SEAN C. CUNNINGHAM (Bar No. 174931)
sean.cunningham@dlapiper.com
STANLEY J. PANIKOWSKI (Bar No. 224232)
stanley.panikowski@dlapiper.com
ERIN P. GIBSON (Bar No. 229305)
erin.gibson@dlapiper.com
RYAN W. COBB (Bar No. 277608)
ryan.cobb@dlapiper.com
DLA PIPER LLP (US)
401 B Street, Suite 1700
San Diego, CA 92101-4297
Telephone: 619-699-2700
Facsimile: 619-699-2701

RAJIV DHARNIDHARKA (Bar No. 234756)
rajiv.dharnidharka@dlapiper.com
DLA PIPER LLP (US)
400 Capitol Mall, #2400
Sacramento, CA 95814
Telephone: 916-930-3200
Facsimile: 916-930-3201

THOMAS J. WIMBISCUS (*Pro Hac Vice* Pending)
twimbiscus@mcandrews-ip.com
GREGORY C. SCHODDE (*Pro Hac Vice* Pending)
gschodde@mcandrews-ip.com
WAYNE H. BRADLEY (*Pro Hac Vice* Pending)
wbradley@mcandrews-ip.com
McANDREWS, HELD & MALLOY
500 West Madison Street, 34th Floor
Chicago, IL 60661
Telephone: 312-775-8000
Facsimile: 312-775-8100

Attorneys for Plaintiff NETLIST, INC.

FILED
2013 AUG 23 AM 10:45
CLERK U.S. DISTRICT COURT
CENTRAL DIST. OF CALIF.
SANTA ANA
BY

E/S-21

/////

/////

/////

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA
SANTA ANA DIVISION

NETLIST, INC.,

Plaintiff,

v.

SMART MODULAR TECHNOLOGIES, INC., SMART STORAGE SYSTEMS, INC., SMART WORLDWIDE HOLDINGS, INC. and DIABLO TECHNOLOGIES, INC.

Defendants.

Case No. 8:13-cv-00996-DOC-JPR

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT, ANTITRUST VIOLATIONS, TRADEMARK INFRINGEMENT, CORRECTION OF INVENTORSHIP, TRADE SECRET MISAPPROPRIATION, BREACH OF CONTRACT, UNFAIR COMPETITION AND FRAUD; DEMAND FOR JURY TRIAL

Plaintiff Netlist, Inc. ("Netlist") complains against Defendants Smart Modular Technologies, Inc. ("Smart Modular"), Smart Storage Systems, Inc. ("Smart Storage"), Smart Worldwide Holdings, Inc. ("Smart Worldwide") and Diablo Technologies, Inc. ("Diablo") (collectively "Defendants") as follows:

NATURE OF THE ACTION

1. This is an action against Defendants seeking, among other things, the following: (a) damages and injunctive relief for Defendants' infringement of Netlist's patents; (b) relief under federal antitrust laws for Smart Modular and Smart Worldwide's violations of Section 2 of the Sherman Act; and (c) relief under the Lanham Act for Diablo's trademark infringement; (d) damages and other equitable relief under California statutory and common law for Defendants' misappropriation of Netlist's trade secrets, unfair competition, deceptive trade practices and fraud.

/////

THE PARTIES

2. Plaintiff Netlist is a corporation organized and existing under the laws of the State of Delaware with its principal place of business at 51 Discovery, Suite 150, Irvine, California 92618. Netlist is widely regarded in the computing industry as a pioneer in high-performance memory for servers and storage systems, with many of its innovative products being adopted by its primary customers—IBM, HP and Dell. The company was founded in 2000 in Irvine, California by Chuck Hong, Christopher Lopes and Jay Bhakta. The co-founders, who combined have more than 80 years of semiconductor design and marketing experience, recognized that the industry's need for high performance memory products was not being sufficiently met by existing technologies. The company's mission was to create breakthrough technologies that filled the performance gap between the server and storage architectures and the capabilities of existing memory technologies. As a result of its research and development efforts over the past 13 years, Netlist has delivered numerous breakthrough products to the marketplace and amassed a seminal and valuable patent portfolio in the area of high-performance memory architecture and design.

3. Defendant Smart Modular is a Delaware corporation with its principal place of business at 39870 Eureka Drive, Newark, California 94560. On information and belief, Defendant Smart Modular was founded in 1988 as a contract manufacturer of memory products. Its business has been largely building products for others, not creating products from its own design and technology.

