UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

In re Patent Of:	Digital Cache, LLC
U.S. Patent No.:	6,851,015
Issue Date:	February 1, 2005
Appl. Serial No.:	10/152,206
U.S. Filing Date:	May 20, 2002
Title:	Method of Overwriting Data in Nonvolatile Memory and a Control Apparatus Used for the Method

---

IPR2022-00121

PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT NO. 6,851,015 UNDER 35 U.S.C. §312 AND 37 C.F.R. §42.104

I. TABLE OF CONTENTS

I. INTRODUCTION 1
II. MANDATORY NOTICES 2
A. Real Party-in-Interest 2
B. Related Matters 3
C. Lead and Back-up Counsel and Service Information 4
III. GROUNDS FOR STANDING 5
IV. SUMMARY OF '015 PATENT AND PROSECUTION HISTORY 5
A. Summary of the '015 Patent 5
B. Prosecution History of the '015 Patent 10
V. LEVEL OF ORDINARY SKILL IN THE ART 11
VI. CLAIM CONSTRUCTION 12
VII. REQUESTED RELIEF 12
VIII. IDENTIFICATION OF CHALLENGE 12
A. Challenged Claims and Statutory Grounds 12
B. Status as Prior Art 13
IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE 13
A. Ground #1: Claims 1, 2 and 6-11 are obvious over Morihiro 13
1. Summary of Morihiro 13
2. Claim 1 23
3. Claim 2 44
4. Claim 6 45
5. Claim 7 46
6. Claim 8 48
7. Claim 9 49
8. Claim 10 50
9. Claim 11 52

B. Ground #2: Claims 3-5 are unpatentable under 35 U.S.C. §103 as being obvious over Morihiro and Forsman. 56
1. Summary of Forsman 56
2. Reasons to combine Morihiro and Forsman 57
3. Claim 3 60
4. Claim 4 64
5. Claim 5 66
C. Ground #3: Claim 11 is unpatentable under 35 U.S.C. §103 as being obvious over Morihiro and Mitomi. 70
1. Summary of Mitomi 70
2. Reasons to combine Morihiro and Mitomi. 76
3. Claim 11 78
X. THE *FINTIV* FACTORS FAVOR INSTITUTION 87
1. Factor 1 is neutral (possibility of a stay) 87
2. Factor 2 favors institution (proximity of trial date to final written decision) 88
3. Factor 3 favors institution (investment in parallel proceeding). 88
4. Factor 4 favors institution (overlap in issues) 89
5. Factor 5 is neutral (overlap in parties). 90
6. Factor 6 favors institution (other circumstances). 90
XI. CONCLUSION 91
XII. CERTIFICATE OF WORD COUNT 92

PETITIONER'S EXHIBIT LIST

EX1001	U.S. Patent No. 6,851,015 ("'015 Patent")
EX1002	U.S. Prosecution History of the '015 Patent, ("'015 Prosecution History")
EX1003	Notice of Reasons for Refusal - Excerpt from Japanese Prosecution History of Japanese Patent Publication No. JP2001153139A
EX1004	U.S. Patent No. 6,834,331 to Liu ("Liu")
EX1005	English Translation of Japanese Patent Publication No. JP-H10124403 to Mitomi *et al*. ("Mitomi")
EX1006	English Translation of Japanese Patent Publication No. JP-H7168769 to Morihiro ("Morihiro")
EX1007	U.S. Patent No. 6,665,813 to Forsman *et al*. ("Forsman")
EX1008	U.S. Patent No. 6,802,038 to Yu ("Yu")
EX1009	U.S. Patent No. 6,934,904 to Talagala *et al*. ("Talagala")
EX1010	CV of Dr. R. Jacob Baker
EX1011	Declaration of Dr. R. Jacob Baker ("Baker Dec.")
EX1012	Docket Sheet
EX1013	Japanese Patent Publication No. JP-H10124403 to Mitomi *et al*.
EX1014	Declaration of Herman Kahn certifying translation of Japanese Patent Publication No. JP-H10124403 to Mitomi *et al.*
EX1015	Japanese Patent Publication No. JP-H7168769 to Morihiro
EX1016	Declaration of Herman Kahn certifying translation of Japanese Patent Publication No. JP-H7168769 to Morihiro

Note that except for patents and the Baker declaration, the following analysis will cite to our bates-labeled page numbers of the exhibits, as opposed to

the page numbers provided within the exhibit. Also, the cites to foreign applications are to the certified translations of the various documents. Additionally, the following analysis may bold, underline and/or italicize quotations and add color or annotations to the figures from these exhibits for the sake of emphasis, unless otherwise indicated.

I. INTRODUCTION

The '015 Patent is directed to non-volatile[1] memory (NVM) devices, which are capable of retaining their stored data in case of a power failure. NVM devices, such as flash memory, can have a multistep process for writing or updating data in the device, such as is shown in prior art FIG. 4 of the '015 patent. In summary, to write data to flash memory, a data area referred to as a "sector" must first be erased before data can be written therein (generally referred to as "overwriting"). It was a known concern of flash memory that if there was a power outage during the overwrite process, the data to be written could be lost. The solution of the '015 Patent was to write data to be overwritten to a backup region in NVM during the overwriting process.

However, this solution was known in the art. For example, Morihiro is directed "[t]o prevent[ing] data from being lost and becoming unrecoverable ***due to electric service interruption***, etc. in cases where data stored on . . . nonvolatile memory is being updated." EX1006, Abstract (emphasis added). Specifically, Morihiro discloses a flash memory that includes data sectors and a backup region – "data areas 10 . . . are formed in the flash memory 4" along with "a working area

[1] Terms "non-volatile" and "nonvolatile" are used interchangeably in the petition like the '015 Patent. *See* EX1001, Claims 1-11; *see also* EX1011, [0035].

11 for temporarily storing data during data updating is provided within this flash memory 4." EX1006, [0042]. Morihiro discloses that "when a data update request is issued, ***the data to be updated is copied first into the working area 11***, after which updating of the data of the target data area 10 is performed." EX1006, [0053] (emphasis added). Thus, Morihiro discloses writing data to be overwritten to a backup region (i.e. working area 11) in flash memory during an overwriting process. *See* EX1006, [0012]; Baker Dec. (EX1011), [0058]-[0072].

Accordingly, as further explained below, the problem (i.e. data corruption caused by power failure during erasing or writing) and the solution (i.e. copying pre-overwrite data to a backup region in flash memory) were well known in the art prior to the '015 patent. *See* EX1011, [0045]-[0050] and [0057]-[0072]. Thus, Petitioner respectfully requests claims 1-11 be canceled.

II. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. §42.8(b)(1), Pure Storage, Inc. is the real party-in-interest and is designated as Petitioner. No other parties have directed, funded, or controlled the filing of this IPR, and this IPR was not filed at the behest of any other party.

B. Related Matters

Pursuant to 37 C.F.R. §42.8(b)(2), to the best knowledge of the Petitioner, the '015 Patent is or was involved in the following cases ("Related Litigation"):

Case Heading	Number	Court	Filed
Digital Cache, LLC v. Patriot Memory, LLC	1:21-cv-01166	DDE	Aug. 12, 2021
Digital Cache, LLC v. Greenliant Systems, Inc.	1:21-cv-01167	DDE	Aug. 12, 2021
Digital Cache, LLC v. Netlist, Inc.	1:21-cv-01168	DDE	Aug. 12, 2021
Digital Cache, LLC v. PNY Technologies, Inc.	1:21-cv-01169	DDE	Aug. 12, 2021
Digital Cache, LLC v. Storcentric, Inc.	1:21-cv-01170	DDE	Aug. 12, 2021
Digital Cache, LLC v. LTL Group, Inc.	4:21-cv-02592	SDTX	Aug. 10, 2021
Digital Cache, LLC v. Shenzhen Longsys Electronics Co., Ltd	6:21-cv-00824	WDTX	Aug. 10, 2021
Digital Cache, LLC v. Micro-Star International Co. Ltd.	6:21-cv-00732	WDTX	July 16, 2021
Digital Cache, LLC v. Sharp Corporation	6:21-cv-00733	WDTX	July 16, 2021
Digital Cache, LLC v. Giga-Byte Technology Co., Ltd.	6:21-cv-00719	WDTX	July 13, 2021
Digital Cache, LLC v. NetApp, Inc.	1:21-cv-00926	DDE	June 29, 2021
Digital Cache, LLC v. Pure Storage, Inc.	1:21-cv-00927	DDE	June 29, 2021
Digital Cache, LLC v. Panasonic Corporation of North America	6:21-cv-00676	WDTX	June 28, 2021
Digital Cache, LLC v. ADATA Technology Co., Ltd.	6:21-cv-00584	WDTX	June 8, 2021
Digital Cache, LLC v. Avant Technology, Inc.	6:21-cv-00362	WDTX	April 13, 2021

Digital Cache, LLC v. Western Digital Corporation	6:21-cv-00338	WDTX	April 7, 2021
Digital Cache, LLC v. Seagate Technology LLC	1:21-cv-00536	DCO	Feb. 23, 2021
Digital Cache, LLC v. Kingston Technology (Shanghai) Co., Ltd.	6:21-cv-00163	WDTX	Feb. 23, 2021
Digital Cache, LLC v. Transcend Information Inc.	6:21-cv-00164	WDTX	Feb. 23, 2021
Digital Cache, LLC v. Toshiba America Electronic Components, Inc.	6:21-cv-00161	WDTX	Feb. 22, 2021

C. Lead and Back-up Counsel and Service Information

Pursuant to 37 C.F.R. §42.8(b)(3), Petitioner identifies the following counsel. Powers of attorney accompany this Petition.

Lead Counsel Kyle Howard HAYNES AND BOONE, LLP 2323 Victory Ave. Suite 700 Dallas, TX 75219	Phone: 972-739-6931 Fax: 214-200-0853 kyle.howard.ipr@haynesboone.com USPTO Reg. No. 67,568
Back–up Counsel David M. O'Dell HAYNES AND BOONE, LLP 2323 Victory Ave. Suite 700 Dallas, TX 75219	Phone: 972-739-8635 Fax: 214-200-0853 david.odell.ipr@haynesboone.com USPTO Reg. No. 42,044

Please address all correspondence to lead and back-up counsel. Petitioner consents to electronic service.

III. GROUNDS FOR STANDING

Pursuant to 37 C.F.R. §42.104(a), Petitioner certifies the '015 Patent is available for *inter partes* review ("IPR") and Petitioner is not barred or estopped from requesting an IPR challenging the claims of the patent.

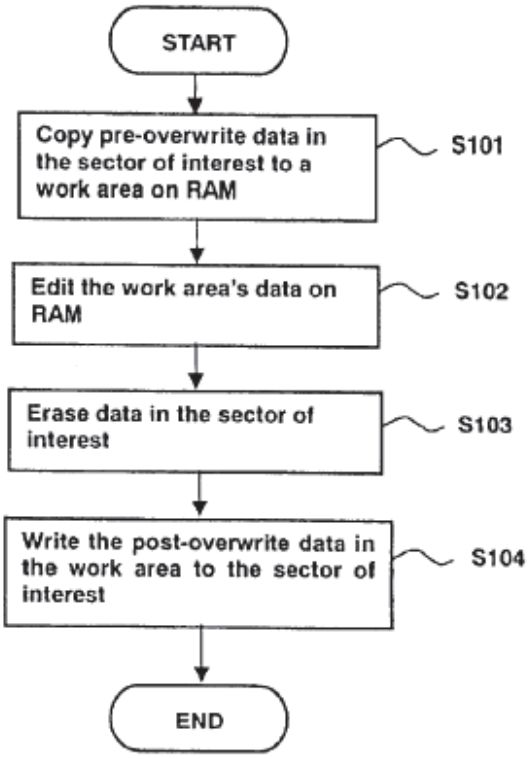
IV. SUMMARY OF '015 PATENT AND PROSECUTION HISTORY

A. Summary of the '015 Patent

The '015 Patent describes "a method of overwriting nonvolatile memory data . . . ." EX1001, 1:14-16; *see also* EX1011, [0035]-[0044]. With respect to overwriting, the '015 Patent acknowledges that flash memory requires all data stored in a sector to be erased prior to overwriting data to that sector – "***Flash memory has a unique property*** that affects the overwriting method: . . . ***one must first erase all data in a sector to which the data to be overwritten."*** EX1001, 1:28-33 (emphasis added).

Because of this unique property, it was already known to overwrite data in flash memory as shown in Fig. 4. *See* EX1001, 1:33-44.

Fig.4



'015 Patent (EX1001), Fig. 4

Specifically, it was known in the art that overwriting data in flash memory included: (1) copying pre-overwrite data[2] from a sector in flash memory into volatile memory (S101); (2) overwriting (i.e. editing) the pre-overwrite data in

[2] The '015 Patent states that pre-overwrite data "is the data to be overwritten" and post-overwrite data "is the new data that overwrites the pre-overwrite data." EX1001, 2:12-14.

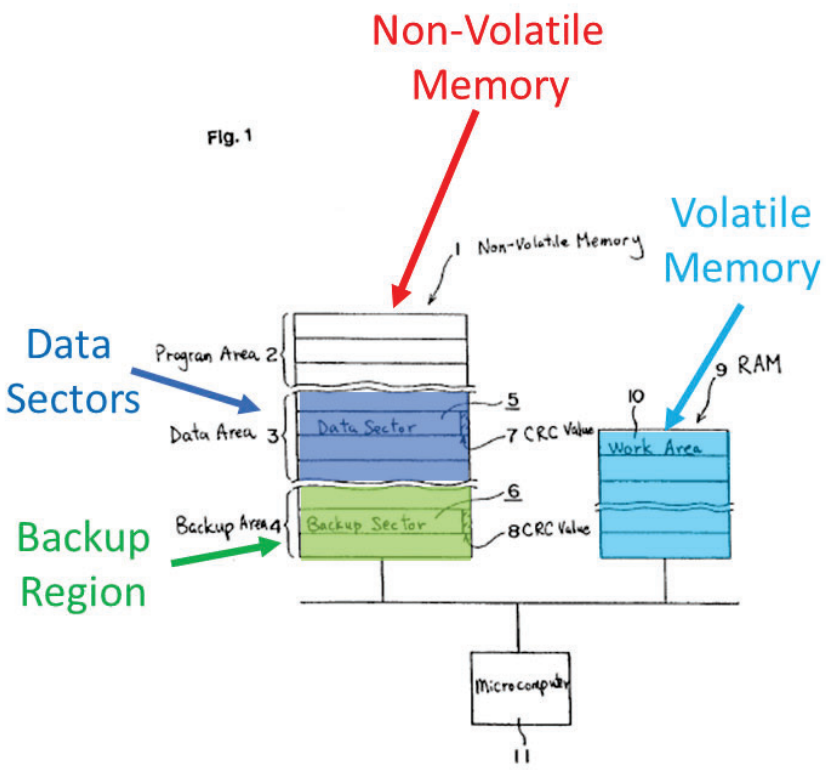
volatile memory to form post-overwrite data (S102); (3) erasing the pre-overwrite data stored in the corresponding sector in flash memory (S103); and (4) writing post-overwrite data into the previously erased sector (S104). EX1011, [0038].

The ’015 Patent further states that data loss could occur if power fails between data erasure and completion of writing post-overwrite data:

> For this reason, ***an interruption of power supply***, if it occurs unexpectedly between the erasure of data in flash memory and the completion of writing the post-overwrite data, interrupts control of flash memory***, causing loss of data saved on RAM, volatile memory; one may be seriously concerned with irrevocable loss of very important data***.

EX1001, 1:51-56 (emphasis added).

To address this possible data loss, the ’015 Patent discloses the nonvolatile memory control apparatus of Fig. 1.



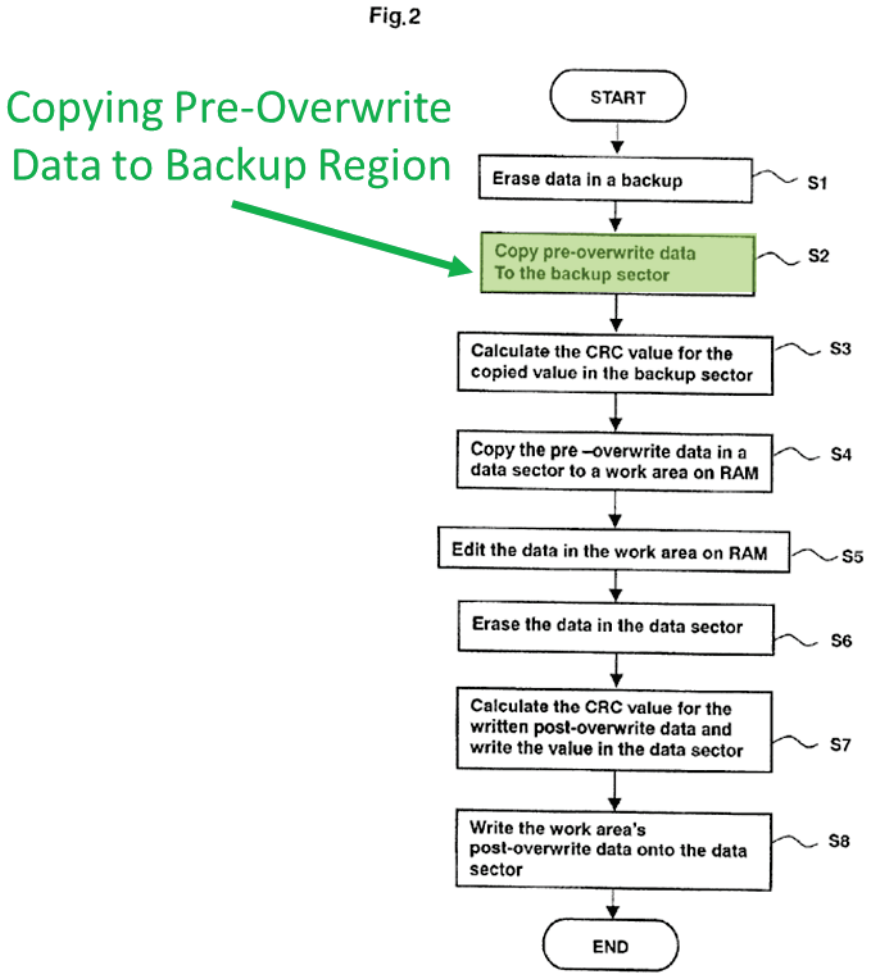
EX1001, Fig. 1

As shown, the apparatus includes nonvolatile memory 1 (i.e. flash memory) "in which required data are stored sector by sector, and provides a backup region[3] 6 having the same memory capacity as sector 5 to which data to be overwritten belongs." EX1001, 3:24-27. The '015 Patent states that "[w]hen overwriting the data of interest in nonvolatile memory 1, pre-overwrite data in sector 5, to which

[3] The '015 Patent interchangeably uses the term "backup region" and "backup sector." The analysis herein uses these terms consistent with the specification of the '015 Patent. EX1011, [0041].

the data to be overwritten belongs, is written in backup region 6 . . .” EX1001, 3:27-30. Thus, the ’015 Patent describes writing pre-overwrite data (i.e. data to be overwritten) to a backup region in nonvolatile memory during the overwriting process.

Fig. 2 below from the ’015 Patent illustrates the overwriting method:



EX1001, Fig. 2

As part of this process, “the control apparatus copies pre-overwrite data in data sector 5 to backup sector 6 (Step 2, S2).” EX1001, 4:19-21. Next, “[t]he

control apparatus copies pre-overwrite data in data sector 5 to work area 10 on RAM 9 (Step 4, S4)" and "edits the data of interest in work area 10 on RAM 9 (Step 5, S5)." EX1001, 4:24-28. Thereafter, "the control apparatus erases the data in data sector 5 (Step 6, S6)" and then "writes the post-overwrite data in work area 10 onto data sector 5 (Step 7, S7)." EX1001, 4:29-31. Accordingly, if power fails during this process, "the pre-overwrite data in flash memory 1 can recover at least its pre-overwrite state." EX1001, 4:37-41.

B. Prosecution History of the '015 Patent

The '015 Patent claims priority to Japanese Patent Application JP2001-153139 ("Japanese Application") filed May 22, 2001. During foreign prosecution, the Japanese Application was rejected due to the Mitomi reference. *See* EX1003 and EX1005. Specifically, the Notice of Reasons for Refusal states "the claimed invention(s) ***could have easily been made by persons who have common knowledge in the technical field*** to which the claimed invention(s) pertains, ***on the basis of the invention(s) described in the distributed publication***(s)," namely the Mitomi reference. EX1003, p.1 (emphasis added). As a result, the Japanese Application was not allowed. Mitomi was never cited nor considered by the USPTO during prosecution of the '015 Patent.

During U.S. prosecution, the Applicant argued the following alleged novel features recited by the claims: (1) preventing data loss by first backing up pre-

overwrite data stored in a non-volatile data sector into a backup region which is also located in the non-volatile memory; (2) copying pre-overwrite data to a volatile memory and editing the pre-overwrite data in volatile memory to produce post-overwrite data; (3) erasing pre-overwrite data from the data sector and then writing post-overwrite data into that same data sector; and (4) upon restoration of power, determining whether the data in the data sector is valid. *See* EX1002, p.74; *see also* EX1011, [0051]-[0055] After filing the response, the patent was allowed and issued on February 1, 2005. Ex1002, p.38.

V. LEVEL OF ORDINARY SKILL IN THE ART

The level of ordinary skill in the art may be reflected by the prior art of record. *Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001). However, to the extent a definition is needed, a person of ordinary skill in the art ("POSITA") at the time of the earliest Japanese filing (2001) would have had a bachelor's degree in electrical engineering, physics, or equivalent training, and two years of technical experience in the field of non-volatile memory including flash memory technology. Furthermore, a person with more technical education but less experience could also meet the relevant standard for POSITAs. Petitioner's technical expert, Dr. R. Jacob Baker, whose declaration this Petition cites, was at least a POSITA at this time. EX1011, [0005]-[0032].

VI. CLAIM CONSTRUCTION

During IPR, claims are construed according to the standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). *See* 83 Fed. Reg. 51341 (Oct. 11, 2018). Petitioner believes that, for the purposes of this proceeding and the analysis presented herein, no claim term requires express construction. *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999). Accordingly, this Petition analyzes the claims consistent with the ordinary and customary meaning as would be understood by a POSITA in light of the specification. *Phillips*, 415 F.3d at 1314-17; EX1011, [0033]-[0034] and [0056] .

VII. REQUESTED RELIEF

Petitioner asks that the Board review the accompanying prior art and analysis, institute a trial for *inter partes* review of claims 1-11 and cancel those claims.

