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
- ☐ Applicant changing to regular undiscounted fee status. — NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature Michael Okamoto Date 10/06/2015

Typed or printed name Michael S. Okamoto Registration No. 47,531

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Patent Application Fee Transmittal

Application Number:	13477600
Filing Date:	22-May-2012
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Filer:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl Issue Fee	1501	1	960	960

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**960**

Electronic Acknowledgement Receipt

EFS ID:	23708280
Application Number:	13477600
International Application Number:	
Confirmation Number:	2672
Title of Invention:	SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING FROM AMONG A PLURALITY OF ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE FOR ROBUSTNESS AND LOW LATENCY
First Named Inventor/Applicant Name:	Philip L. Northcott
Customer Number:	20995
Filer:	Michael S. Okamoto/Tony Do
Filer Authorized By:	Michael S. Okamoto
Attorney Docket Number:	DATUMTE.076A1
Receipt Date:	06-OCT-2015
Filing Date:	22-MAY-2012
Time Stamp:	17:14:32
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$960
RAM confirmation Number	3951
Deposit Account	111410
Authorized User	KNOBBE MARTENS OLSON AND BEAR

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	IF_Transmittal_DATUMTE076A1.pdf	819952 1d3b1272772e1a587d3098225103aacf379f93f3	no	1
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30866 2c34f15fb61a14297aca908a801e0c98ba34049c	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	850818		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Change(s) applied to document, /V.D.W./ 7/29/2015

Title	Application No.	Attorney Docket No.
SYSTEMS AND METHODS FOR RECLAIMING FLASH BLOCKS OF A FLASH DRIVE	13/477845	DATUMTE.076A8
SYSTEMS AND METHODS FOR REDUNDANTLY STORING DATA IN A FLASH DRIVE SUCH THAT A FAILED PAGE CAN BE TOLERATED	13/477635	DATUMTE.076A9
SYSTEMS AND METHODS FOR RECOVERING DATA FROM FAILED PORTIONS OF A FLASH DRIVE	13/477601	DATUMTE.076A10

[maxlife]
13316155_1
052112

Appendix A

Incorporation by Reference of Commonly Owned Applications

The following patent applications, commonly owned and filed on the same day as the present application, are hereby incorporated herein in their entirety by reference thereto:

Change(s) applied to document, /N.D.W./ 7/29/2015

Title	Application No.	Attorney Docket No.
SYSTEMS AND METHODS FOR INITIALIZING REGIONS OF A FLASH DRIVE HAVING DIVERSE ERROR CORRECTION CODING (ECC) SCHEMES	13/477633	DATUMTE.076A2
SYSTEMS AND METHODS FOR ADAPTIVELY SELECTING AMONG DIFFERENT ERROR CORRECTION CODING SCHEMES IN A FLASH DRIVE	13/477629	DATUMTE.076A3
SYSTEMS AND METHODS FOR REDUNDANTLY STORING ERROR CORRECTION CODES IN A FLASH DRIVE	13/477598	DATUMTE.076A4
SYSTEMS AND METHODS FOR LOW LATENCY, HIGH RELIABILITY ERROR CORRECTION IN A FLASH DRIVE	13/477595	DATUMTE.076A5
SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	13/477568	DATUMTE.076A6
SYSTEMS AND METHODS FOR TRANSPARENTLY VARYING ERROR CORRECTION CODE STRENGTH IN A FLASH DRIVE	13/477599	DATUMTE.076A7

PTO/SB/08 Equivalent

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application No.	13/477600
	Filing Date	May 22, 2012
	First Named Inventor	Philip L. Northcott
	Art Unit	2112
(Multiple sheets used when necessary)	Examiner	Dorman, Christian M.
SHEET 1 OF 2	Attorney Docket No.	DATUMTE.076A1

U.S. PATENT DOCUMENTS

Examiner Initials	Cite No.	Document Number *Number - Kind Code (if known)* Example: 1,234,567 B1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear
	1	6,279,830	08-28-2001	Ishibashi, Toshiharu	
	2	7,415,651	08-19-2008	Argon, Cenk	
	3	7,545,689	06-09-2009	Ilkbahar, et al.	
	4	8,255,620	08-25-2012	Frost, et al.	
	5	8,402,217	03-19-2013	Burd, Gregory	
	6	8,464,095	06-11-2013	Bonwick, Jeffrey S.	
	7	8,560,881	10-15-2013	Frost, et al.	
	8	8,689,082	04-01-2014	Oh, et al.	
	9	8,713,245	04-29-2014	Frost, et al.	
	10	8,713,411	04-29-2014	Kong, et al.	
	11	8,725,944	~~03-13-2014~~ 5-2014	Burd, Gregory	
	12	8,750,042	06-10-2014	Sharon, et al.	
	13	2011/0038203	02-17-2001	Camp, et al.	
	14	2011/0072196	03-24-2011	Forhan, et al.	
	15	2012/0110417	05-03-2012	D'Abreu, et al.	
	16	2012/0311406	12-06-2012	Ratnam, et al.	
	17	2013/0060565	03-07-2013	Nair, et al.	
	18	2013/0173955	07-04-2013	Hallak, et al.	

Change(s) applied to document, /V.D.W./ 8/6/2015

FOREIGN PATENT DOCUMENTS

Examiner Initials	Cite No.	Foreign Patent Document *Country Code-Number-Kind Code* Example: JP 1234567 A1	Publication Date MM-DD-YYYY	Name of Patentee or Applicant	Pages, Columns, Lines Where Relevant Passages or Relevant Figures Appear	T[1]

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[1]
	19	USPTO; Office Action dated May 27, 2014, from related application U.S. Patent Application No. 13/477,629 (filed May 22, 2012).	
	20	USPTO; Office Action dated June 5, 2014, from related application U.S. Patent Application No. 13/477,598 (filed May 22, 2012).	

Examiner Signature	Date Considered

***Examiner:** Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

T[1] - Place a check mark in this area when an English language Translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.D./

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/477,600	11/10/2015	9183085	DATUMTE.076A1	2672

20995 7590 10/21/2015

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 66 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Philip L. Northcott, Coquitlam, CANADA;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	REQUEST ID
13/477,600	9183085	2112	55116

PAIR Correspondence Address/Fee Address Change

The following fields have been changed to Customer Number 197 on 01/15/2018 via Private PAIR in view of the certification copied below that authorized the change.

- Maintenance Fee Address

The address for Customer Number 197 is:
197
CPA GLOBAL LIMITED
2318 Mill Road 12th Floor
ALEXANDRIA, VA 22314

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

An attorney or Agent of Record registered to practice before the Patent and Trademark Office who has been given power of attorney in this application

Signature:	/David N. Weiss/
Name:	David N. Weiss
Registration Number:	41371

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	REQUEST ID
13/477,600	9183085	2112	56057

PAIR Correspondence Address/Fee Address Change

The following fields have been changed to Customer Number 197 on 01/25/2018 via Private PAIR in view of the certification copied below that authorized the change.

- Maintenance Fee Address

The address for Customer Number 197 is:
197
CPA GLOBAL LIMITED
2318 Mill Road 12th Floor
ALEXANDRIA, VA 22314

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

An attorney or Agent of Record registered to practice before the Patent and Trademark Office who has been given power of attorney in this application

Signature:	/David N. Weiss/
Name:	David N. Weiss
Registration Number:	41371