4. Defendant Smart Storage is an Arizona corporation with its principal place of business at 39870 Eureka Drive, Newark, California 94560. On information and belief, Defendant Smart Storage designs, develops, manufactures, and deploys enterprise memory modules and solid-state storage products.

5. On information and belief, Defendant Smart Worldwide Holdings, Inc. purports to be a corporation governed by at least three corporate officers, whose

principal place of business is 39870 Eureka Drive, Newark, California 94560. On information and belief, Defendant Smart Worldwide is the parent company of Defendant Smart Modular and Defendant Smart Storage and, as alleged below, directed and continues to direct the relevant actions of both Smart Modular and Smart Storage.

6. Defendant Diablo is a Canadian corporation with its principal place of business at 80 Aberdeen Street, Suite 401, Ottawa, Ontario, K1S 5R5, Canada. On information and belief, Diablo was founded in 2003 and holds itself out as delivering product solutions that enhance the performance and capability of memory system designs. On information and belief, Defendant Diablo has partnered with Defendant Smart Storage to develop the infringing ULLtraDIMM product, as alleged below.

JURISDICTION AND VENUE

7. This Court has jurisdiction over the subject matter of these claims under 28 U.S.C. §§ 1331, 1338(a), 2201 and 2202, because this action involves claims arising under the patent laws of the United States, 35 U.S.C. § 1, et seq., the United States antitrust laws and the Lanham Act. The Court has supplemental jurisdiction under 28 U.S.C. § 1367 for Netlist's related claims under California law.

8. Personal jurisdiction and venue are proper in this Court pursuant to 28 U.S.C. §§ 1391, 1391 and/or 1400, as well as 15 U.S.C. §§ 15 and 22, because Smart Modular, Smart Storage and Smart Worldwide reside in this state and because all Defendants do business in this state and district and have committed acts of infringement in this district.

GENERAL ALLEGATIONS

A. Overview of Defendants' Unlawful Conduct.

9. On information and belief, Defendants have engaged in a pattern of deceptive and unlawful conduct, alone or in concert with one another and other

parties, over a period of at least five years, which includes (a) misappropriation of Netlist's trade secrets, (b) incorporation of Netlist's proprietary technology into competing products, (c) infringement of Netlist's HyperCloud® trademark, (d) filing sham litigation based on a fraudulently procured patent, and (e) making misleading statements to the public in an attempt to put Netlist out of business. On information and belief, Defendants' unlawful scheme was planned and executed in advance of the announcement of an important new product by Defendants, ULLtraDIMM, which infringes multiple Netlist patents and incorporates misappropriated Netlist trade secrets.

10. On information and belief, the chronology of Defendants' misconduct includes the following:

a. In or about early 2008, Diablo approached Netlist seeking to license Netlist's proprietary server memory technology and to design under contract a semiconductor chipset for Netlist's HyperCloud® product. On information and belief, Diablo has been a contract design house that implements semiconductor chip designs on silicon under "for hire" contracts, using its customers' own designs. In March 2008, Netlist and Diablo signed a nondisclosure agreement to begin discussing a business arrangement between the two companies. In April 2008, Netlist provided to Diablo two documents containing specifications of a new product, HyperCloud®, which was to become the heart of Netlist's business. Netlist had been working on HyperCloud® for years, and those years of work were consolidated into the specifications provided to Diablo. Soon afterwards, the technical teams from Netlist and Diablo began holding meetings, during which Netlist presented many aspects of its proprietary design to Diablo. In late April 2008, Netlist provided Diablo with presentation slides entitled "DxD/LRD Opportunity Review," marked "Netlist Confidential" on each page, which detail the architecture and implementation of HyperCloud®. Throughout the summer of

2008, Netlist and Diablo continued to hold numerous meetings and telephone conferences to discuss Netlist's proprietary technology.

b. In September 2008, Netlist signed an agreement with Diablo for Diablo to design and supply chipsets for Netlist based on Netlist's specifications and under Netlist's direction. Netlist continued to disclose to Diablo its most valuable technical trade secrets under strict confidentiality restrictions. Within one week of signing this agreement with Netlist, however, Diablo filed a provisional patent application with the United States Patent & Trademark Office ("Patent Office") that included significant portions of the confidential information Netlist had provided to Diablo. That provisional patent application contains many of the technical features disclosed in the presentation slides Netlist had provided to Diablo, including interoperability, transparency to system, plug-and-play/no modification to BIOS, and other features. Diablo did not name any of the Netlist engineers as inventors on that patent application, nor did Diablo notify Netlist that Diablo was planning to file the patent application.