VIII. IDENTIFICATION OF CHALLENGE

A. Challenged Claims and Statutory Grounds

This Petition challenges claims 1-11 of the '015 Patent on the following grounds. The '015 Patent was filed on May 20, 2002, with a priority claim to May 22, 2001. As such, the patent is subject to 35 U.S.C. §§ 102, 103 (pre-AIA).

Ground	Claim(s)	Basis
Ground #1	1, 2 and 6-11	35 U.S.C. §103 over **Morihiro** (EX1006)
Ground #2	3-5	35 U.S.C. §103 over **Morihiro and Forsman** (EX1007)
Ground #3	11	35 U.S.C. §103 over **Morihiro and Mitomi** (EX1005)

B. Status as Prior Art

All prior art pre-dates the earliest claimed priority date of the '015 Patent of May 22, 2001.

Morihiro, which was filed December 16, 1993 and published July 4, 1995, is prior art under at least 35 U.S.C. §102(b).

Forsman, which was filed August 3, 2000 and published Dec. 16, 2003, is prior art under at least 35 U.S.C. §102(e).

Mitomi, which was filed October 24, 1996 and published May 15, 1998, is prior art under at least 35 U.S.C. §102(b).

IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Ground #1: Claims 1, 2 and 6-11 are obvious over Morihiro.

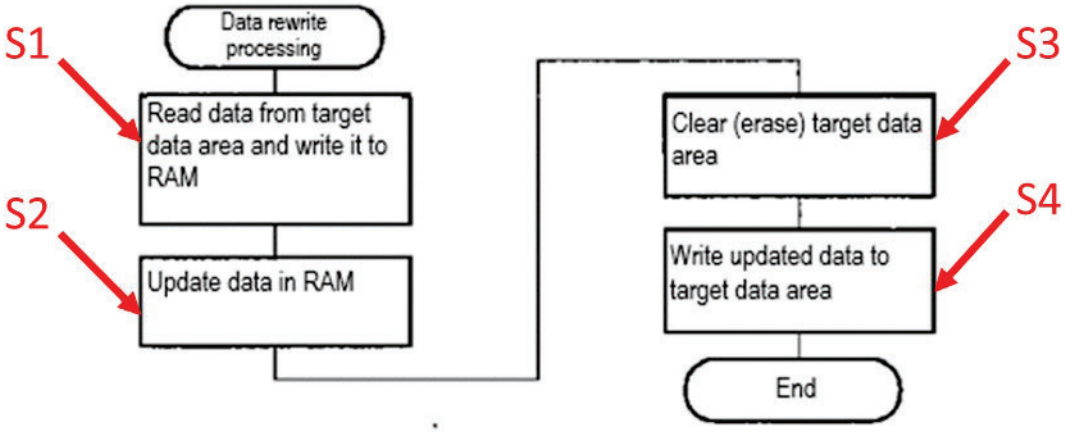
1. Summary of Morihiro

Morihiro is directed "[t]o prevent[ing] data from being lost and becoming unrecoverable ***due to electric service interruption***, etc. in cases where data stored on . . . nonvolatile memory is being updated." EX1006, Abstract (emphasis

added); *see also* EX1011, [0057]-[0072].

In that regard, as shown below in Fig. 5, Morihiro recognized that overwriting data in flash memory includes: (1) copying pre-overwrite data from a data area (i.e. sector) in flash memory into volatile memory (S1); (2) overwriting (i.e. editing/rewriting) the pre-overwrite data in volatile memory to form post-overwrite data (S2); (3) erasing the pre-overwrite data stored in the corresponding data area (i.e. sector) in flash memory (S3); and (4) writing post-overwrite data into the previously erased data area (i.e. sector) (S4). *See* EX1006, [0005]-[0006]; EX1011, [0060].

[FIG. 5]



EX1006, Figure 5

Accordingly, as evidenced by the '015 Patent and Morihiro, a POSITA understood that flash memory requires erasure of the data being stored in a data area (i.e. sector) prior to the updated data being written to the previously erased

data area. EX1011, [0061]. A POSITA would have understood this erasure and then writing of updated data is considered the "overwriting" of data in the '015 Patent and Morihiro. *See* EX1001, 1:33-44, EX1006, [0005]-[0007] and EX1005, [0002]-[0003]; *see also* EX1011, [0061]. Therefore, even though Morihiro states that in flash memory, "writing new data over already written data (overwriting) is not possible," a POSITA would have understood that Morihiro discloses "overwriting" data in the same way as the '015 Patent. EX1011, [0059]-[0061].

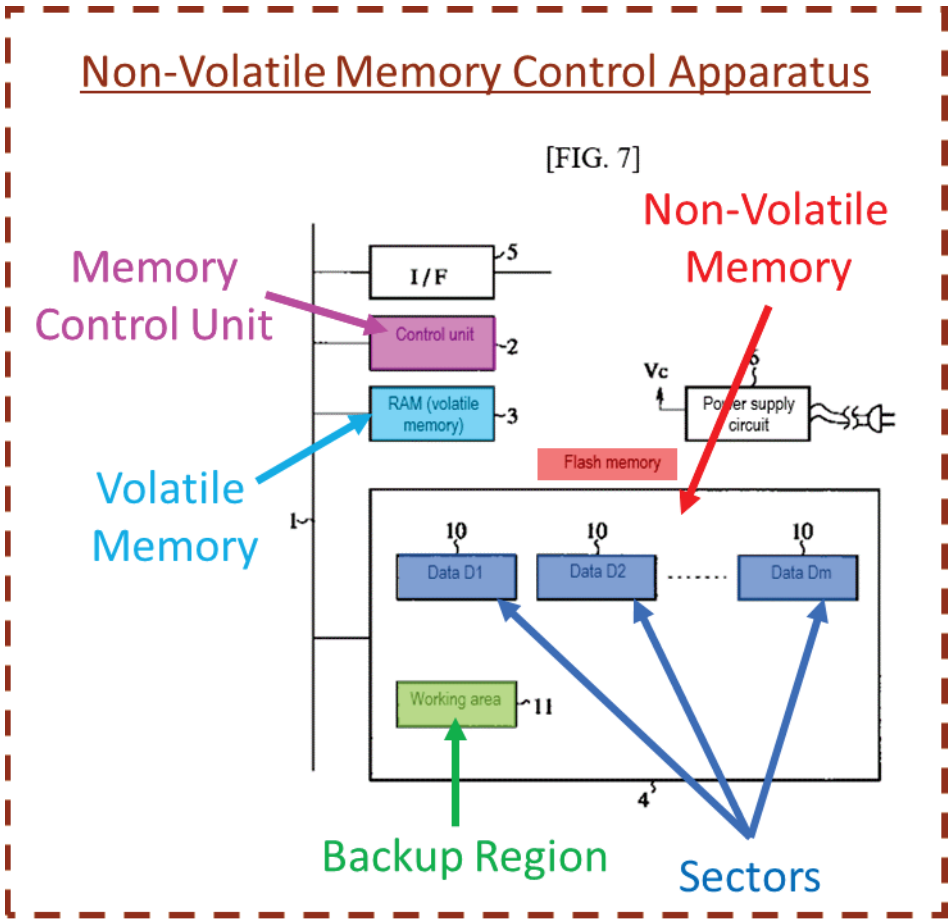
Still further, Morihiro recognized that data loss could occur if power fails between data erasure and completion of writing post-overwrite data:

> Therefore, ***if the power supply of a computer is interrupted due to electric service interruption, etc. during the execution of steps (3) and (4), in which the data area of the flash memory is cleared and updated data is then written to the cleared data area, the updated data retained in volatile memory such as RAM will be lost***, and since this data has not been stored in both volatile memory and flash memory, there is no means for recovering the lost data, for example, upon restoration of electric service.

EX1006, [0008] (emphasis added).

To address this possible data loss, Morihiro discloses the nonvolatile memory control apparatus shown below in Fig. 7 – "FIG. 7 is a block diagram

illustrating the simplified configuration of ***a computer employing a data updating method for nonvolatile memory*** of another example of the present invention." EX1006, [0041] (emphasis added); *see also* EX1011, [0062]-[0063].



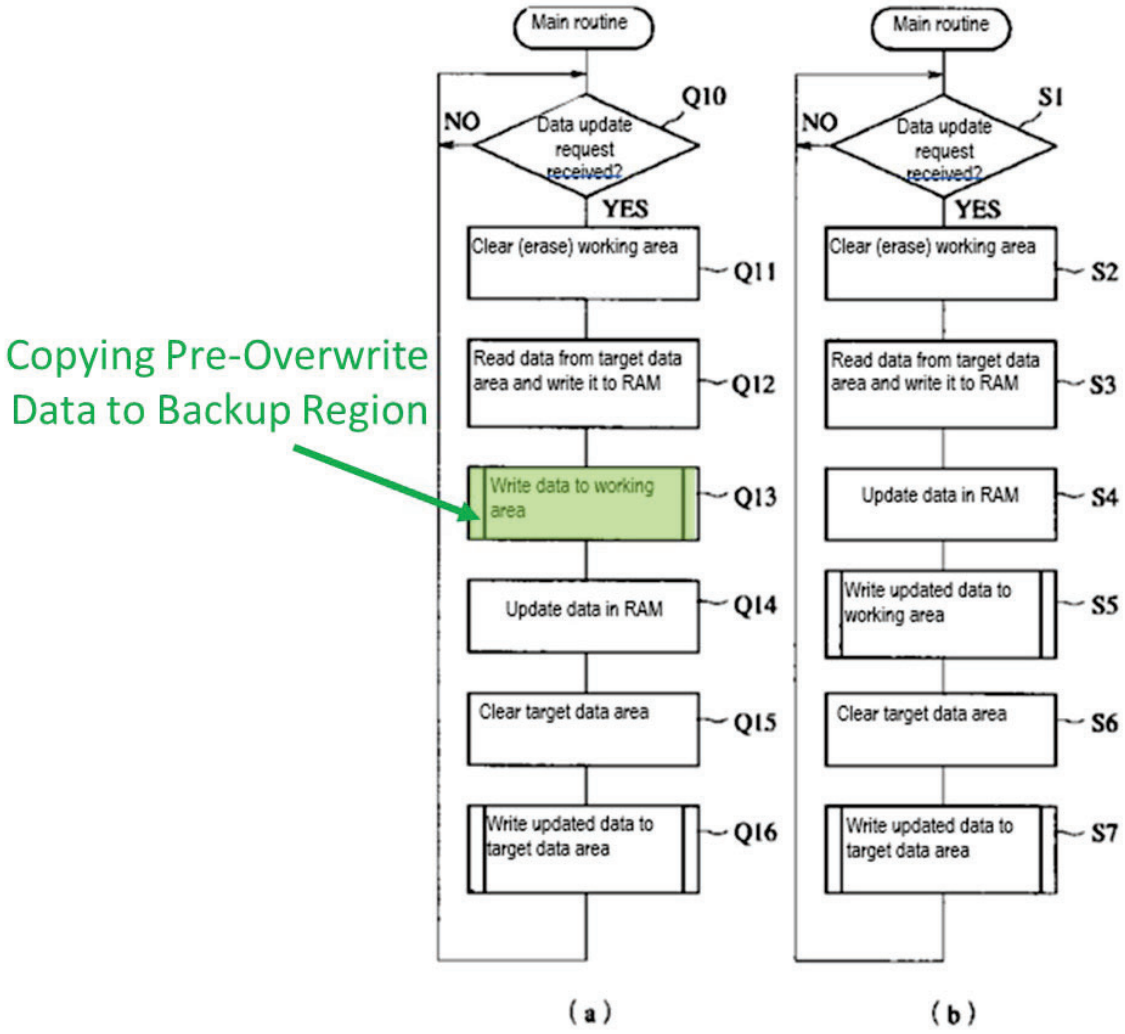
EX1006, Fig. 7

As shown, Morihiro discloses that "data areas 10 . . . are formed in the flash memory 4" and that "a working area 11 for temporarily storing data during data updating is provided within this flash memory 4." EX1006, [0042]. That is, Morihiro discloses writing pre-overwrite data (i.e. data to be overwritten) to a backup region (i.e. working area 11) in flash memory 4 during the overwriting

process. EX1011, [0063]-[0064]. Thus, "in the event that data is corrupted due to electric service interruption, etc. in the update process of the data of data area 10, the lost data can be recovered using the data from before the update stored in the working area 11." EX1006, [0054].

Fig. 10(a) below from Morihiro illustrates the overwriting process implemented by the above nonvolatile memory control apparatus. EX1011, [0065]-[0066].

[FIG. 10]



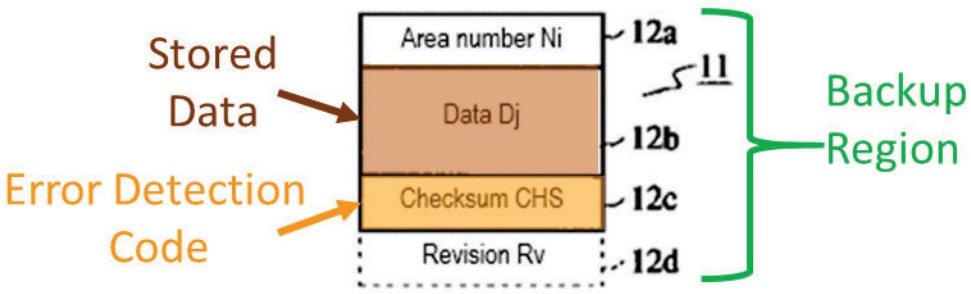
EX1006, Figs. 10(a)-(b)

As part of overwriting, Morihiro states "the data area 10 in which the data Dj indicated by the update request is stored is identified and the data Dj from the target data area 10 is read and loaded into RAM 3 (Q12)" and that "the data from before the update in RAM 3 is written to the working area 11 of the flash memory 4." EX1006, [0050]-[0051]. In other words, Morihiro discloses that "when a data

update request is issued, ***the data to be updated is copied first into the working area 11***, after which updating of the data of the target data area 10 is performed." EX1006, [0053] (emphasis added). Next, Morihiro discloses that the "data update processing for the current data update request is performed on the data in RAM 3 (Q14)." EX1006, [0052]. Thereafter, Morihiro discloses "clearing the target data area 10 of the flash memory 4 (Q15)" and then "the updated data is written to the target data area 10 (Q16)." EX1006, [0052]

Moreover, as shown below in Fig. 8, Morihiro discloses that working area 11 includes "an area 12b for storing data D_j, and an area 12c for storing a checksum CHS for checking if the stored data D_j is normal or not." EX1006, [0043]; *see also* EX1011, [0067].

[FIG. 8]

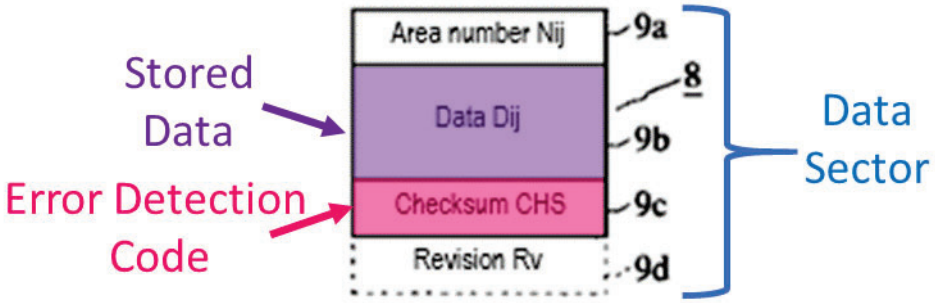


EX1006, Fig. 8

Still further, Morihiro states that "[t]he configuration of each of said data areas 10 follows the configuration of data areas shown in FIG. 2." EX1006, [0044]. As shown below, each data area 10 includes "an area 9b for storing data

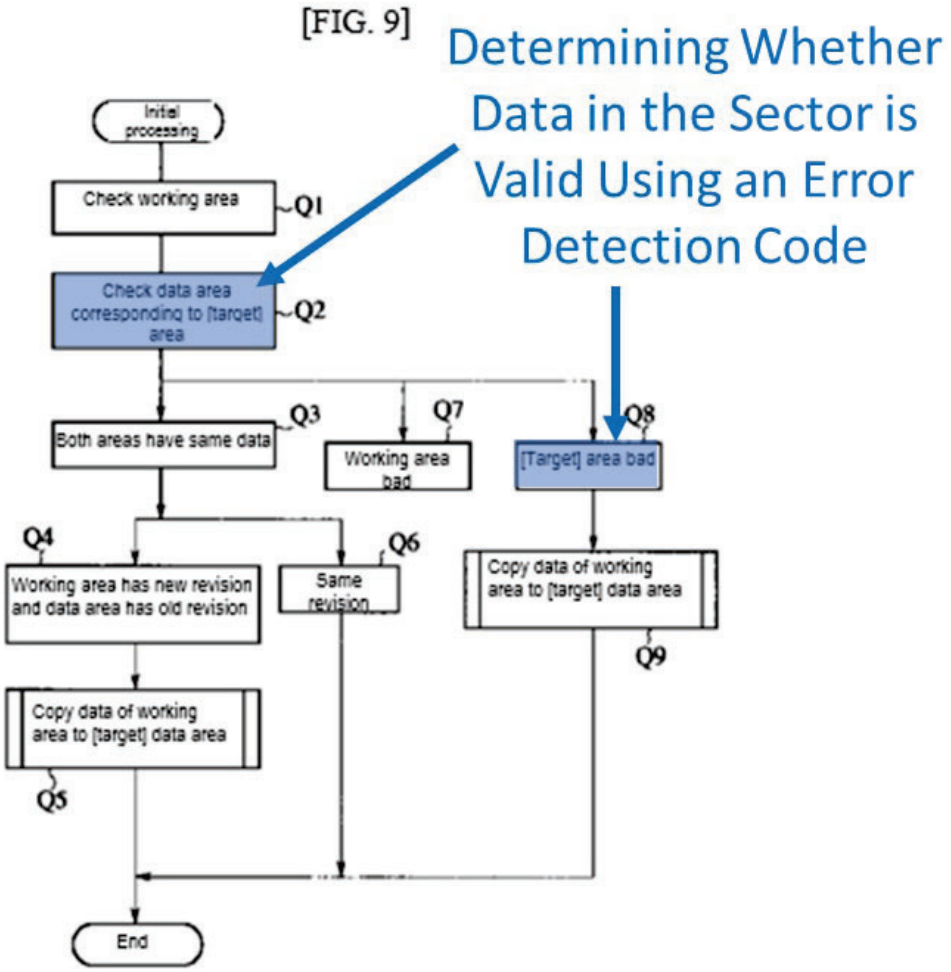
D_{ij}, and an area 9c for storing a checksum CHS for checking if the stored data D_{ij} is normal or not." Morihiro [0022]. That is, Morihiro discloses that each data area 10 includes an error detection code, namely a checksum CHS, for detecting whether the stored data has errors. EX1011, [0068].

[FIG. 2]



EX1006, Fig. 2

Morihiro explains that, by using the above nonvolatile memory control apparatus, "in the event that data is corrupted due to electric service interruption, etc. in the update process of the data of data area 10, the lost data can be recovered using the data from before the update stored in the working area 11." EX1006, [0054]. In that regard, Morihiro discloses that "[t]he control unit 2, ***responding to . . . restoration of electric service***, executes initialization processing of each data area 10 of the flash memory 4 shown in FIG. 9." Morihiro [0044]; *see also* EX1011, [0069]-[0071].



EX1006, Fig. 9

As shown above in Fig. 9[4], upon restoration of electric service, the validity of each data area 10 is checked. Specifically, Morihiro states "[i]n the error check of Q2, ***if an error is discovered in the data D of the target data area 10*** (Q8), the

[4] The text in Fig. 9 has inadvertent typographical errors. Brackets have been placed around the term "target" which is consistent with the explanation of the figure in the specification of Morihiro. *See* EX1011, [0071]

data D of the working area 11 is written (copied) to the data area 10 (Q9)." EX1006, [0048] (emphasis added). Moreover, because Morihiro teaches that data area 10 includes a checksum CHS for checking if the stored data is normal or not, it would have been obvious to a POSITA to check the data area 10 in step Q2 with the stored checksum CHS. EX1011, [0071]. It was well known in the art to utilize the checksum associated with the data to check for errors in the data. (*See* EX1007, 5:21-27, discussing checksums used to check the validity of data – "CRC is a technique where a 'checksum' is appended to the end of a block of data that is being checked for possible corruption. A new checksum is calculated based on the data received and compared with the checksum appended to the data. If the two values agree, it is highly likely that the data has not been corrupted.").

Accordingly, Morihiro discloses the argued novel features: (1) preventing data loss by first backing up pre-overwrite data stored in a non-volatile data sector into a backup region which is also located in the non-volatile memory; (2) copying pre-overwrite data to a volatile memory and editing the pre-overwrite data in volatile memory to produce post-overwrite data; (3) erasing pre-overwrite data from the data sector and then writing post-overwrite data into that same data sector; and (4) upon restoration of power, determining whether the data in the data sector is valid. *See* EX1002, p.74; *see also* EX1011, [0055] and [0072]. Moreover, Morihiro even performs these processes so that if there is any power

interruption, at least the old data (pre-overwrite data) can be saved if it turns out that the data sector contains corrupted data. *Id*.

2. Claim 1

[1.0] ***A method of overwriting nonvolatile memory data in which data of interest is stored sector by sector, comprising, when overwriting data of interest, the steps of:***

Morihiro teaches claim element [1.0]. EX1011, [0073]-[0082].

Morihiro states that "[t]he present invention relates to a ***data updating method for nonvolatile memory*** in which data items stored in a non-overwritable nonvolatile memory are updated." Morihiro [0001] (emphasis added). More specifically, Morihiro's method includes:

> Furthermore, in the invention as in claim 2, ***multiple data areas for storing data items and one working area for temporarily storing data are formed in a nonvolatile memory;*** in response to a data update request, data of a data area corresponding to data for which updating has been requested is read and written to a volatile memory; the working area is cleared and the data that was stored in the volatile memory is written to the cleared working area; ***the data which has been stored in the volatile memory is updated; and the stored content of the data area is cleared and the updated data is written to the cleared data area***.

Morihiro [0012] (emphasis added).

As discussed above, a POSITA would have understood that Morihiro discloses "overwriting" data in the same way as the '015 Patent. EX1011, [0059]-[0061] and [0075]-[0078]. As evidenced by both the '015 Patent and Morihiro, a POSITA understood that flash memory requires erasure of the data being stored in a data area (i.e. sector) prior to the updated data being written to the previously erased data area (i.e. sector). *Id*. A POSITA would have understood this erasure and then writing of updated data is considered the "overwriting" of data in the '015 Patent and Morihiro. *Id*.; *see also* EX1001, 1:33-44, EX1006, [0005]-[0007] and EX1005, [0002]-[0003].

Still further, Morihiro discloses that the data areas and working area "are formed in the flash memory." Morihiro [0042]. Morihiro may not explicitly use the term "sector" when discussing Morihiro's data areas and working area. However, a POSITA would have understood that Morihiro's data areas and working area are sectors. EX1011, [0079]-[0082].

Similar to Morihiro, U.S. Patent No. 6,834,331 to Liu "relates to methods for erasing a flash memory, for writing data to a flash memory, and for reading data from a flash memory, all with improved data integrity." EX1004, 1:17-19. Moreover, as evidenced by Liu, a POSITA knew that flash memory includes sectors and that the writing, reading and erasing of data is performed on these sectors:

> ***A flash memory generally consists of a number of sectors of memory locations***. . . . ***The most basic functions performed on a flash memory include writing data, reading data, and erasing data. Flash memory may only be erased an entire sector at a time***. . . .The flash memory may then be written or programmed by changing selected bits to a binary zero.

EX1004, 1:20-33 (emphasis added); *see also* EX1011, [0080].

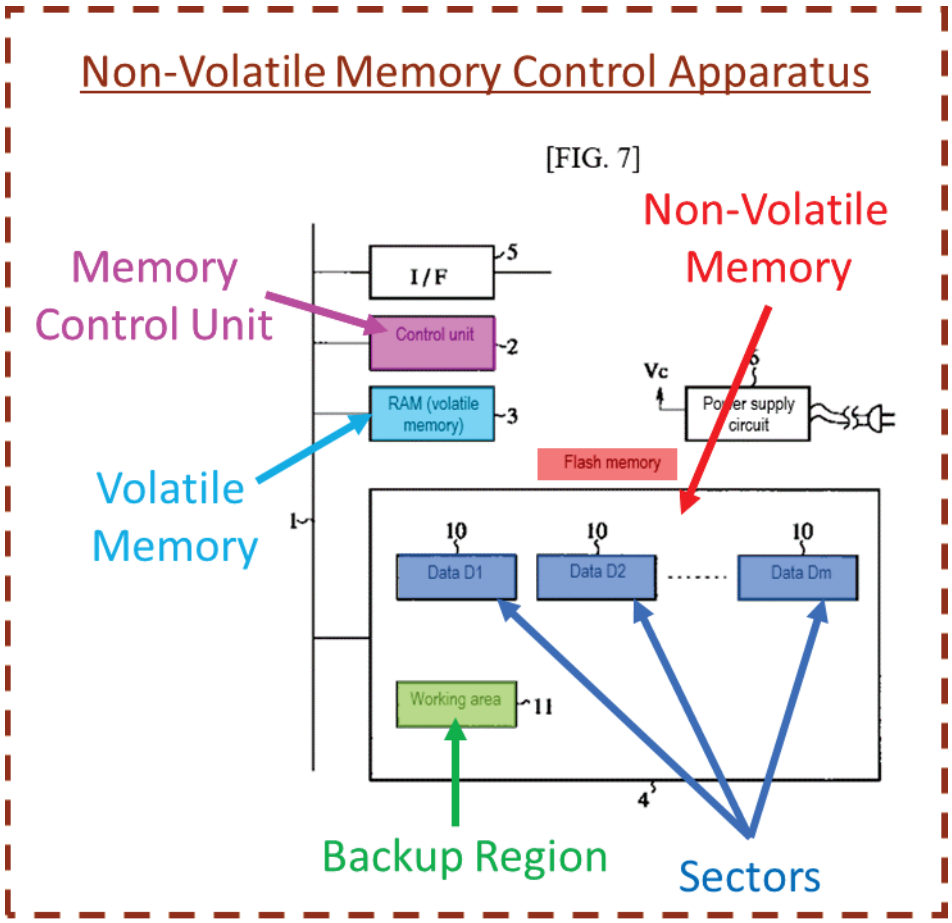
Accordingly, a POSITA would have understood that Morihiro's data areas and working area are data sectors because, like Liu and the '015 Patent, Morihiro teaches these areas are (1) formed in flash memory; (2) used for storing data, (3) erased entirely before a writing process occurs thereto and (4) rewritten on an area basis. EX1011, [0035]-[0044], [0058]-[0072], and [0074]-[0082].

Therefore, as evidenced by Liu, a POSITA would have understood that Morihiro's "data updating method for nonvolatile memory in which data items stored in a non-overwritable nonvolatile memory are updated" teaches *"[a] method of overwriting nonvolatile memory data in which data of interest is stored sector by sector, comprising, when overwriting data of interest, the steps of*," as claimed.

[1.1] ***providing a backup region having at least the same memory capacity as one sector in which data to be overwritten is stored, said backup region being provided in non-volatile memory;***

Morihiro teaches claim element [1.1]. EX1011, [0083]-[0091].

Figure 7 below illustrates the nonvolatile memory control apparatus employing the data updating method disclosed by Morihiro. EX1011, [0084]; *see also* EX1006, [0041].



EX1006, Fig. 7

As shown, Morihiro discloses that "data areas 10 . . . are formed in the flash memory 4" and that "a working area 11 for temporarily storing data during data updating is provided within this flash memory 4." EX1006, [0042]. That is, Morihiro discloses a backup region (i.e. working area 11) and data sectors (i.e. data

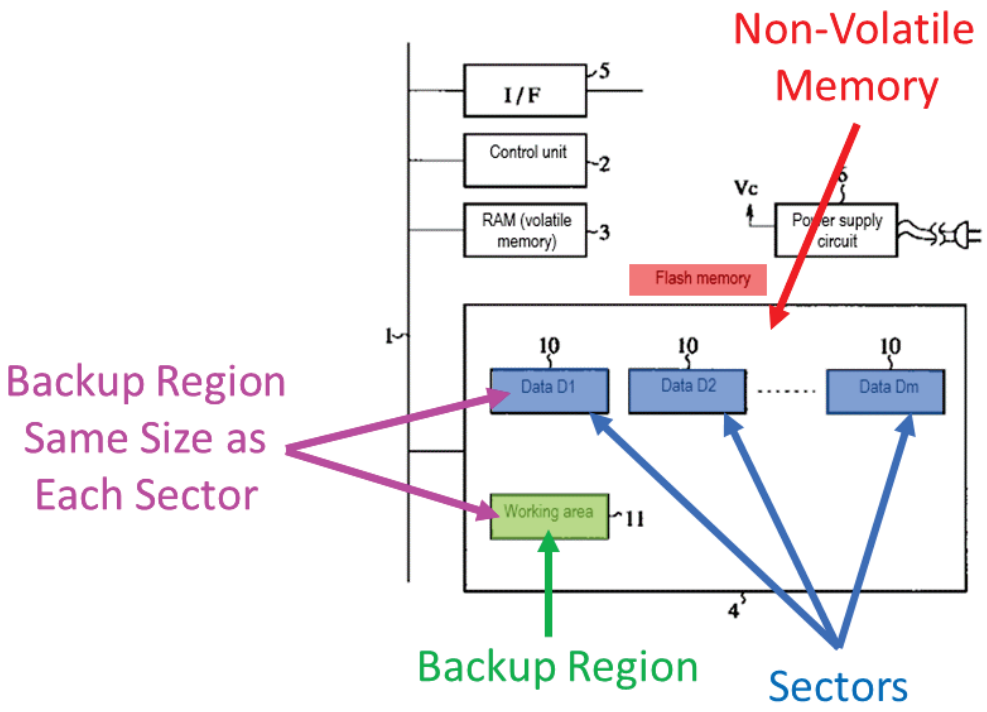
areas 10) in a non-volatile memory and that the data areas 10 store data to be overwritten:

> Furthermore, in the invention as in claim 2, ***multiple data areas for storing data items and one working area for temporarily storing data are formed in a nonvolatile memory; in response to a data update request, data of a data area corresponding to data for which updating has been requested is read and written to a volatile memory; the working area is cleared and the data that was stored in the volatile memory is written to the cleared working area***; the data which has been stored in the volatile memory is updated; and the stored content of the data area is cleared and the updated data is written to the cleared data area.

Morihiro [0012] (emphasis added); *see also* EX1011, [0084]-[0085].

Still further, as shown below, Morihiro discloses that working area 11 is the same relative size as each data area 10. EX1011, [0084]-[0086].

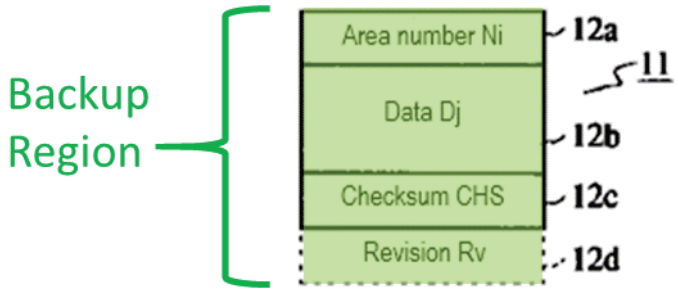
[FIG. 7]



EX1006, Fig. 7

Moreover, Fig. 8 below illustrates the configuration of Morihiro's backup region, namely, working area 11:

[FIG. 8]



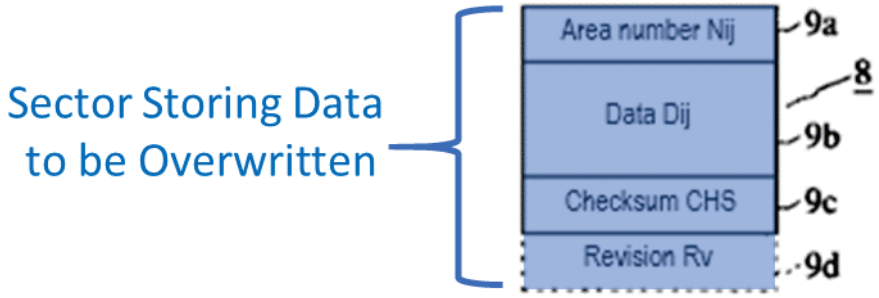
EX1006, Fig. 8

Within the working area 11, as shown in FIG. 8, there is formed an area 12a for storing the area number N_j of the

> data area 10 of the data temporarily stored in the working area 11, ***an area 12b for storing data D_j***, and an area 12c for storing a checksum CHS for checking if the stored data D_j is normal or not.

Morihiro [0043] (emphasis added); *see also* EX1011, [0087]. And, as shown below in Fig. 2, Morihiro discloses that each data area 10 (i.e. one sector) includes data to be overwritten and has the same configuration as working area 11:

[FIG. 2]



EX1006, Fig. 2

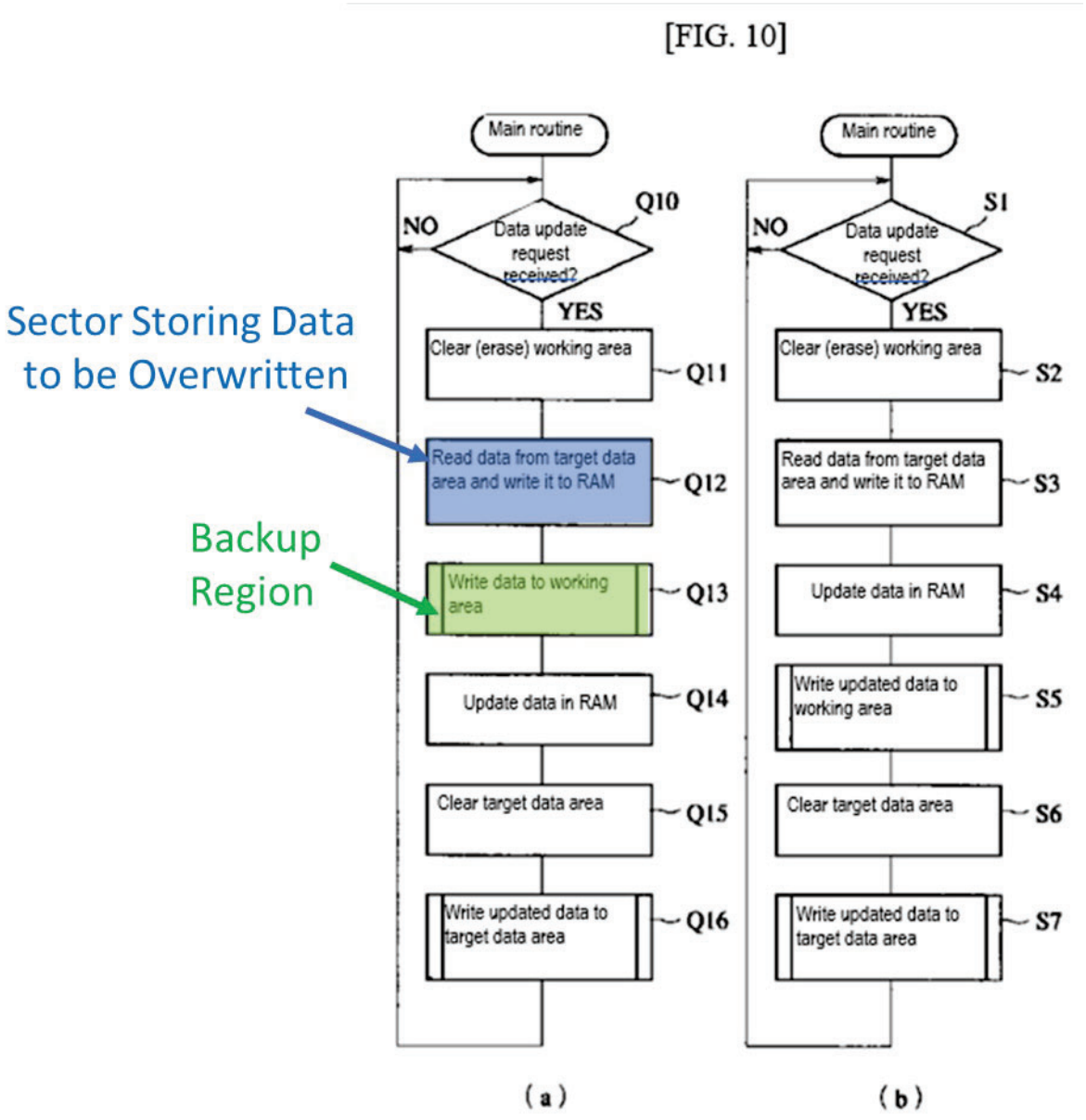
> "The ***configuration of each of said data areas 10*** follows the configuration of data areas ***shown in FIG. 2***."

Morihiro [0044] (emphasis added).

> "***[A]s shown in FIG. 2***, there is formed an area 9a for storing the area number N_{ij}, ***an area 9b for storing data D_{ij}***, and an area 9c for storing a checksum CHS for checking if the stored data D_{ij} is normal or not."

Morihiro [0022] (emphasis added); *see also* EX1011, [0087].

Still further, Morihiro states that "upon receiving input of a data update request from outside, the control unit 2 executes data update processing in accordance with the flow chart shown in FIG. 10 (a)." Morihiro [0049]; *see also* EX1011, [0088].



EX1006, Figs. 10(a)-(b)

As shown above, Morihiro's data update processing includes "***the data area 10 in which the data D_j indicated by the update request is stored*** is identified and the data D_j from the target data area 10 is read and loaded into RAM 3 (Q12)" and "***[t]he data from before the update in RAM 3 is written to the working area 11 of the flash memory 4."*** Morihiro [0050]-[0051] (emphasis added). That is, Morihiro discloses that "when a data update request is issued, ***the data to be updated is copied first into the working area 11***, after which updating of the data of the target data area 10 is performed." EX1006, [0053] (emphasis added).

Accordingly, because Morihiro discloses that working area 11 and data area 10 have the same relative size (*See* Fig. 7) and configuration (*See* Figs. 2 and 8) in combination with working area 11 configured to temporarily store the data of data area 10, a POSITA would have understood that working area 11 has at least the same memory capacity as data area 10. EX1011, [0084]-[0091]. Otherwise working area 11 may not be able to temporarily store the data of data area 10 in the data updating process disclosed by Morihiro. *Id*.

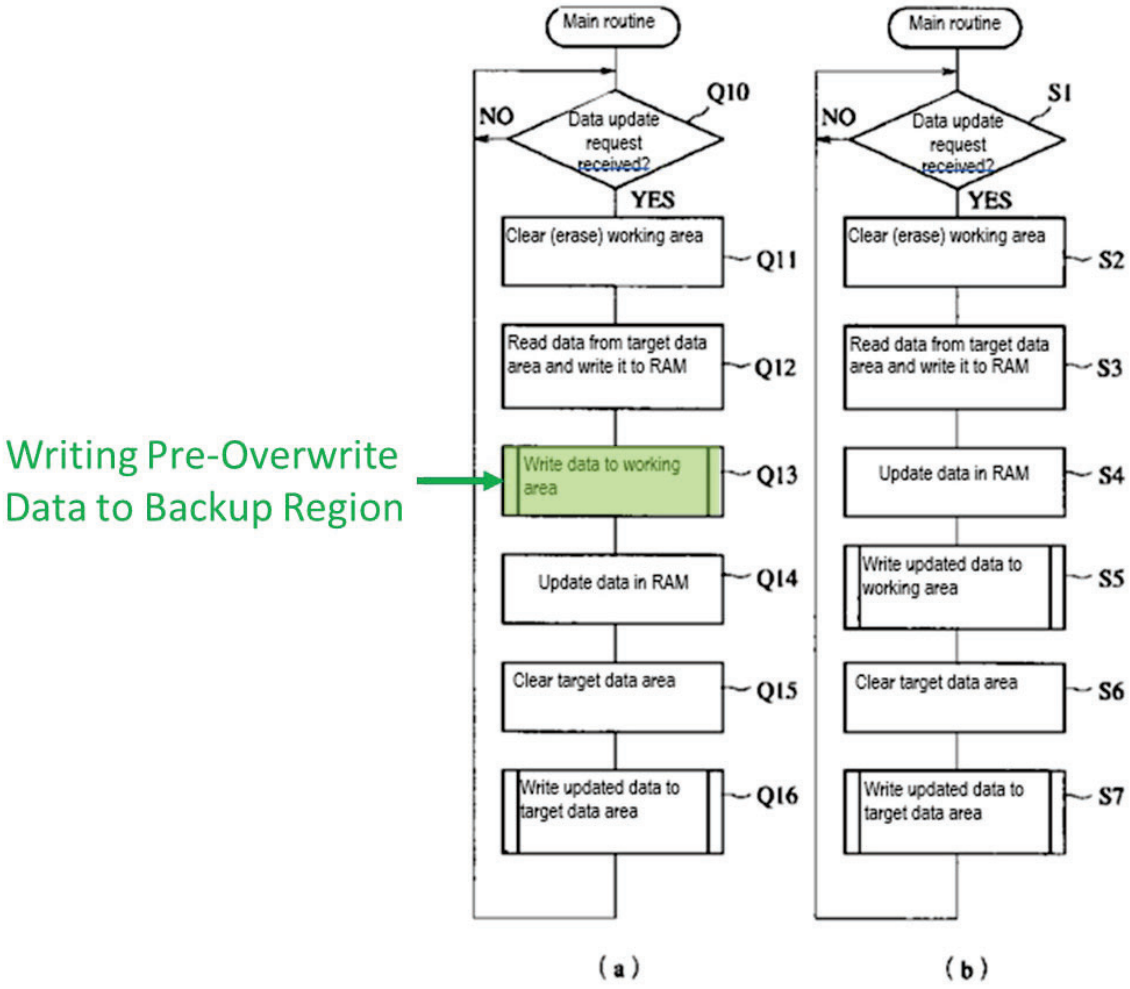
Thus, Morihiro renders obvious "*providing a backup region having at least the same memory capacity as one sector in which data to be overwritten is stored, said backup region being provided in non-volatile memory*" as claimed.

[1.2] ***writing in said backup region pre-overwrite data stored in said one sector in which data to be overwritten is stored along with an error detection code for said pre-overwrite data;***

Morihiro teaches claim element [1.2]. EX1011, [0092]-[0096].

Morihiro states that "upon receiving input of a data update request from outside, the control unit 2 executes data update processing in accordance with the flow chart shown in FIG. 10 (a)." Morihiro [0049]; *see also* EX1011, [0093].

[FIG. 10]



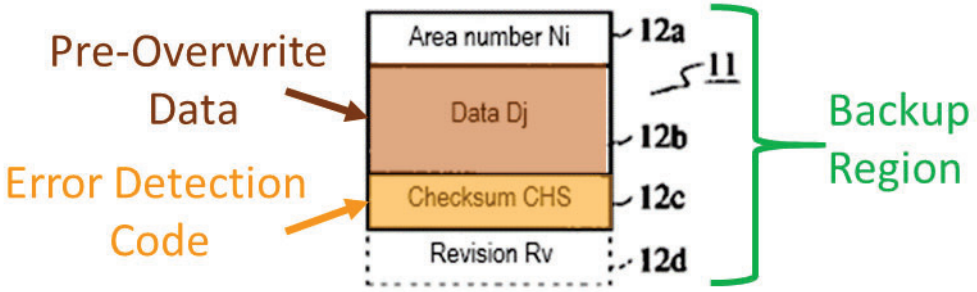
EX1006, Figs. 10(a)-(b)

As shown above, "[w]hen a data update request is inputted in Q10" then "***the working area 11 is cleared*** (Q11)." Morihiro [0050] (emphasis added). Next, "***the data area 10 in which the data D_j indicated by the update request is stored*** is identified and the data D_j from the target data area 10 is read and loaded into RAM 3 (Q12)" and "[t]***he data from before the update in RAM 3 is written to the working area 11 of the flash memory 4."*** Morihiro [0050]-[0051] (emphasis added). That is, Morihiro discloses that "when a data update request is issued, ***the data to be updated is copied first into the working area 11***, after which updating of the data of the target data area 10 is performed." EX1006, [0053] (emphasis added).

Still further, Morihiro discloses that "[w]ithin the working area 11, as shown in FIG. 8, ***there is formed . . . an area 12b for storing data D_j, and an area 12c for storing a checksum CHS for checking if the stored data D_j is normal or not***." EX1006, [0043] (emphasis added). And, as discussed above, Morihiro teaches that "***the working area 11 is cleared*** (Q11)" prior to "[t]he data . . . [being] written to the working area 11 of the flash memory 4" at Q13. EX1006, [0050]-[0051] and Fig. 10(a) (emphasis added). That is, Morihiro teaches that when "the data to be updated is copied first into the working area 11" that the process includes writing the pre-overwrite data along with an error detection code, namely a checksum

CHS, to the previously cleared working area 11. EX1006, [0053]; *see also* EX1011, [0093]-[0096].

[FIG. 8]



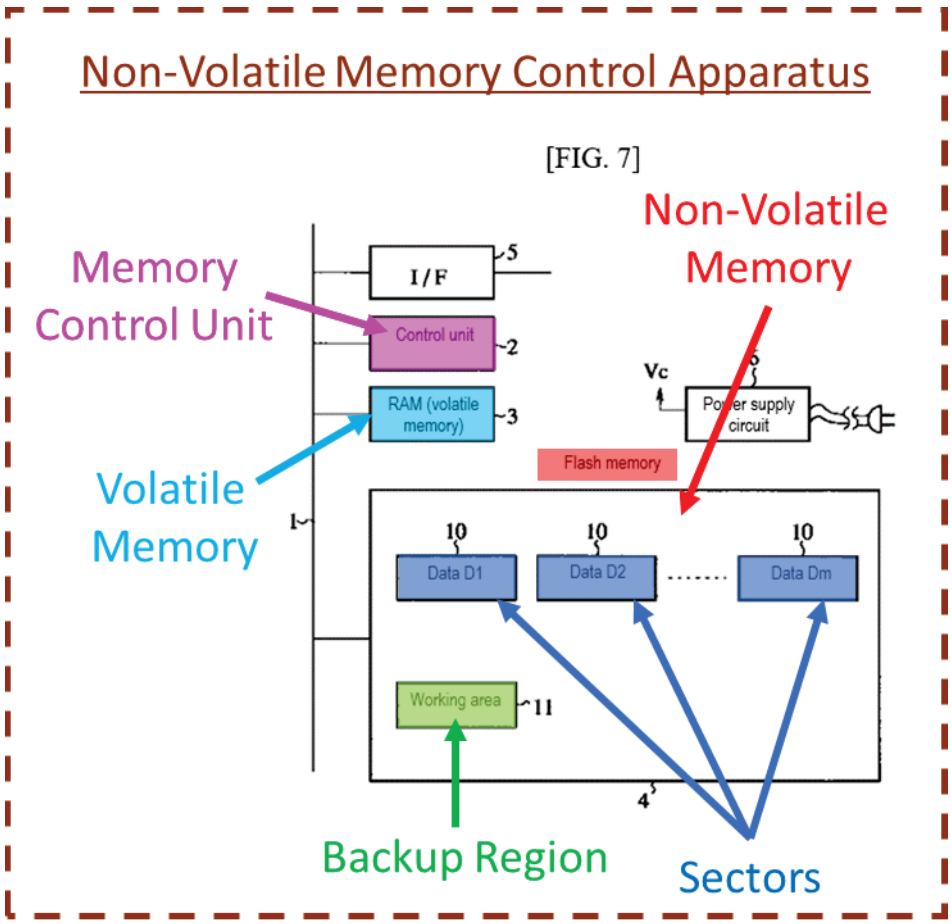
EX1006, Fig. 8

Thus, Morihiro renders obvious "*writing in said backup region pre-overwrite data stored in said one sector in which data to be overwritten is stored along with an error detection code for said pre-overwrite data*" as claimed.

[1.3] ***copying said pre-overwrite data to a volatile memory;***

Morihiro teaches claim element [1.3]. EX1011, [0097]-[0101].

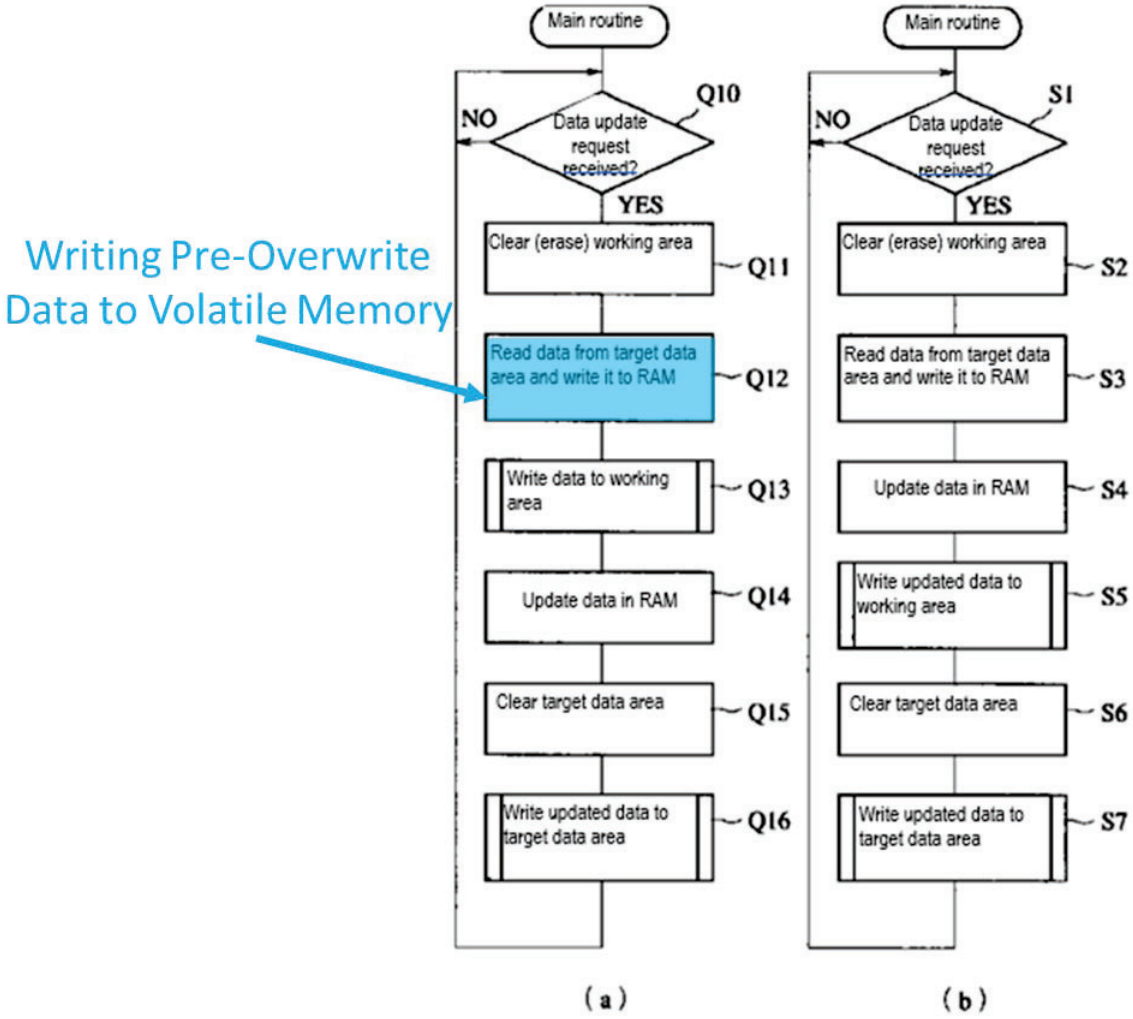
As shown, Morihiro's nonvolatile memory control apparatus includes "RAM 3 as a ***volatile memory*** for storing various types of variable data . . . ." EX1006, [0020] (emphasis added); *see also* EX1011, [0098].



EX1006, Fig. 7

Morihiro states that "upon receiving input of a data update request from outside, the control unit 2 executes data update processing in accordance with the flow chart shown in FIG. 10 (a)." Morihiro [0049].

[FIG. 10]



EX1006, Figs. 10(a)-(b)

As shown above, "the data area 10 in which the data D_j indicated by the update request is stored is identified and ***the data D_j from the target data area 10 is read and loaded into RAM 3*** (Q12)." Morihiro [0050] (emphasis added). That is, Morihiro teaches "in response to a data update request, ***data of a data area corresponding to data for which updating has been requested is read and written***

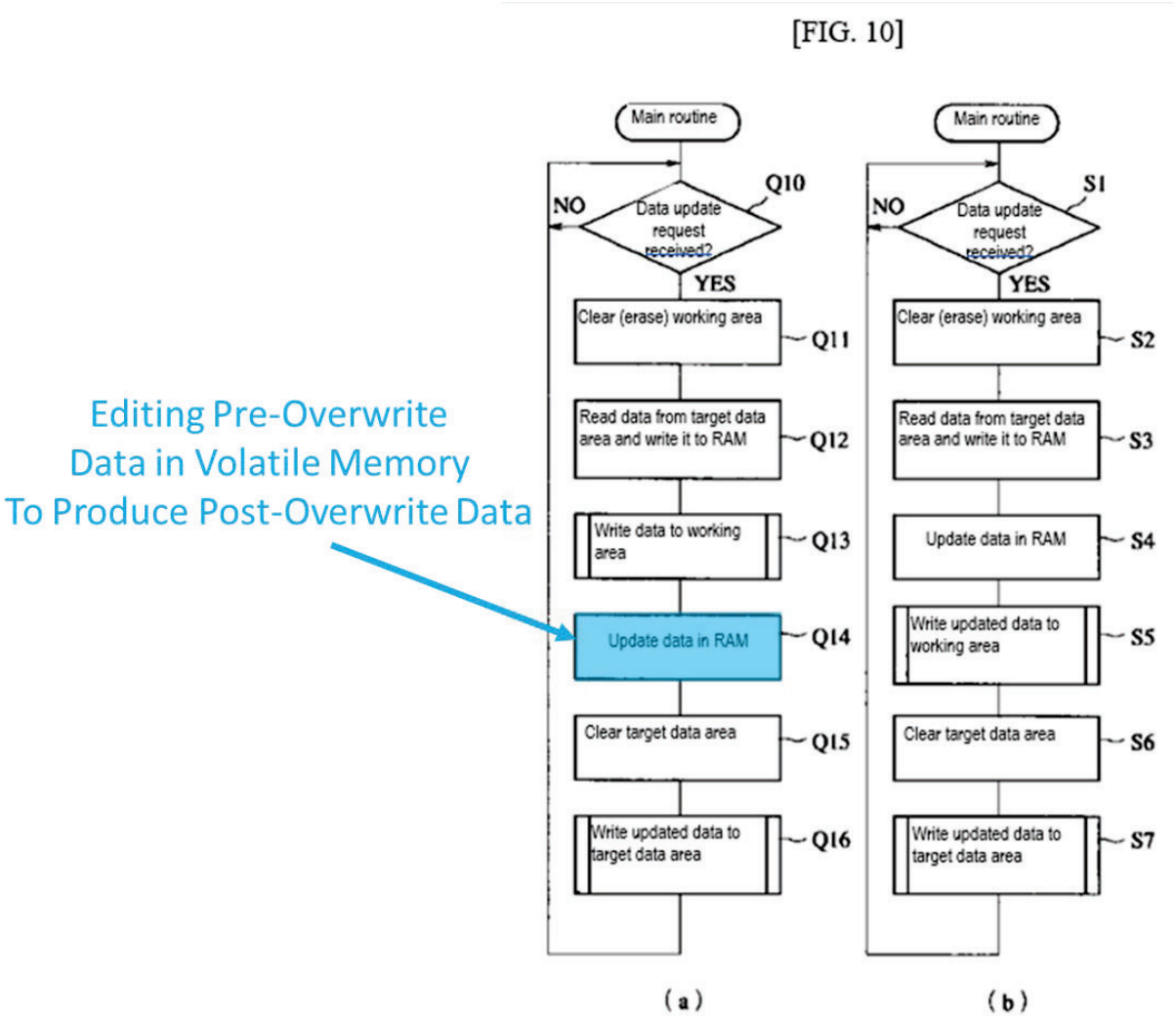
to a volatile memory." EX1006, [0012] (emphasis added); *see also* EX1011, [0097]-[0101].

Thus, Morihiro renders obvious "*copying said pre-overwrite data to a volatile memory"* as claimed.

[1.4] ***editing said pre-overwrite data stored in said volatile memory to produce post-overwrite data;***

Morihiro teaches claim element [1.4]. EX1011, [0102]-[0105].

As discussed above, Morihiro states that "upon receiving input of a data update request from outside, the control unit 2 executes data update processing in accordance with the flow chart shown in FIG. 10 (a)." Morihiro [0049].



EX1006, Figs. 10(a)-(b)

As shown above, Morihiro discloses editing the pre-overwrite data in volatile memory to produce post-overwrite data – "Next, ***data update processing*** for the current data update request ***is performed on the data in RAM 3*** (Q14)." Morihiro [0050]-[0052]. That is, Morihiro discloses that "the data which has been stored in the volatile memory is updated" thereby producing updated data (i.e. post-overwrite data). EX1006, [0012]; *see also* EX1011, [0103]-[0105].

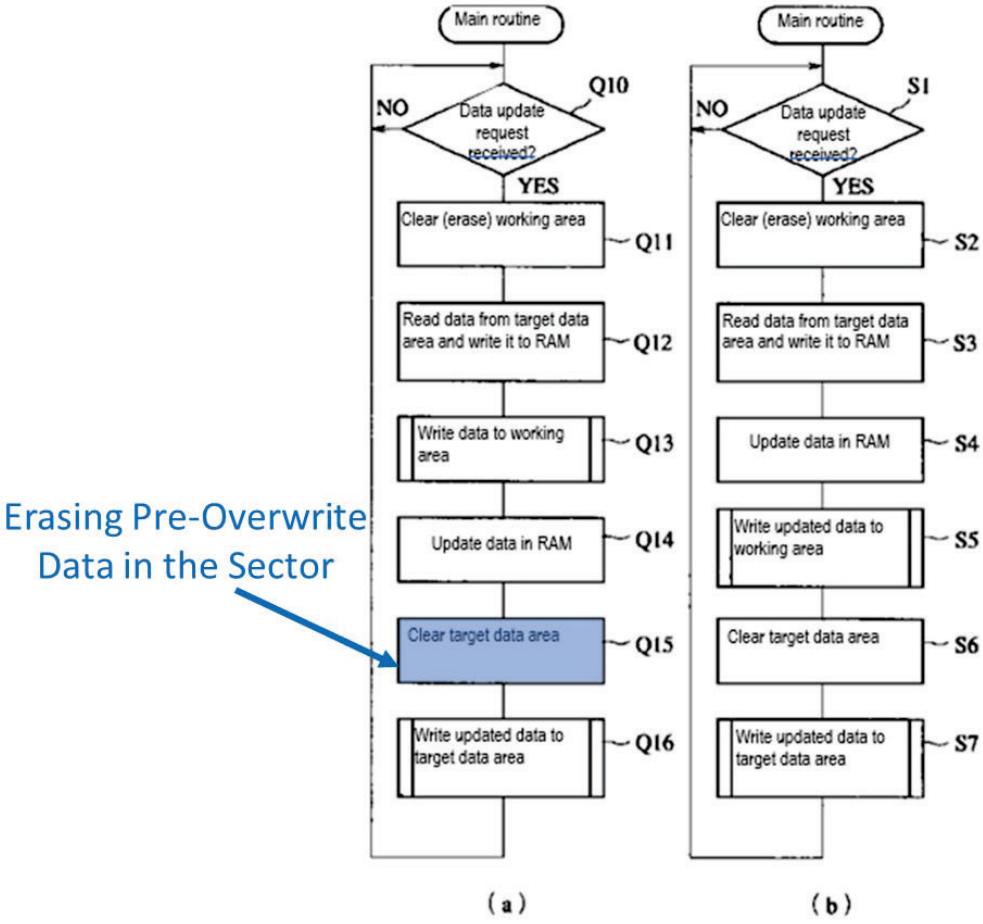
Thus, Morihiro renders obvious "*editing said pre-overwrite data stored in said volatile memory to produce post-overwrite data*" as claimed.

[1.5] ***erasing said pre-overwrite data from said one sector;***

Morihiro teaches claim element [1.5]. EX1011, [0106]-[0109].

Morihiro states that "upon receiving input of a data update request from outside, the control unit 2 executes data update processing in accordance with the flow chart shown in FIG. 10 (a)." Morihiro [0049].

[FIG. 10]



EX1006, Figs. 10(a)-(b)

As shown above, Morihiro discloses erasing the pre-overwrite data in the sector (i.e. data area 10) – "***After clearing the target data area 10*** of the flash memory 4 (Q15), ***the updated data*** is written to the target data area 10 (Q16)." Morihiro [0052] (emphasis added). That is, Morihiro discloses that "the stored content of the data area is cleared . . . ." Morihiro [0012]; *see also* EX1011, [0103]-[0105].

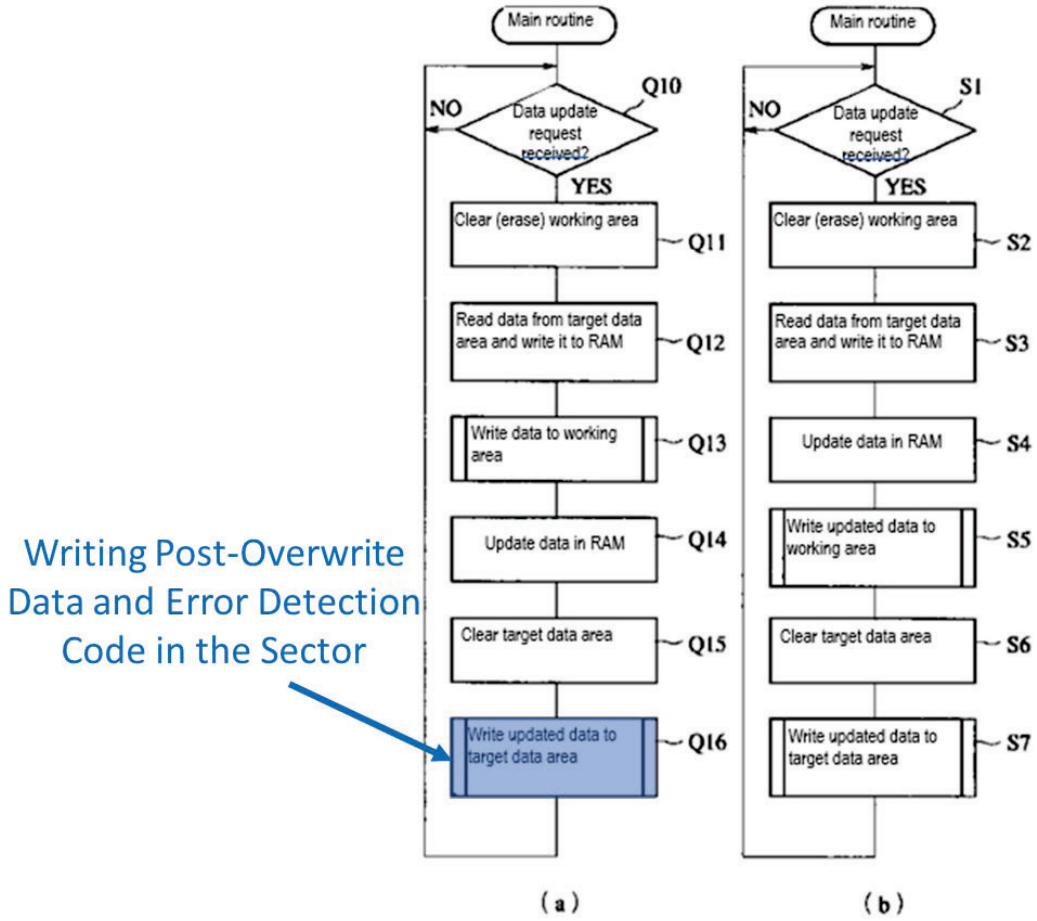
Thus, Morihiro renders obvious "*erasing said pre-overwrite data from said one sector"* as claimed.

[1.6] ***then, writing in said one sector said post-overwrite data stored in said volatile memory together with an error detection code for said post-overwrite data; and***

Morihiro teaches claim element [1.6]. EX1011, [0110]-[0114].

As discussed above, Morihiro states that "upon receiving input of a data update request from outside, the control unit 2 executes data update processing in accordance with the flow chart shown in FIG. 10 (a)." Morihiro [0049]; *see also* EX1011, [0111]-[0112].

[FIG. 10]

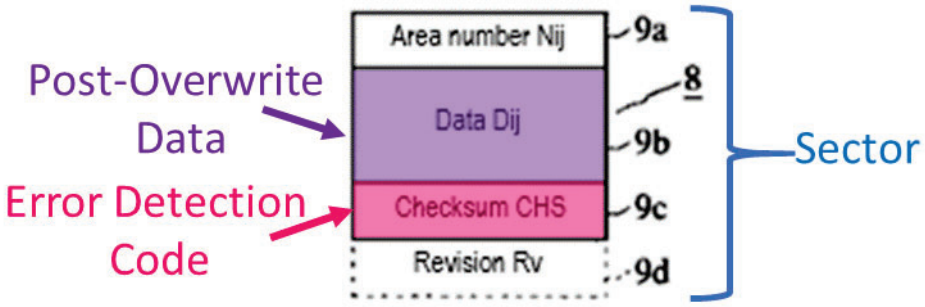


EX1006, Figs. 10(a)-(b)

As shown above, Morihiro discloses writing post-overwrite data (i.e. updated data) in the previously erased data area 10 – "***After clearing the target data area 10*** of the flash memory 4 (Q15), ***the updated data is written to the target data area 10*** (Q16)." Morihiro [0052] (emphasis added). That is, Morihiro discloses "the stored content of the data area is cleared and the ***updated data is written to the cleared data area.***" Morihiro [0012] (emphasis added).

Still further, Morihiro discloses that "[t]he ***configuration of each of said data areas 10*** follows the configuration of data areas ***shown in FIG. 2***." EX1006, [0044] (emphasis added). In that regard, Morihiro states that "as shown in FIG. 2, ***there is formed . . . an area 9b for storing data D_{ij}, and an area 9c for storing a checksum CHS for checking if the stored data D_{ij} is normal or not***." EX1006, [0022] (emphasis added). That is, Morihiro teaches that when "updated data is written to the cleared data area" that the process includes writing the post-overwrite data along with an error detection code, namely a checksum CHS. Morihiro [0012]; *see also* EX1011, [0111]-[0114].

[FIG. 2]



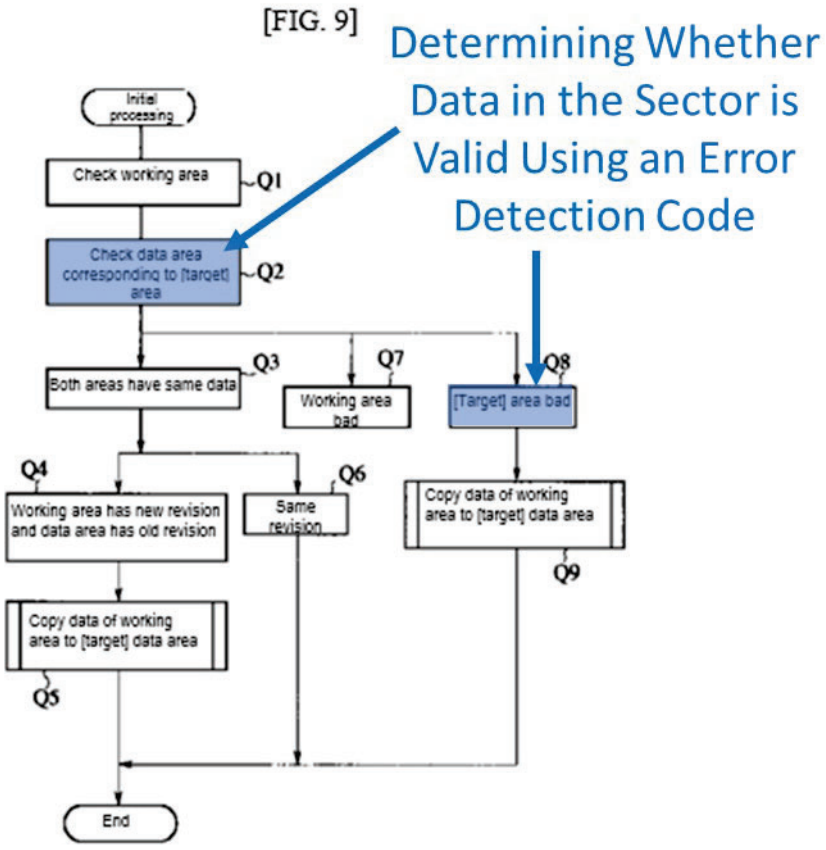
EX1006, Fig. 2

Thus, Morihiro renders obvious "*then, writing in said one sector said post-overwrite data stored in said volatile memory together with an error detection code for said post-overwrite data"* as claimed.

[1.7] ***upon restoration of power, determining whether the data in said one sector is valid using said error detection code.***

Morihiro teaches claim element [1.7]. EX1011, [0115]-[0120].

As discussed above, Morihiro discloses that data area 10 includes a checksum CHS for checking if the stored data is normal or not. EX1006, [0022] and [0044]; *see also* EX1011, [0116]. Morihiro further states "***control unit 2, responding to*** . . . ***restoration of electric service***, executes initialization processing of each data area 10 of the flash memory 4 shown in FIG. 9." Morihiro [0044] (emphasis added).



EX1006, Fig. 9

As shown above, upon restoration of power, an error check is performed on the data stored in data area 10:

> ***A data area 10 is specified*** on the basis of the read area number Nj, and the data Dj stored in this data area 10 is stored ***(Q2). An error check is performed at the same time***.

Morihiro [0045] (emphasis added). Because Morihiro teaches that data area 10 includes a checksum CHS for checking if the stored data is normal or not, it would have been obvious to a POSITA to check the data area 10 in step Q2 with the stored checksum CHS. EX1006, [0022], [0043] and [0044]; *see also* EX1011, [0116]-[0120]. It was well known in the art to utilize the checksum associated with the data to check for errors in the data. *Id*.; s*ee also* EX1007, 5:21-27.

Thus, Morihiro renders obvious "*upon restoration of power, determining whether the data in said one sector is valid using said error detection code*" as claimed.

3. Claim 2

[2.0] ***The method of overwriting data in nonvolatile memory as set forth in claim 1***

See analysis of claim 1. EX1011, [0121].

[2.1] ***wherein said nonvolatile memory is flash memory.***

Morihiro teaches claim element [2.1]. EX1011, [0122]-[0124].

Morihiro discloses that "data areas 10 . . . ***are formed in the flash memory 4***" and "working area 11 . . . ***is provided within this flash memory 4***." Morihiro [0042] (emphasis added).

Thus, Morihiro renders obvious "*wherein said nonvolatile memory is flash memory"* as claimed.

4. Claim 6

[6.0] ***A method of overwriting nonvolatile memory data in which data of interest is stored sector by sector, comprising, when overwriting data of interest, the steps of:***

See analysis of claim element [1.0]. EX1011, [0073]-[0082] and [0125].

[6.1] ***providing a nonvolatile memory control apparatus that includes a non-volatile backup region having at least the same memory capacity as one sector in which data to be overwritten is stored;***

See analysis of claim element [1.1]. EX1011, [0083]-[0091] and [0126]-[0129].

[6.2] ***writing in said backup region pre-overwrite data stored in the sector in which data to be overwritten is stored along with an error detection code for said pre-overwrite data;***

See analysis of claim element [1.2]. EX1011, [0092]-[0096] and [0130].

[6.3] ***copying said pre-overwrite data to a volatile memory;***

See analysis of claim element [1.3]. EX1011, [0097]-[0101] and [0131].

[6.4] ***editing said pre-overwrite data stored in said volatile memory to produce post-overwrite data;***

See analysis of claim element [1.4]. EX1011, [0102]-[0105] and [0132].

[6.5] ***erasing said data to be overwritten from said one sector;***

See analysis of claim element [1.5]. EX1011, [0106]-[0109] and [0133].

[6.6] ***then writing in said one sector said post-overwrite data stored in said volatile memory together with an error detection code for said post-overwrite data; and;***

See analysis of claim element [1.6]. EX1011, [0110]-[0114] and [0134].

[6.7] ***upon restoration of power, determining whether the data in the sector of interest is valid using said error detection code.***

See analysis of claim element [1.7]. EX1011, [0115]-[0120] and [0135].

5. Claim 7

[7.0] ***A method of overwriting pre-overwrite data stored in a nonvolatile memory with post-overwrite data in which data is stored sector by sector, comprising:***

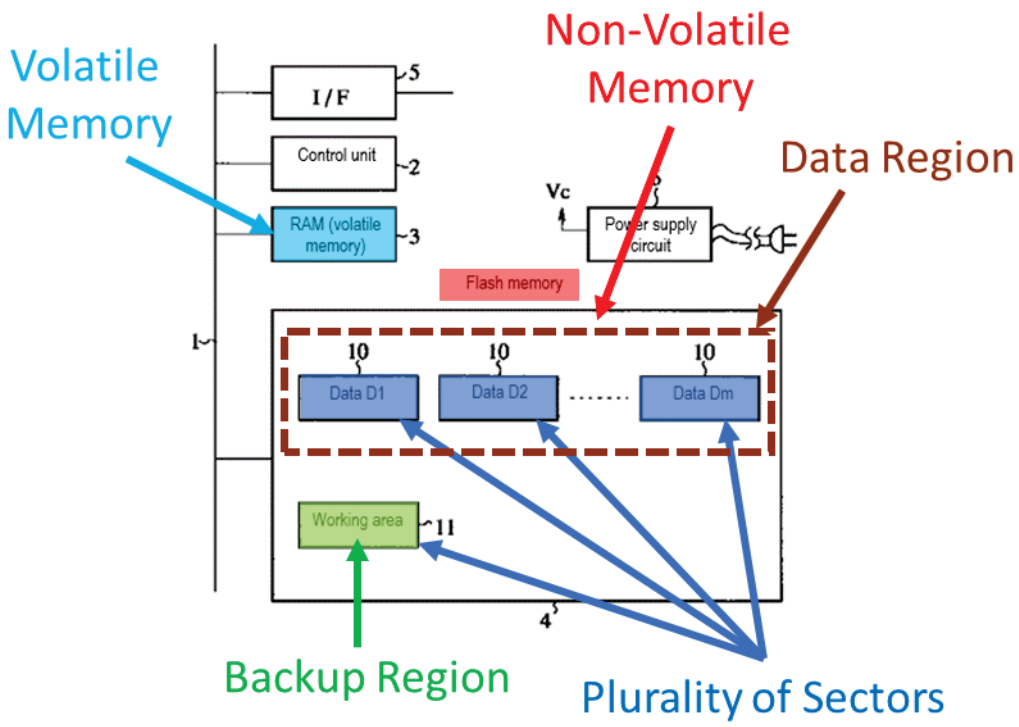
See analysis of claim element [1.0]. EX1011, [0073]-[0082] and [0136].

[7.1] ***providing in a non-volatile memory a backup region and a data region having a plurality of data sectors in which a plurality of pre-overwrite data are stored, the backup region and data region being both provided in the non-volatile memory;***

See analysis of claim element [1.1]. EX1011, [0083]-[0091] and [0137]-[0140].

Additionally, as shown below, Morihiro teaches that working area 11 and data areas 10 form a backup region and a data region having a plurality of data sectors. EX1011, [0138]-[0140].

[FIG. 7]



EX1006, Fig. 7

Thus, Morihiro renders obvious "*providing in a non-volatile memory a backup region and a data region having a plurality of data sectors in which a plurality of pre-overwrite data are stored, the backup region and data region being both provided in the non-volatile memory"* as claimed.

[7.2] ***copying the pre-overwrite data stored in one of the data sectors along with an error detection code for the pre-overwrite data to the backup region;***

See analysis of claim element [1.2]. EX1011, [0092]-[0096] and [0141].

[7.3] ***erasing the pre-overwrite data stored in the one data sector; and***

See analysis of claim element [1.5]. EX1011, [0106]-[0109] and [0142].

[7.4] ***writing the post-overwrite data in the one data sector where the pre-overwrite data was stored together with an error detection code for the post-overwrite data.***

See analysis of claim element [1.6]. EX1011, [0110]-[0114] and [0143].

6. Claim 8

[8.0] ***The method according to claim 7,***

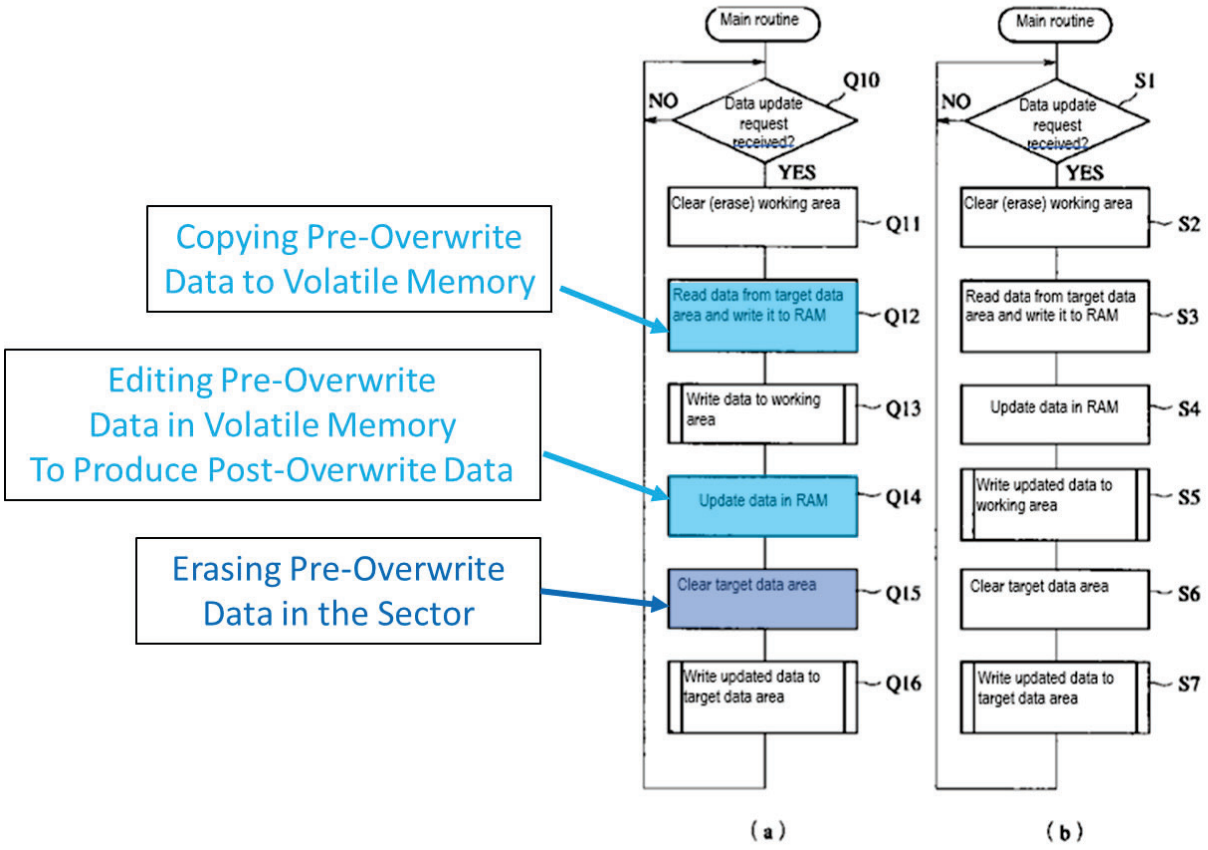
See analysis of claim 7. EX1011, [0144].

[8.1] ***prior to the step of erasing, further comprising: copying the pre-overwrite data to a volatile memory; and editing the copied pre-overwrite data in the volatile memory to produce the post-overwrite data.***

See analysis of claim elements [1.3] and [1.4]. EX1011, [0097]-[0105] and [0145]-[0149].

As shown, Morihiro discloses copying the pre-overwrite data to volatile memory (step Q12) and editing the pre-overwrite data in volatile memory to produce post-overwrite data (step Q14) ***occurs prior to*** erasing the pre-overwrite data in the targeted data area 10 (step Q15). *See* EX1006 [0050]-[0052]; *see also* EX1011, [0146]-[0149].

[FIG. 10]



EX1006, Figs. 10(a)-(b)

Thus, Morihiro renders obvious "*prior to the step of erasing, further comprising: copying the pre-overwrite data to a volatile memory; and editing the copied pre-overwrite data in the volatile memory to produce the post-overwrite data*" as claimed.

7. Claim 9

[9.0] ***The method according to claim 7, further comprising:***

See analysis of claim 7. EX1011, [0150].

[9.1] ***upon restoration of power, determining whether the data in the one data sector is valid using the error detection code associated therewith.***

See analysis of claim element [1.7]. EX1011, [0115]-[0120] and [0151].

8. Claim 10

[10.0] ***The method according to claim 9, further comprising:***

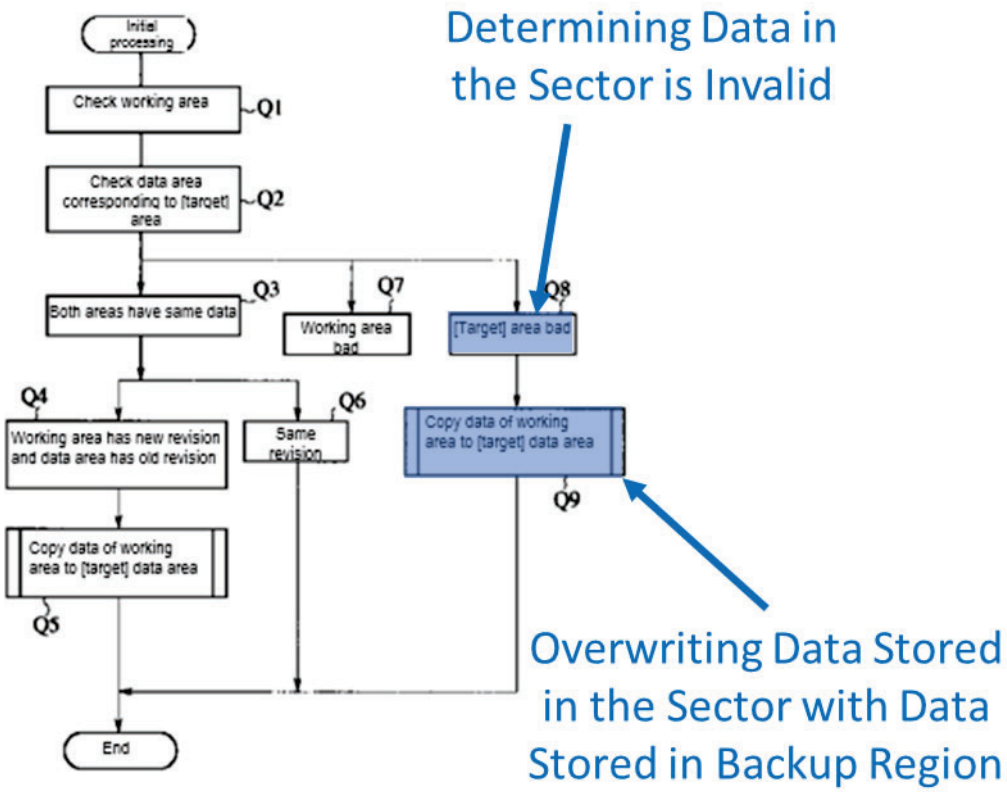
See analysis of claim 9. EX1011, [0152].

[10.1] ***overwriting the data stored in the one data sector with the data stored in the backup region if the data in the one data sector is determined to be invalid.***

Morihiro teaches claim element [10.1]. EX1011, [0153]-[0159].

Morihiro states that upon restoration of power, the control unit 2 performs the processing steps shown below in Fig. 9. EX1006, [0044]. In that regard, Morihiro discloses that if an error is discovered with the stored data in data area 10 (step Q8) then the stored data of working area 11 is written to data area 10 (step Q9):

[FIG. 9]



EX1006, Fig. 9

> In the error check of Q2, ***if an error is discovered in the data D of the target data area 10 (Q8)***, ***the data D of the working area 11 is written (copied) to the data area 10 (Q9)***."

Morihiro [0048] (emphasis added); *see also* EX1011, [0154]-[0159]. And a POSITA understood that flash memory requires erasure of the data being stored in a data area (i.e. sector) prior to updated data being written thereto as evidenced by the '015 Patent and Morihiro. EX1011, [0157]. A POSITA would have understood this erasure and then writing of updated data is considered the

"overwriting" of data in the '015 Patent. *Id.*; *see* also EX1001, 1:33-44, EX1006, [0005]-[0007] and EX1005, [0002]-[0003]. Thus, a POSITA would have understood that Morihiro teaches overwriting the data stored in the data area 10 by writing the data of working area 11 to the data area 10 when an error is discovered therein. *Id*.

Thus, Morihiro renders obvious "*overwriting the data stored in the one data sector with the data stored in the backup region if the data in the one data sector is determined to be invalid"* as claimed.

9. Claim 11

[11.0] ***The method according to claim 10,***

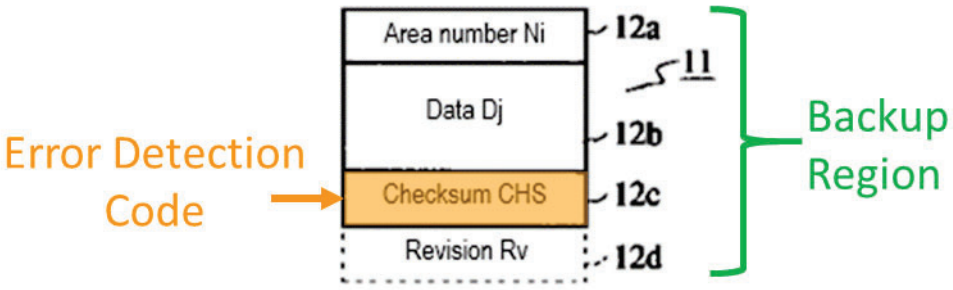
See analysis of claim 10. EX1011, [0160].

[11.1] ***prior to the step of overwriting the data stored in the one data sector, further comprising: if the data in the one data sector is determined to be invalid, determining whether the data stored in the backup region is valid using the error detection code associated therewith.***

Morihiro teaches claim element [11.1]. EX1011, [0161]-[0166].

Morihiro discloses that working area 11 includes "an area 12b for storing data D_j, and ***an area 12c for storing a checksum CHS for checking if the stored data D_j is normal or not***." EX1006, [0043] (emphasis added); *see also* EX1011, [0162].

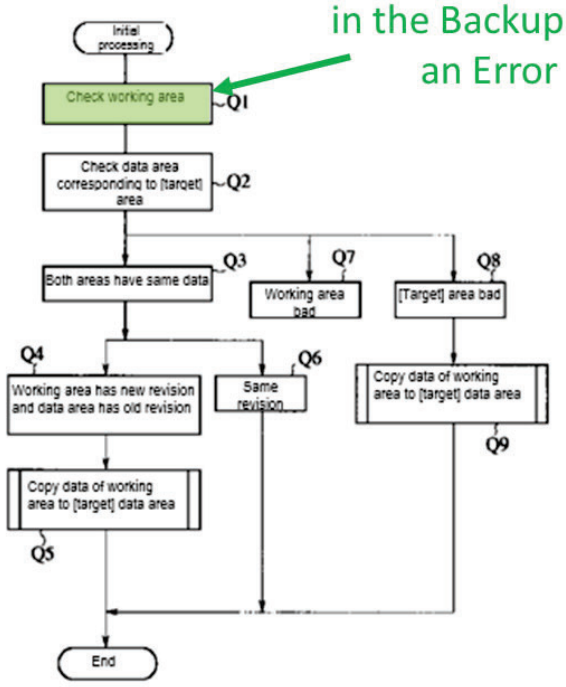
[FIG. 8]



EX1006, Fig. 8

Still further, Morihiro states that "***control unit 2, responding to*** . . . ***restoration of electric service***, executes initialization processing of each data area 10 of the flash memory 4 shown in FIG. 9." Morihiro [0044] (emphasis added); *see also* EX1011, [0163].

[FIG. 9] Determining Whether Data Stored in the Backup Region is Valid Using an Error Detection Code



EX1006, Fig. 9

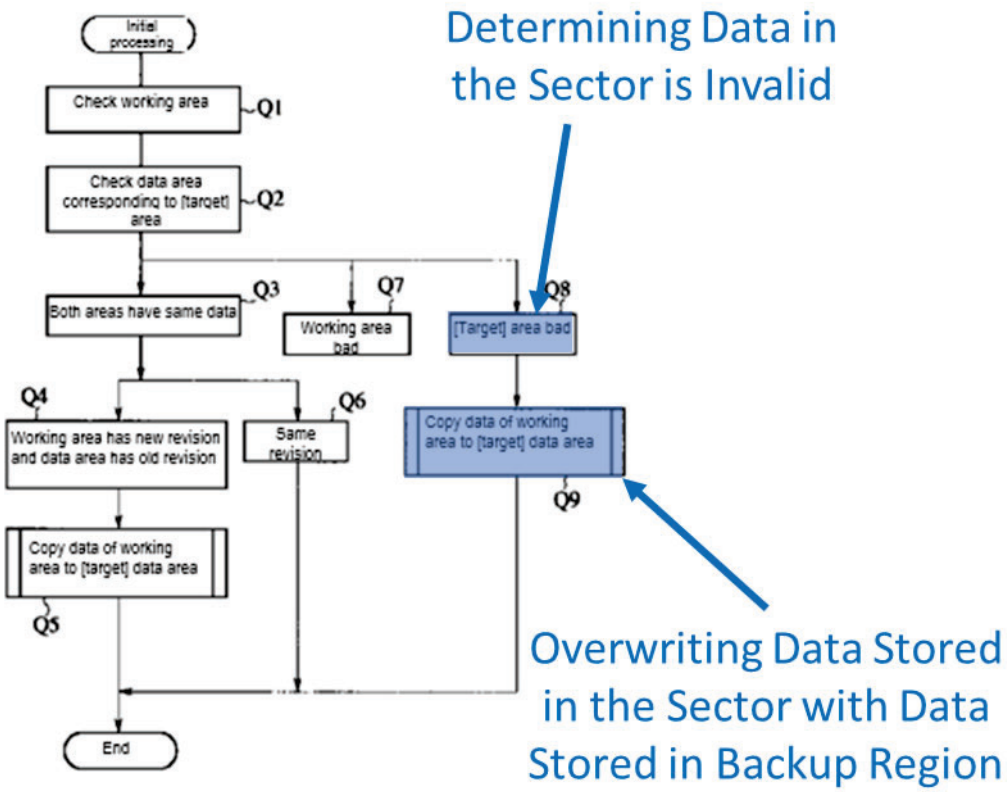
As shown above, upon restoration of power, an error check is performed on the data stored in working area 11:

> When the flow is started, in Q1, the data D_j of area 12b of the working area 11 of flash memory 4 and the data number N_j of area 12a are read (Q1). ***An error check is performed at the same time.***

Morihiro [0045] (emphasis added). Because Morihiro teaches that working area 11 includes a checksum CHS for checking if the stored data is normal or not, it would have been obvious to a POSITA to check the working area 11 in step Q1 with the stored checksum CHS. EX1011, [0162]-[0164]; *see also* EX1006, [0043]. It was well known in the art to utilize the checksum associated with the data to check for errors in the data. *See* EX1007, 5:21-27; *see also* EX1011, [00164].

Moreover, Morihiro discloses that if an error is discovered in data area 10 (step Q8) then the stored data of the working area 11 is written to the data area 10 (step Q9):

[FIG. 9]



EX1006, Fig. 9

> In the error check of Q2, ***if an error is discovered in the data D of the target data area 10 (Q8)***, ***the data D of the working area 11 is written (copied) to the data area 10 (Q9)***."

Morihiro [0048] (emphasis added). That is, Morihiro teaches determining whether the data stored in the working area 11 is valid using the checksum CHS associated therewith prior to writing the stored data of the working area 11 to data area 10 if the data in data area 10 is determined to be invalid. EX1011, [0162]-[0166].

Thus, Morihiro renders obvious "*prior to the step of overwriting the data stored in the one data sector, further comprising: if the data in the one data sector is determined to be invalid, determining whether the data stored in the backup region is valid using the error detection code associated therewith"* as claimed.

B. Ground #2: Claims 3-5 are unpatentable under 35 U.S.C. §103 as being obvious over Morihiro and Forsman.

1. Summary of Forsman

Forsman "relates generally to an improved flash memory design and in particular to a method and an apparatus for recovery using a flash memory system." EX1007, 1:10-12; *see also* EX1011, [0167]-[0171]. More specifically, Forsman states that "the present invention provides a method and an apparatus for fail-safe flash memory recovery with minimal redundancy." EX1007, 1:12-15.

Like Morihiro and the '015 Patent, Forsman recognized that data in flash memory may be corrupted by power failures – "If a major system error, such as a power failure, occurs during the update process, the flash memory can be corrupted." EX1007, 1:24-26; *see also* EX1011, [0169]. As a result, Forsman states that "it is important that there be a mechanism to recover the contents of the flash memory firmware in the event of corruption during update . . . ." EX1007, 1:26-28.

In addressing this issue, Forsman teaches that "a cyclic redundancy check

(CRC), is used to detect whether a block of code has been corrupted or not." EX1007, 5:19-21. Moreover, Forsman recognized that CRC is a form of a checksum used to detect corruption of data – "CRC is a technique where a 'checksum' is appended to the end of a block of data that is being checked for possible corruption." EX1007, 5:21-23. In that regard, Forsman teaches that "[a] new checksum is calculated based on the data received and compared with the checksum appended to the data." EX1007, 5:23-25. Furthermore, Forsman teaches the comparison of these checksums allows for checking data corruption – "[i]f the two values agree, it is highly likely that the data has not been corrupted." EX1007, 5:25-27; *see also* EX1011, [0170].

Accordingly, Forsman teaches that it was well known in the art that data within flash memory may be corrupted by power failures and to use a checksum, such a cyclic redundancy check (CRC), to detect corruption of data in flash memory. EX1011, [0168]-[0171].

2. Reasons to combine Morihiro and Forsman

A POSITA would have found it obvious to implement the checksum CRC technique as taught by Forsman in Morihiro's method that already uses checksums in data area 10 and working area 11. EX1011, [0172]. As discussed above, both Morihiro and Forsman recognized that data in flash memory may be corrupted by

power failures. *See* EX1006, [0008]; *see also* EX1007, 1:24-26. In addressing this issue, Morihiro teaches that within data area 10 and working area 11 "there is formed . . . an area . . . for storing a checksum CHS for checking if the stored data . . . is normal or not." EX1006, [0022], [0043] and [0044]. Forsman further recognized that a CRC is a form of a "checksum" used for detecting data corruption in a flash memory device. *See* EX1007, 5:21-23. That is, like Morihiro, Forsman discloses a checksum technique, namely CRC, that is used to check for data corruption in flash memory. EX1011, [0172]. Implementing Forsman's checksum CRC technique in Morihiro's method that already uses checksums in data areas 10 and working area 11 yields a predictable result, namely, a known technique for checking data corruption. *Id*.

Still further, a POSITA would have found it a simple substitution of one known element, namely Morihiro's checksum, for another known element, namely Forsman's checksum CRC technique, to obtain the predictable result of detecting data corruption in a flash memory. *Id*. at [0173]. That is, it would have been a mere design choice in selecting what form of checksum to implement in Morihiro's method. *Id*. A POSITA would have found it an obvious design choice to implement a known checksum technique, such as Forsman's CRC technique, in Morihiro's method that already includes using a checksum in data area 10 and working area 11. *Id*. A POSITA would have viewed this as a simple substitution

of known checksum techniques that still yields the predictable result of checking for data corruption in flash memory. *Id*.

Additionally, a POSITA would have had a reasonable expectation of success of implementing the checksum CRC technique as taught by Forsman in Morihiro's method that already uses checksums in data area 10 and working area 11. *Id*. at [0174]. In that regard, both Morihiro and Forsman recognized that data in flash memory may be corrupted by power failures. *See* EX1006, Abstract; *see also* EX1007, 1:24-26. In addressing this issue, both Morihiro and Forsman disclose using checksums to detect corruption of data in flash memory. *See* EX1006, [0022], [0043] and [0044]; *see also* EX1007, 5:21-23. Thus, both Morihiro and Forsman recognized that it was known in the art that data within flash memory may be corrupted by power failures and to use a checksum to detect corruption of data in flash memory. EX1011, [0174]. Accordingly, because Morihiro and Forsman recognized similar problems (i.e. flash memory corruption via power failure) and provide similar solutions (i.e. checksums for detecting data corruption) a POSITA would have had a reasonable expectation of success of implementing Forsman's checksum CRC technique in Morihiro's method that already uses checksums in data area 10 and working area 11. *Id*.

3. Claim 3

[3.0] ***The method of overwriting data in nonvolatile memory as set forth in claim 1,***

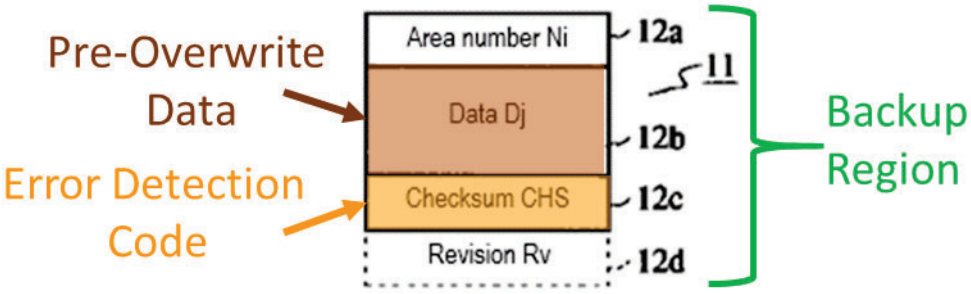
See analysis of claim element [1.0]. EX1011, [0175].

[3.1] ***wherein the step of writing in said backup region pre-overwrite data includes calculating a CRC value for said pre-overwrite data and writing the calculated CRC value in said backup region.***

Morihiro and Forsman teach claim element [3.1]. EX1011, [0176]-[0184].

Morihiro discloses that data to be updated (i.e. pre-overwrite data) is first written into working area 11 – "***when a data update request is issued, the data to be updated is copied first into the working area 11*** . . . ." Morihiro [0053]-[0054] (emphasis added). Additionally, Morihiro discloses that "[w]ithin the working area 11, as shown in FIG. 8, there is formed . . . an area 12b for storing data $D_{j,}$ and an area 12c for storing a checksum CHS for checking if the stored data D_j is normal or not." EX1006, [0043]. And, as discussed above, Morihiro's method teaches that "the working area 11 is cleared (Q11)" prior to "[t]he data . . . [being] written to the working area 11 of the flash memory 4" at Q13. EX1006, [0050]-[0051]. Thus, when "the data to be updated is copied first into the working area 11" the process includes writing the pre-overwrite data along with an error detection code, namely a checksum CHS, to the previously cleared working area 11. EX1006, [0043] and [0050]-[0054]; *see also* EX1011 [0177]-[0178].

[FIG. 8]



EX1006, Fig. 8

As discussed above, it would have been obvious to implement the checksum CRC technique as taught by Forsman in Morihiro's method that already uses a checksum in working area 11. EX1011, [0172]-[0174]. As evidenced by Yu, a POSITA would have understood that in order to ***form*** an area for storing a CRC checksum that one must also calculate the CRC checksum (i.e. the CRC value) in addition to writing the checksum:

> ***The use of a Cyclic Redundancy Check (CRC) to verify the integrity of data is well known in the art***. . . . upon receiving the data and the transmitted CRC value, ***computes a new CRC value on the received data using the same CRC algorithm as used by the source***. The new CRC value computed on the received data is compared to the transmitted CRC value computed and transmitted by the source along with the data. ***If the two CRC values match, the data was transmitted and received without corruption***.

EX1008, 1:11-24 (emphasis added). Accordingly, a POSITA would have understood that the combination of Morihiro and Forsman teaches calculating a CRC value for pre-overwrite data and writing the calculated CRC value in working area 11 during the writing of the pre-overwrite data in working area 11. EX1011, [0179]-[0180].

Thus, for at least these reasons, the combination of Morihiro and Forsman renders obvious "wherein the step of writing in said backup region pre-overwrite data includes calculating a CRC value for said pre-overwrite data and writing the calculated CRC value in said backup region" as claimed.

To the extent it is argued that Morihiro just copies the CRC value of the data stored in data area 10 into working area 11, it would have been obvious to a POSITA to calculate a new checksum for such data as taught by Forsman. EX1011, [0181]-[0184]. Specifically, Forsman teaches to calculate a new checksum on data received and compare it with the checksum already appended to the received data - "A new checksum is calculated based on the data received and compared with the checksum appended to the data." EX1007, 5:23-25; *see also* EX1008, 1:11-24. Moreover, Forsman teaches the comparison of these checksums allows for checking data corruption – "If the two values agree, it is highly likely that the data has not been corrupted." EX1007, 5:25-27. Accordingly, it would have been obvious to a POSITA to calculate a new checksum, as taught by

Forsman, as part of copying the data stored in data area 10 into working area 11, as taught by Morihiro, to ensure the received data was not corrupted during the copying thereof, as taught by Forsman. EX1011, [0181].

Still further, as discussed above, Morihiro teaches that when "the data to be updated is copied first into the working area 11" that the process includes writing the pre-overwrite data along with a checksum. EX1006, [0043] and [0050]-[0054]; *see also* EX1011 [0177]-[0178]. That is, Morihiro teaches writing the value of the checksum to working area 11. EX1011, [0182]. And, as discussed above, Forsman teaches to calculate a new checksum on data received and compare it with the checksum already appended to the received data. *See* EX1007, 5:23-25.

Moreover, a POSITA knew that there was no data corruption "[i]f the two CRC values match." EX1011, [0183], *see also* EX1008, 1:11-24 and EX1007, 5:25-27. That is, when the data is received without corruption, the newly calculated checksum on data received and the checksum already appended to the received data have the same CRC value. EX1011, [0183]. As a result, the CRC value of the checksum written to the working area 11, as taught by Morihiro and Forsman, is the same for the newly calculated checksum and the checksum received with the data as taught by Forsman. *Id*. Therefore, under at least this additional scenario, the combination of Morihiro and Forsman additionally teach writing the calculated CRC value to working area 11. EX1011, [0183]-[0184].

Thus, for at least these additional reasons, the combination of Morihiro and Forsman renders obvious "*wherein the step of writing in said backup region pre-overwrite data includes calculating a CRC value for said pre-overwrite data and writing the calculated CRC value in said backup region*" as claimed.

4. Claim 4

[4.0] ***The method of overwriting data in nonvolatile memory as set forth in claim 1,***

See analysis of claim element [1.0]. EX1011, [0185].

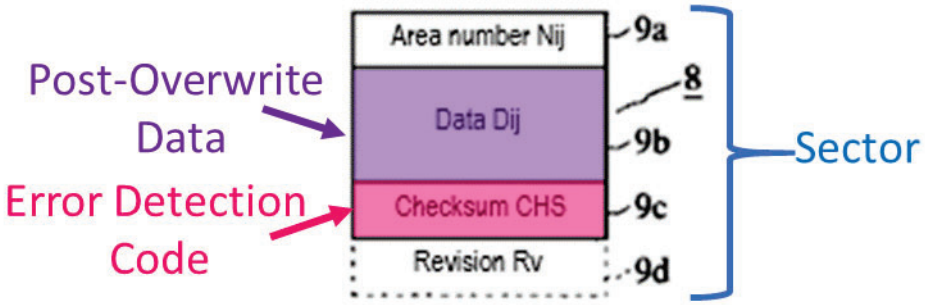
[4.1] ***wherein the step of writing in said one sector said post-overwrite data includes calculating a CRC value for said post-overwrite data and writing the calculated CRC value in said one sector.***

Morihiro and Forsman teach claim element [4.1]. EX1011, [0186]-[0192].

Morihiro discloses that the updated data (i.e. post-overwrite data) is written into the one sector, namely data area 10 – "After clearing the target data area 10 of the flash memory 4 (Q15), ***the updated data is written to the target data area 10*** (Q16)." Morihiro [0052] (emphasis added). Additionally, Morihiro discloses that "***there is formed . . . an area 9b for storing data D_{ij}, and an area 9c for storing a checksum CHS for checking if the stored data D_{ij} is normal or not***" for each data area 10. EX1006, [0022] and [0044] (emphasis added). Accordingly, when "updated data is written to the cleared data area" the process includes writing post-

overwrite data along with a checksum CHS. Morihiro [0012] and [0052]; *see also* EX1011, [0187]-[0188].

[FIG. 2]



EX1006, Fig. 2

As discussed above, it would have been obvious to implement the checksum CRC technique as taught by Forsman in Morihiro's method that already uses a checksum in data area 10. EX1011, [0172]-[0174]. Moreover, as discussed above in claim 3, a POSITA would have understood that in order to ***form*** an area for storing a CRC checksum that one must also calculate the CRC checksum (i.e. the CRC value) in addition to writing the checksum. *See* EX1008, 1:11-24; *see also* EX1011, [0189].

Still further, because Morihiro teaches that the data has been "updated" it would have been obvious to a POSITA to calculate a new checksum based on the updated data as part of Morihiro's writing process. EX1011, [0190]. Specifically, a POSITA knew that the checksum for data is calculated based on the data itself. *See* EX1009, 4:57-5:4; *see also* EX1011, [0190]. As a result, because Morihiro

teaches that the data has been updated (i.e. changed) the previous checksum associated with the data prior to the update is likely no longer providing a valid check for corruption of the data. *Id*. That is, as evidenced by Talagala (EX1009), a POSITA recognized that a checksum is dependent on the data and thereby it would have been obvious to calculate a new checksum (i.e. CRC value) for updated data. *Id*.

Accordingly, a POSITA would have understood that the combination of Morihiro and Forsman teaches calculating a CRC value for post-overwrite data and writing the calculated CRC value in data area 10 during the writing of the post-overwrite data in data area 10. *Id*. at [0187]-[0192].

Thus, the combination of Morihiro and Forsman renders obvious "*wherein the step of writing in said one sector said post-overwrite data includes calculating a CRC value for said post-overwrite data and writing the calculated CRC value in said one sector*" as claimed.

5. Claim 5

[5.0] ***The method according to claim 4, further comprising:***

See analysis of claim element [4.0]. EX1011, [0193].

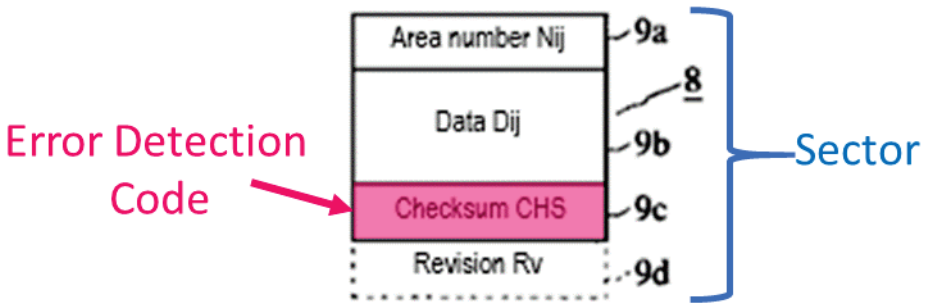
[5.1] ***overwriting the data stored in said one sector with the data stored in said backup region if the data in said one sector is determined to be invalid.***

Morihiro teaches claim element [5.1]. EX1011, [0194]-[0200].

As discussed above, Morihiro discloses that data area 10 includes "an area

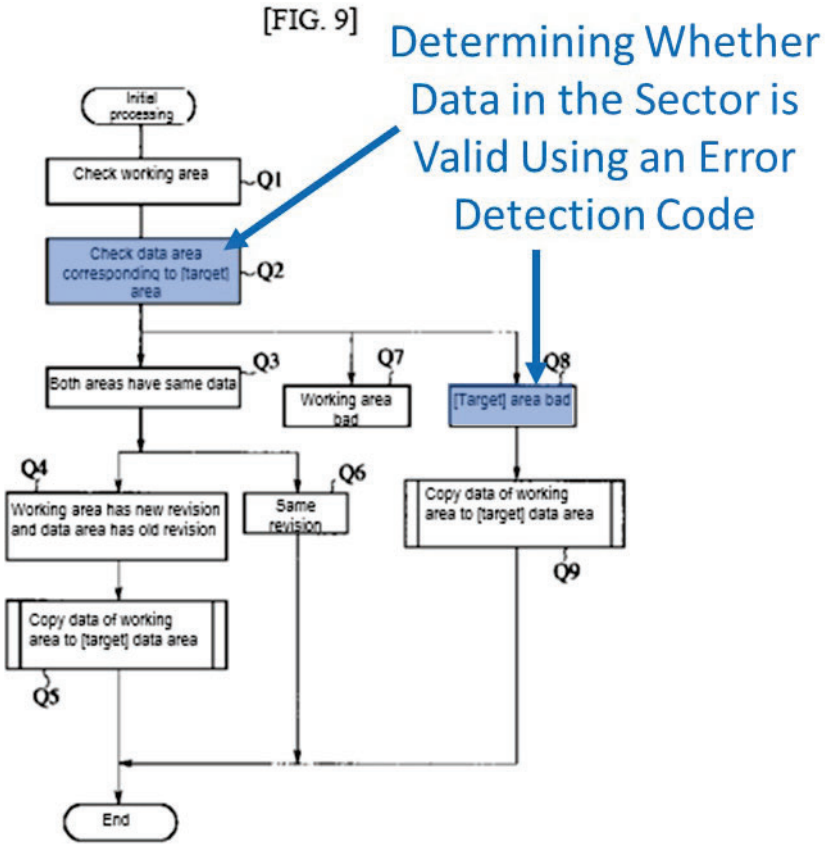
9c ***for storing a checksum CHS for checking if the stored data D_{ij} is normal or not***." EX1006, [0022] and [0044] (emphasis added); *see also* EX1011, [0195]-[0197].

[FIG. 2]



EX1006, Fig. 2

Still further, Morihiro states "***control unit 2, responding to*** . . . ***restoration of electric service***, executes initialization processing of each data area 10 of the flash memory 4 shown in FIG. 9" Morihiro [0044] (emphasis added).



EX1006, Fig. 9

As shown above, upon restoration of power, an error check is performed on the data stored in data area 10:

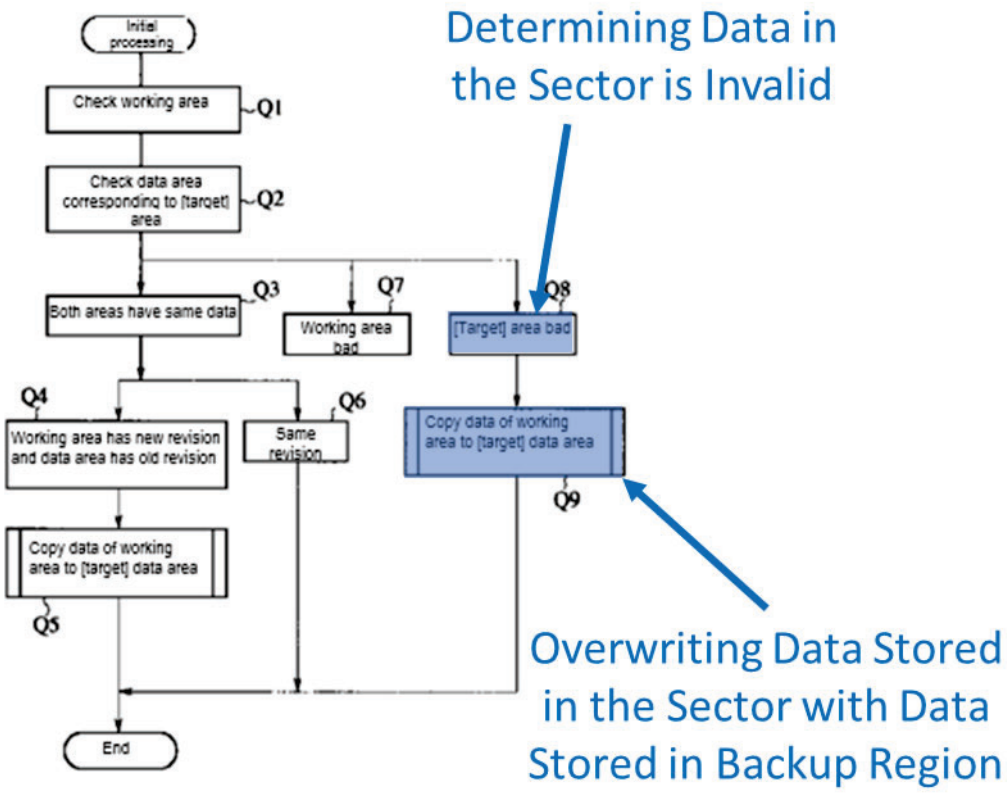
> ***A data area 10 is specified*** on the basis of the read area number N_j, and the data D_j stored in this data area 10 is stored (Q2). ***An error check is performed at the same time***."

Morihiro [0045] (emphasis added). Because data area 10 includes a checksum CHS for checking if the stored data is normal or not, it would have been obvious to a POSITA to check data area 10 in step Q2 with the stored checksum CHS.

EX1006, [0022], [0043] and [0044]; *see also* EX1011, [0197]. It was well known in the art to utilize the checksum associated with the data to check for errors in the data. *See* EX1007, 5:21-27; *see also* EX1011, [0195]-[0197].

Moreover, as shown below, if an error is discovered with the stored data in data area 10 (step Q8) then the stored data of the working area 11 is written to the data area 10 (step Q9):

[FIG. 9]



EX1006, Fig. 9

In the error check of Q2, ***if an error is discovered in the data D of the target data area 10 (Q8), the data D of the***

> ***working area 11 is written (copied) to the data area 10 (Q9)***."

Morihiro [0048] (emphasis added); *see also* EX1011, [0198]. And a POSITA understood that flash memory requires erasure of the data being stored in a data area (i.e. sector) prior to the updated data being written to the previously erased data area (i.e. sector) as evidenced by the '015 Patent and Morihiro. EX1011, [0059]-[0061] and [0198]. A POSITA would have understood this erasure and then writing of updated data is considered the "overwriting" of data in the '015 Patent and Morihiro. *See* EX1011, [0198]; *see also* EX1001, 1:33-44, EX1006, [0005]-[0007] and EX1005, [0002]-[0003]. Thus, a POSITA would have understood that Morihiro teaches overwriting the data stored in the data area 10 by writing the data of working area 11 to the data area 10 when an error is discovered therein. *See* EX1006, [0054]; *see also* EX1011, [0198]-[0200].

Thus, Morihiro renders obvious "*overwriting the data stored in the one data sector with the data stored in the backup region if the data in the one data sector is determined to be invalid"* as claimed.

C. Ground #3: Claim 11 is unpatentable under 35 U.S.C. §103 as being obvious over Morihiro and Mitomi.

1. Summary of Mitomi

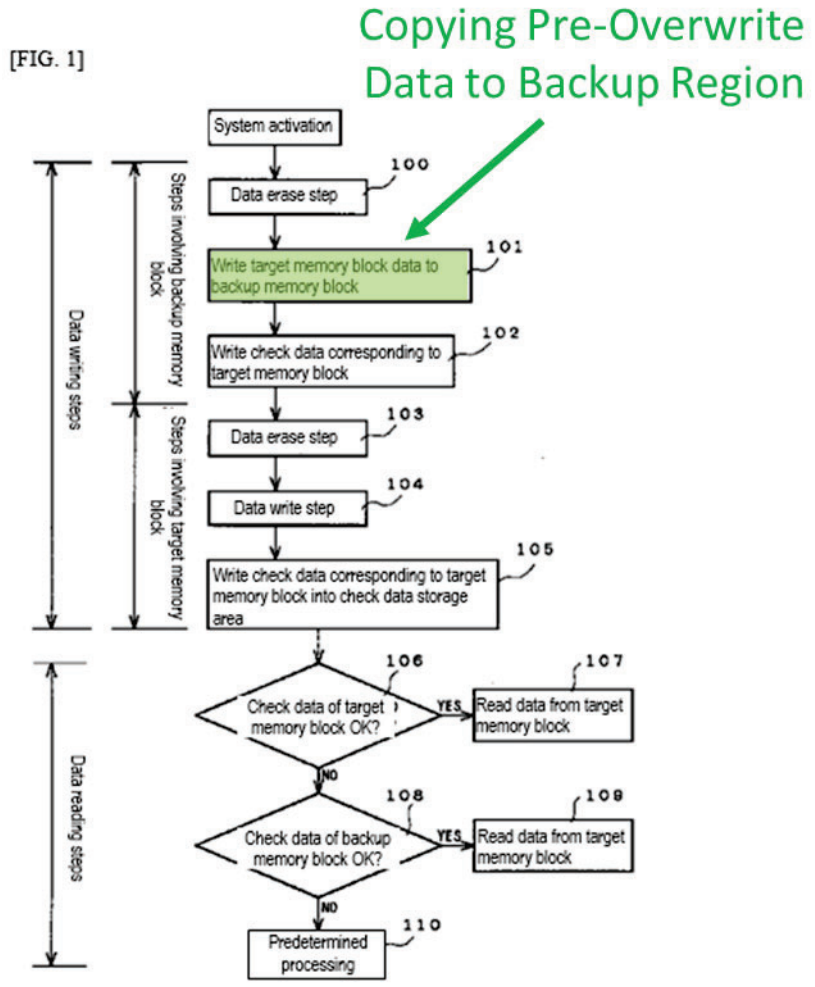
Like the '015 Patent and Morihiro, Mitomi recognized that data loss could occur if power fails between data erasure and completion of writing post-overwrite

data:

> According to the conventional writing method for block erase flash memory, ***since erasure of data is performed at once on one memory block as a minimal unit, prior to completion of a write operation sequence, a state occurs in which neither the old data from before the rewrite nor the new data from after the rewrite is present in the block erase flash memory***, and since such data erase and data write operations (especially data erase) on block erase memory take time, ***if a power supply interruption or the like occurs during such a write operation sequence and the data writing process is interrupted, there is the problem that a state will occur in which a portion of the data will not be present in the block erase flash memory***.

EX1005, [0005] (emphasis added); *see also* EX1011, [0201]-[0208]. Thus, Mitomi is directed to "a writing method for block erase flash memory which solves this problem and makes it possible to prevent data loss even if a data writing process is interrupted." EX1005, [0006].

Mitomi teaches the known solution in the art includes copying pre-overwrite data to a backup region in a flash memory as shown below in Fig. 1:



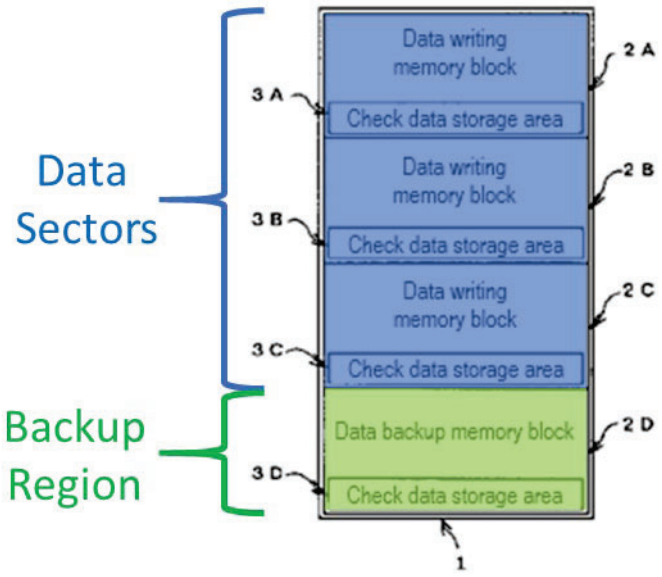
EX1005, Fig. 1

To achieve the aforesaid purpose, in the present invention, ***at least one of the memory blocks making up the block erase flash memory is allocated as a data backup memory block, and when performing a data write, first, all of the data already written to the target memory block to which the data write is to be performed is written to the data backup memory block***, after which writing of the new data to the target memory block is performed.

EX1005, [0007] (emphasis added); *see also* EX1011, [0203].

Specifically, referring to Fig. 2, Mitomi discloses a flash memory where "2A through 2C are data writing memory blocks, 2D is a data backup memory block, and 3A through 3D are check data storage areas." EX1005, [0012]. Thus, Mitomi discloses a flash memory with data areas, namely writing memory blocks 2A-2C and a backup region, namely data backup memory block 2D. EX1011, [0203]-[0205].

[FIG. 2]



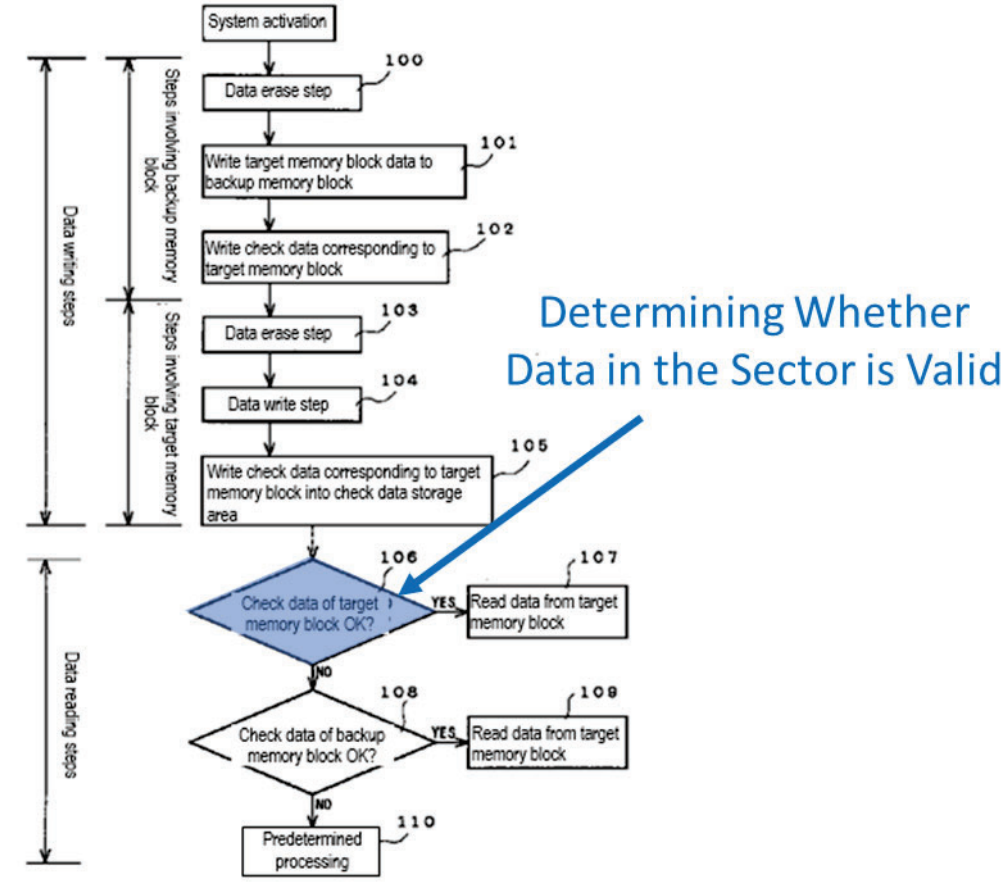
EX1005, Fig. 2

And, as shown above, each of these memory blocks includes a respective check data storage area – "Each of the memory blocks 2A through 2D is provided with a check data storage area 3A through 3D for storing check data." EX1005, [0014]. Mitomi further discloses that "***an error detection/correction code may of***

course be appended to the check data, and error correction processing may be performed during reading of the data." EX1005, [0025]; *see also* EX1011, [0205].

For error correction, Mitomi first teaches determining whether data stored in the data area (i.e. memory block 2A) is valid at step 106 of Fig. 1:

[FIG. 1]



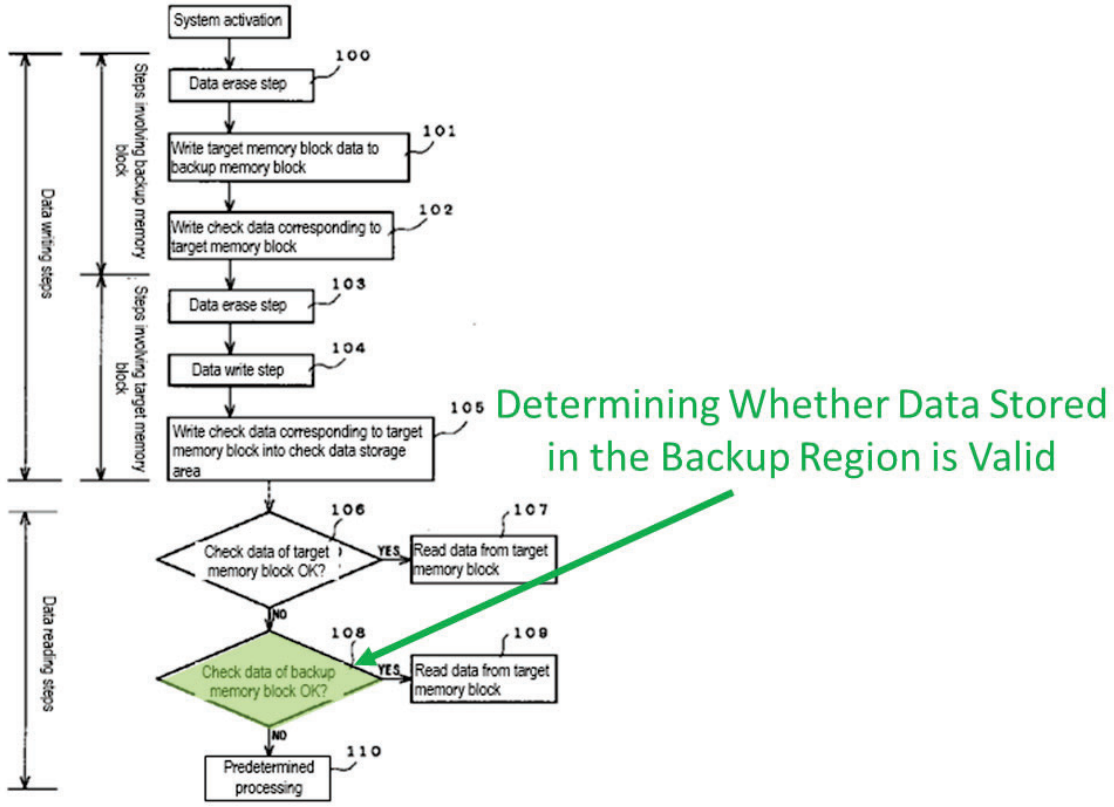
EX1005, Fig. 1

> [T]he check data is read from the check data storage area 3A of this target memory block 2A, and ***it is determined on the basis of this check data if the data which has been written to the target memory block 2A is valid (step 106)***.

EX1005, [0021] (emphasis added); *see also* EX1011, [0206].

As further shown below, when data stored in the data area (i.e. memory block 2A) is determined invalid Mitomi then teaches determining whether the data stored in the backup region (i.e. backup memory block 2D) is valid at step 108:

[FIG. 1]



EX1005, Fig. 1

> [I]t is determined that there is no valid data in the target memory block 2A (step 106), ***and next, it is determined if the data which has been written to the data backup memory block 2D is valid as data of the target memory block*** on the basis of whether or not check data specific to

> the target memory block 2A can be read from the check data storage area 3D (step 108***)***.

EX1005, [0022] (emphasis added); *see also* EX1011, [0207].

Thus, like the '015 Patent and Morihiro, "even if a power supply interruption or the like occurs during the data write operation on the target memory block and the operation is interrupted, the data of the target memory block will have been saved in the data backup memory block." EX1005, [0008]; *see also* EX1011, [0208].

2. Reasons to combine Morihiro and Mitomi.

A POSITA would have found it obvious to implement Mitomi's teachings of checking the validity of data stored in a data area prior to checking the validity of data stored in a backup region in Morihiro's method. EX1011, [0209]. As discussed above, Morihiro already teaches that "[a]n error check is performed" on the data stored in data area 10 and working area 11 upon restoration of power. *See* EX1006, [0044]-[0045]. That is, Morihiro's method already discloses checking the validity of the data stored in both the data area 10 and working area 11. EX1011, [0209]. Mitomi further teaches that it was also known in the art to check the validity of the data stored in a data area prior to checking the validity of the data in a backup region. *See* EX1005, [0021]-[0022]. Implementing Mitomi's validity checking order (i.e. checking the validity of the data area prior to checking

the validity of the backup region) in Morihiro's method that already discloses checking the validity of the stored data in both data area 10 and working area 11 yields a predictable result, namely, a known technique for checking data corruption in both areas. EX1011, [0209].

Still further, a POSITA would have found it a simple substitution of one known order, namely Morihiro's validity checking both data area 10 and working area 11, for another known order, namely Mitomi's teaching of ordering the validity checking (i.e. checking the validity of the data area prior to checking the validity of the backup region). *Id*. at [0210]. That is, it would have been a mere design choice in selecting which validity checking is performed first, either error checking data area 10 or working area 11 in Morihiro's method. *Id*. A POSITA would have found it an obvious design choice to implement a known validity checking order, such as the order disclosed by Mitomi, in Morihiro's method that already includes checking the validity of the data stored in both data area 10 and working area 11. *Id*. A POSITA would have viewed this as a simple substitution of known validity checking orders that still yields the predictable result of checking for data corruption in flash memory. *Id*.

Additionally, a POSITA would have had a reasonable expectation of success of implementing Mitomi's teachings of checking the validity of data stored in a data area prior to checking the validity of data stored in a backup region in

Morihiro's method. *Id*. at [0211]. In that regard, both Morihiro and Mitomi recognized that data in flash memory may be corrupted by power failures. *See* EX1006, Abstract; *see also* EX1005, [0005]. In addressing this issue, both Morihiro and Mitomi disclose checking the validity of data stored in the data area and the backup region. *See* EX1006, [0045]; *see also* EX1005, [0021]-[0022]. Thus, both Morihiro and Mitomi recognized that it was known in the art that data within flash memory may be corrupted by power failures and to check the validity of stored data in the data area and backup region to detect corruption of data in flash memory. EX1011, [0211]. Accordingly, because Morihiro and Mitomi recognized similar problems (i.e. flash memory corruption via power failure) and provide similar solutions (i.e. validity checks on the stored data in the data area and backup region to detect data corruption) a POSITA would have had a reasonable expectation of success of implementing Mitomi's teachings of checking the validity of data stored in a data area prior to checking the validity of data stored in a backup region in Morihiro's method. *Id*.

3. Claim 11

[11.0] ***The method according to claim 10, prior to the step of overwriting the data stored in the one data sector, further comprising:***

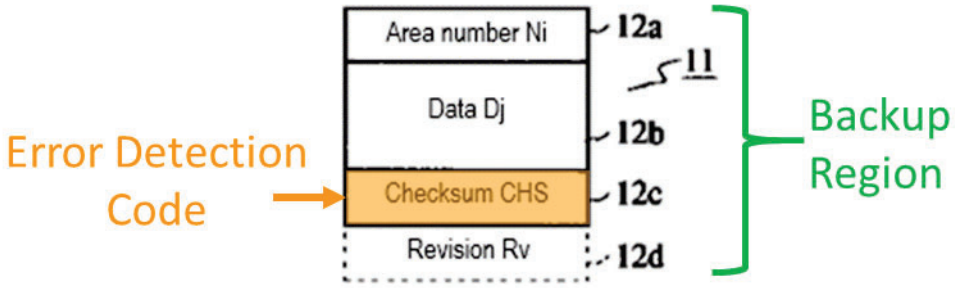
See analysis of claim 10. EX1011, [0212].

[11.1] ***if the data in the one data sector is determined to be invalid, determining whether the data stored in the backup region is valid using the error detection code associated therewith.***

Morihiro and Mitomi teach claim element [11.1]. EX1011, [0213]-[0224].

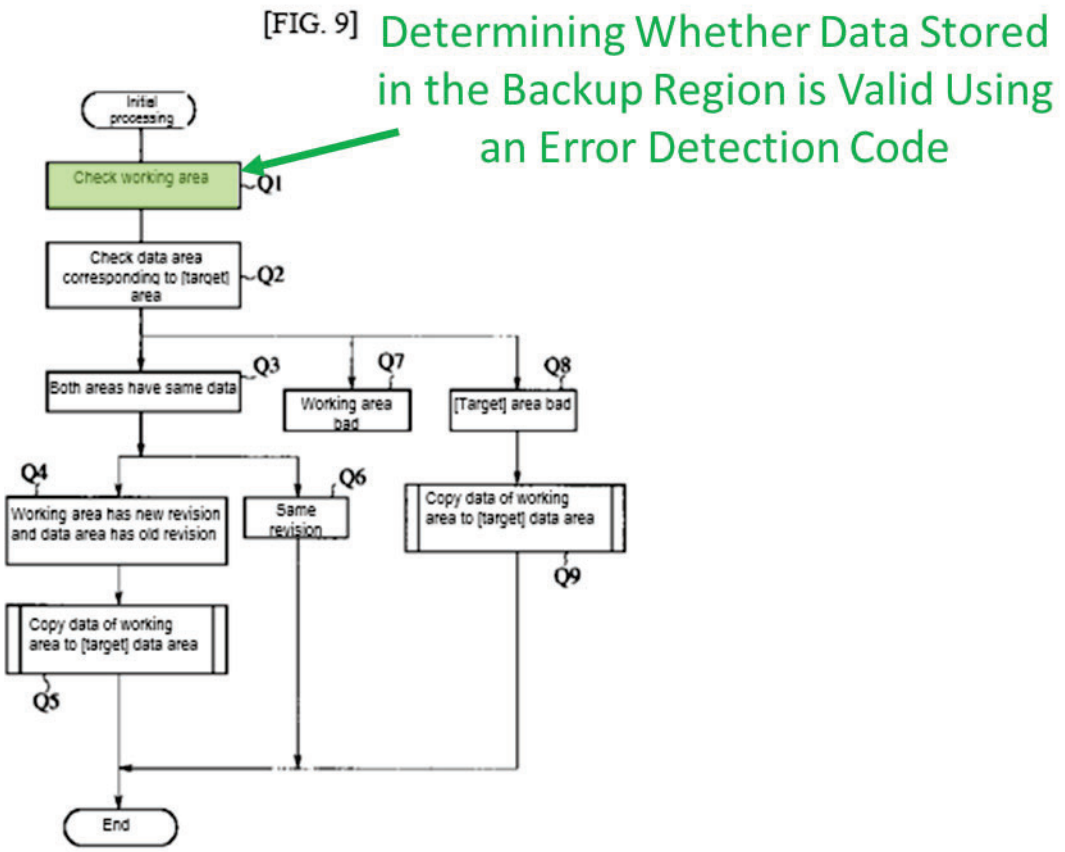
As shown, Morihiro discloses that working area 11 includes "an area 12b for storing data D_j, and ***an area 12c for storing a checksum CHS for checking if the stored data D_j is normal or not***." EX1006, [0043] (emphasis added); *see also* EX1011, [0214].

[FIG. 8]



EX1006, Fig. 8

Additionally, Morihiro states "***control unit 2, responding to . . . restoration of electric service***, executes initialization processing of each data area 10 of the flash memory 4 shown in FIG. 9." Morihiro [0044] (emphasis added).



EX1006, Fig. 9

As shown above, upon restoration of power, an error check is performed on the data stored in working area 11:

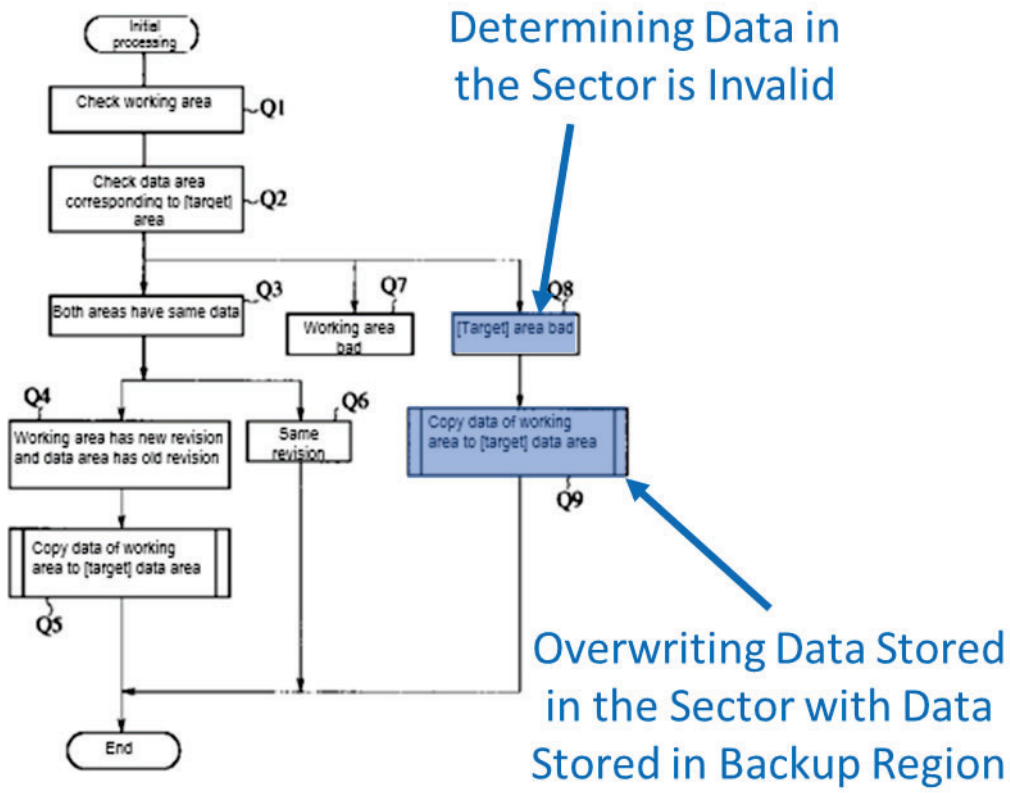
> When the flow is started, in Q1, ***the data D_j of area 12b of the working area 11 of flash memory 4 and the data number N_j of area 12a are read (Q1). An error check is performed at the same time.***

Morihiro [0045] (emphasis added). Because working area 11 includes a checksum CHS for checking if the stored data is normal or not, it would have been obvious to a POSITA to check working area 11 in step Q1 with the stored checksum CHS.

See EX1006, [0043]; *see also* EX1011, [0214]-[0216]. It was well known in the art to utilize the checksum associated with the data to check for errors in the data. *See* EX1007, 5:21-27; *see also* EX1011, [0216].

Moreover, as shown below, if an error is discovered with the stored data in data area 10 (step Q8) then the stored data of the working area 11 is written to the data area 10 (step Q9):

[FIG. 9]



EX1006, Fig. 9

In the error check of Q2, ***if an error is discovered in the data D of the target data area 10 (Q8), the data D of the***

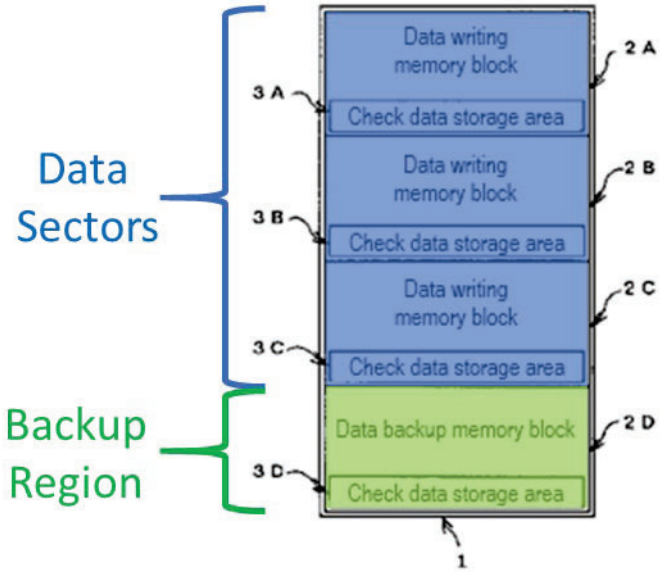
> ***working area 11 is written (copied) to the data area 10 (Q9)***."

Morihiro [0048] (emphasis added). That is, Morihiro teaches determining whether the data stored in the working area 11 is valid using the checksum CHS associated therewith prior to writing the stored data of the working area 11 to data area 10 if the data in data area 10 is invalid. EX1011, [0217].

As discussed above, it would have been obvious to implement Mitomi's teachings of checking the validity of data stored in a data area prior to checking the validity of data stored in a backup region in Morihiro's method. EX1011, [0209]-[0211]. Like the '015 Patent and Morihiro, Mitomi is directed to a method of writing to flash memory that prevents data loss occurring from power failures. *See* EX1005, [0006]; *see also* EX1011, [0218].

As discussed above, in reference to Fig. 2 below, Mitomi discloses a flash memory that includes data areas, namely writing memory blocks 2A-2C and a backup region, namely data backup memory block 2D. *See* EX1005, [0012]. Mitomi further discloses these memory blocks include a respective check data storage area "for storing check data" in which "***an error detection/correction code may of course be appended***." EX1005, [0014] and [0025] (emphasis added); *see also* EX1011, [0219]-[0220].

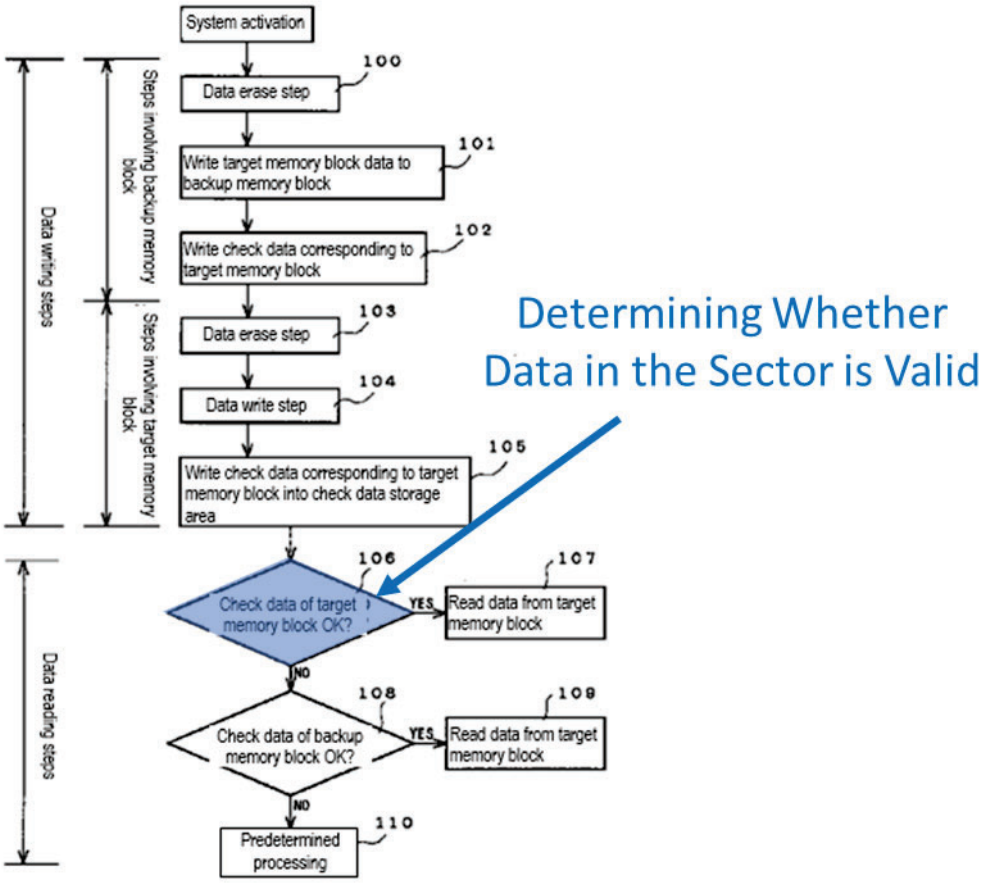
[FIG. 2]



EX1005, Fig. 2

With respect to error correction, Mitomi first teaches determining whether data stored in the data area (i.e. memory block 2A) is valid (step 106):

[FIG. 1]



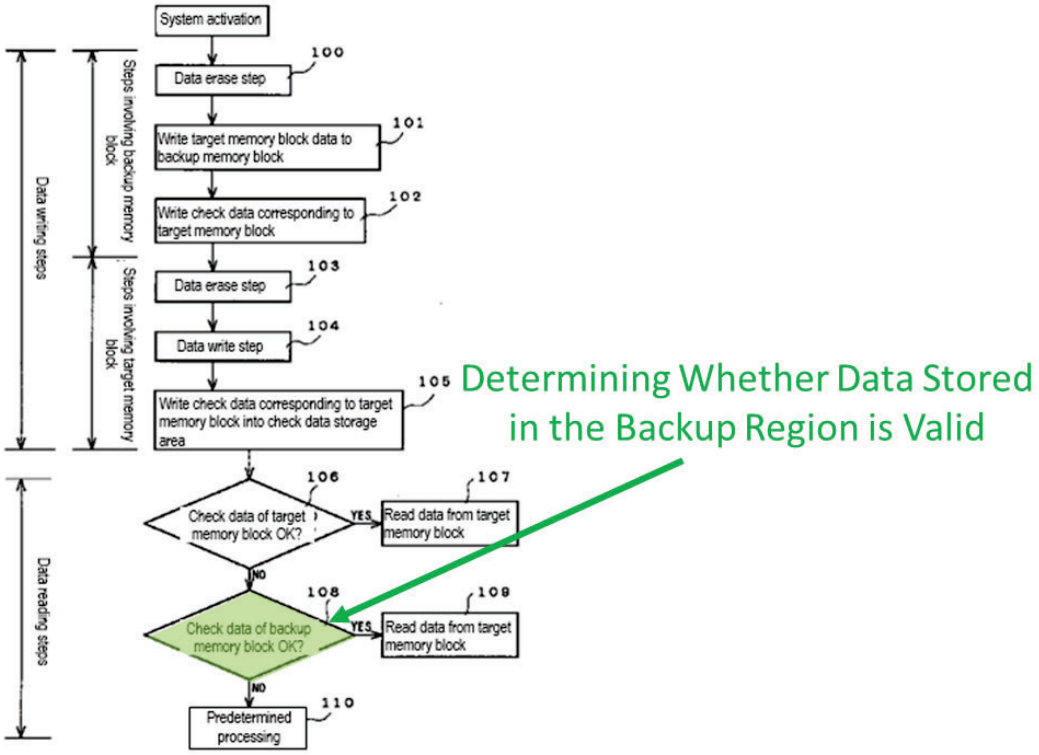
EX1005, Fig. 1

> [T]he check data is read from the check data storage area 3A of this target memory block 2A, and ***it is determined on the basis of this check data if the data which has been written to the target memory block 2A is valid (step 106)***.

EX1005, [0021] (emphasis added); *see also* EX1011, [0221].

When it is determined that the data stored in the data area (i.e. memory block 2A) is invalid, Mitomi then teaches determining whether the data stored in the backup region (i.e. backup memory block 2D) is valid (step 108):

[FIG. 1]



EX1005, Fig. 1

> [I]t is determined that there is no valid data in the target memory block 2A (step 106), ***and next, it is determined if the data which has been written to the data backup memory block 2D is valid as data of the target memory block*** on the basis of whether or not check data specific to the target memory block 2A can be read from the check data storage area 3D (step 108***)***.

EX1005, [0022] (emphasis added); *see also* EX1011, [0222]. As a result, "even if a power supply interruption or the like occurs during the data write operation on the target memory block and the operation is interrupted, the data of the target

memory block will have been saved in the data backup memory block." EX1005, [0008].

Accordingly, both Morihiro and Mitomi recognized that data in flash memory may be corrupted by power failures. *See* EX1006, Abstract; *see also* EX1005, [0005]. In addressing this issue, both Morihiro and Mitomi disclose checking the validity of the data stored in the data area and the backup region. *See* EX1006, [0045]; *see also* EX1005, [0021]-[0022]. Thus, both Morihiro and Mitomi recognized that data within flash memory may be corrupted by power failures and to check the validity of stored data in the data area and backup region. EX1011, [0223]-[0224]. Therefore, because Morihiro and Mitomi recognized similar problems (i.e. flash memory corruption via power failure) and provide similar solutions (i.e. validity checks on the stored data) it would have been obvious to a POSITA to implement Mitomi's teachings of checking the validity of data stored in a data area prior to checking the validity of data stored in a backup region in Morihiro's method. *Id*.

Thus, the combination of Morihiro and Mitomi renders obvious "*prior to the step of overwriting the data stored in the one data sector, further comprising: if the data in the one data sector is determined to be invalid, determining whether the data stored in the backup region is valid using the error detection code associated therewith"* as claimed.

X. THE *FINTIV* FACTORS FAVOR INSTITUTION

Patent Owner has asserted the '015 Patent against Petitioner in the U.S. District Court for the District of Delaware. The Board balances six factors in considering discretionary denial under 35 U.S.C. §314(a) when parallel litigation exists. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential). Here, these factors ("*Fintiv* factors") favor institution.

1. Factor 1 is neutral (possibility of a stay).

A stay pending the outcome of an IPR in the co-pending District Court litigation has not been requested. If an IPR is instituted, a stay would be appropriate under the legal standard in the District of Delaware. *See, e.g., Wilson Wolf Manufacturing Corporation v. Brammer Bio, LLC,* C.A. No. 19-2315-RGA, at 4 n.7 (D. Del. Dec. 8, 2020) (stating "it has become fairly routine to stay cases after IPRs have been instituted").

Litigation between the Petitioner and Patent Owner is at a very early stage. Patent Owner filed its Complaint on June 29, 2021, and served Petitioner on July 12, 2021. EX1012, p.1-2. This Petition was filed in less than four months of service. At the time of filing, the District Court had not yet held a scheduling conference or issued a scheduling order. Therefore, a stay pending IPR would be appropriate due to the early stage of the district court litigation.

Moreover, given that a motion to stay has not yet been filed, the Board should not infer the outcome of that motion. *Intel Corp. v. VLSI Tech. LLC*, IPR2020-00158, Paper 16, at 7 (PTAB May 20, 2020); *Apple Inc. v. Fintiv, Inc.,* IPR2020-00019, Paper 15, at 12 (PTAB May 13, 2020). Thus, this factor is neutral.

2. Factor 2 favors institution (proximity of trial date to final written decision).

With respect to the litigation between Patent Owner and Petitioner, the District Court has not yet issued a scheduling order. *See* EX1012. Thus, no trial date has been set. Because of the uncertainty of the trial date, this factor weighs against discretionary denial. *See Sand Revolution II, LLC v. Continental Intermodal Grp.—Trucking LLC*, IPR2019-01393, Paper 24 (June 16, 2020) (informative) (uncertainty over trial date weighed against discretionary denial).

3. Factor 3 favors institution (investment in parallel proceeding).

All of the related litigations are in their early stages. As discussed above, this Petition was filed promptly without delay in less than four months of service, favoring institution. *Fintiv*, Paper 11 at 11.

With respect to the litigation between Patent Owner and Petitioner, the parties have not exchanged preliminary positions on claim construction, invalidity, or alleged infringement, and fact and expert discovery has not begun. *See* EX1012. Further, there is no evidence that the District Court will conduct a *Markman* hearing or issue a *Markman* ruling before the prospective date for the

Board's institution decision. The early stage of the litigation weighs against discretionary denial.

4. Factor 4 favors institution (overlap in issues).

Petitioner has not served its preliminary invalidity contentions in the district court proceeding (nor has a date been set for service of such contentions). Accordingly, no overlap exists between this proceeding and the District Court proceeding between Patent Owner and Petitioner. Consequently, this factor favors institution, or is at least neutral.

Regardless, to the extent overlap occurs in the future, Petitioner further stipulate herein that, if this IPR proceeding is instituted, it will withdraw any identical grounds from the district court, thus eliminating any overlap in issues. The Board has found that such stipulations weigh in favor of institution. *See Sand Revolution* at 11-12; *Sotera Wireless, Inc. v. Masimo Corporation*, IPR2020-01019, Paper 12 at 19 (Dec. 1, 2020). Regarding non-petitioners in the related litigations, any future overlap is speculative and beyond Petitioners' control, as Petitioners cannot dictate the actions of others. *See Walmart Inc. v. Caravan Canopy Int'l, Inc.*, IPR2020-01026, Paper 12 at 13 (PTAB Dec. 15, 2020) ("Petitioner will have no control over which invalidity issues are presented in any trial or how they are presented"). Consequently, *Fintiv* factor 4 favors institution, or is at least neutral.

5. Factor 5 is neutral (overlap in parties).

As is true of most petitioners in IPR proceedings, Petitioner is a defendant in the litigation. In *Google LLC v. Parus Holdings, Inc*., IPR2020-00846, Paper 9 (Oct. 21, 2020), the panel noted that this factor "could weigh either in favor of, or against, exercising discretion to deny institution, depending on which tribunal was likely to address the challenged patent first." *Id.* at 21. This factor is neutral because it is unknown which tribunal will address the validity of the '015 Patent first. *See id*. (the panel stating "we decline to speculate as to whether we are likely to address the challenged patent before the [district] court. Thus, this factor is neutral.").

6. Factor 6 favors institution (other circumstances).

As discussed in detail above, the merits of Petitioner's arguments are strong, and thus this factor favors institution. *Fintiv*, Paper 11 at 14-15 ("[I]f the merits of a ground raised in the petition seem particularly strong on the preliminary record, this fact has favored institution.").

As such, because each of the *Fintiv* factors are either neutral or weigh against discretionary denial, and because Petitioner diligently filed this Petition in less than four months of service in the District Court litigation, institution should not be denied on discretionary factors.

XI. CONCLUSION

For the reasons detailed above, institution of *inter partes* review of claims 1-11 of the ’015 Patent is requested.

Dated: October 29, 2021

Respectfully submitted,

By: /Kyle L. Howard/
Kyle L. Howard
Registration No. 67,568
Customer No. 27683
Attorney Docket No. 56733.12
Lead Counsel for Petitioner

XII. CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24, the undersigned attorney for the Petitioner, declares that the argument section of this Petition (Sections I, III–XI) has a total of 13,563 words according to the word count tool in Microsoft Word™.

/Kyle L. Howard/
Kyle L. Howard
Registration No. 67,568
Lead Counsel for Petitioner

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of: Akahane *et al.*	§	Petition for *Inter Partes* Review
	§	
U.S. Patent No. 6,851,015	§	Attorney Docket No.: 56733.12
	§	
Issued: Feb. 1, 2005	§	Customer No.: 27683
	§	
Title: Method of Overwriting Data in Nonvolatile Memory and a Control Apparatus Used for the Method	§	Real Parties in Interest: Pure Storage, Inc.
	§	
	§	

CERTIFICATE OF SERVICE

The undersigned certifies, in accordance with 37 C.F.R. §§ 42.6(e) and 42.105, that service was made on the Patent Owner as detailed below.

Date of service October 29, 2021

Manner of service FEDERAL EXPRESS

Documents served Petition for *Inter Partes* Review Under 35 U.S.C. §312 and 37 C.F.R. §42.104; Petitioner's Exhibit List; Certificate of Word Count; Exhibits: EX1001 through EX1016

Persons served Schwabe Williamson & Wyatt
Pacwest Center, Suite 1900
1211 SW Fifth Avenue
Portland, OR 97204

The undersigned further certifies that a courtesy copy of the complete and entire Petition along with exhibits was provided by Federal Express to counsel retained by Patent Owner in the litigation matters involving the Patent Owner and Petitioner identified herein:

Date of service	October 29, 2021
Manner of service	FEDERAL EXPRESS
Documents served	Petition for *Inter Partes* Review Under 35 U.S.C. §312 and 37 C.F.R. §42.104; Petitioner's Exhibit List; Certificate of Word Count; Exhibits: EX1001 through EX1016
Persons served	David W. deBruin GAWTHROP GREENWOOD, PC 3711 Kennett Pike, Suite 100 Wilmington, DE 19807 Isaac Rabicoff RABICOFF LAW LLC 5680 King Centre Dr., Suite 645 Alexandria, VA 22315

/Kyle L. Howard/
Kyle L. Howard
Registration No. 67,568
Lead Counsel for Petitioner