c. Upon learning of Diablo's provisional patent application, Netlist immediately demanded that the relevant Netlist engineers be named as inventors on the patent application and that it be assigned to Netlist according to the agreement between the companies, which required that all architecture-related intellectual property belong to Netlist. Rather than meeting Netlist's demands, Diablo attempted to use the patent application as leverage to demand that Diablo be permitted to supply more market share of Netlist's chipsets than was called for under the agreement.

d. In May 2009, Diablo filed a second provisional patent application disclosing more technical information provided by Netlist to Diablo, again without notifying Netlist or including Netlist's engineers as inventors. In September 2009, Diablo filed a utility patent application claiming priority to the two provisional patent applications, again without notifying Netlist or including

Netlist's engineers as inventors. Diablo chose to allow the patent application to be published, thereby intentionally disclosing Netlist's proprietary and trade secret technology to the world in March 2010. Netlist was not aware of the pending publication of its technology when it attempted to resolve its disputes with Diablo, resulting in a settlement agreement between Netlist and Diablo in January 2010.

e. In 2012, Diablo began to offer on its website Netlist's proprietary HyperCloud® technology in Diablo's own chipset, VT-Berlinetta. Diablo repeatedly referred to Netlist's HyperCloud® trademark to promote the VT-Berlinetta chipset. Diablo also invited visitors to its website to contact Diablo for more information about the availability and specifications of the VT-Berlinetta chipset, which contains Netlist's trade secrets. Diablo's conduct in offering a competing product using Netlist's trade secrets and without Netlist's consent is a clear breach of the agreements between the parties, among other things. As result of Diablo's repeated unlawful behavior, Netlist sent several demand letters to Diablo before formally terminating the agreement with Diablo in or about July 2012.

f. On information and belief, in or about May 2013, Diablo announced that it had entered into an "exclusive relationship" with Defendant Smart Storage to design and manufacture products for Smart Storage. In or about August 2013, Diablo and Smart Storage announced the introduction of a new product, ULLtraDIMM, that contains Netlist trade secrets misappropriated by Diablo. Smart Storage claims that the "ULLtraDIMM SSD connects flash directly to the CPU through the memory channel by leveraging Diablo Technologies, Inc.'s Memory Channel Storage (MCS) architecture, which SMART Storage Systems has exclusive access to through the companies' technology partnership." On information and belief, Smart Storage knows that Diablo worked closely with Netlist on the HyperCloud® product, and that Diablo's so-called "Memory Channel Storage" architecture was derived from Netlist's HyperCloud® architecture. And

as alleged below, ULLtraDIMM infringes multiple Netlist patents and incorporates numerous Netlist trade secrets that Diablo misappropriated.

g. In September 2012, Smart Modular filed a lawsuit against Netlist alleging that HyperCloud® infringes United States Patent No. 8,250,295 ("the '295 patent") owned by Smart Modular. The '295 patent issued on August 21, 2012 from application No. 10/752,151 ("the '295 application"). Smart Modular prosecuted the '295 application at the Patent Office for eight years, during which time Smart Modular was participating in the standard-setting organization known as JEDEC. The bylaws of JEDEC require its members to disclose all patents and patent applications that its members believe are relevant to the work of the committees in which they participate. Despite now claiming that the '295 patent is relevant to the JEDEC LRDIMM standard, Smart Modular intentionally failed to disclose the '295 application to JEDEC and its members. On information and belief, at the same time Smart Modular was voting to adopt the future LRDIMM standard, it was secretly amending its patent claims to try to cover that same standard. Smart Modular did not disclose the '295 patent to JEDEC until after voting had closed on the LRDIMM standard, after the '295 patent issued, and just before Smart Modular sued Netlist. When Smart Modular disclosed the '295 patent to JEDEC, it promised that licenses would be offered to applicants desiring to utilize the license for the purpose of implementing the relevant JEDEC standards under reasonable terms that are demonstrably free of any unfair discrimination ("RAND" terms). Despite its promise, Smart Modular never offered a RAND license to Netlist, a maker of JEDEC-compliant HyperCloud® products, before suing Netlist and seeking a preliminary injunction.

h. Smart Modular also procured the '295 patent through inequitable conduct at the Patent Office. In February 2012, Netlist provided a list of prior art references to Smart Modular's counsel in a reexamination proceeding on one of Netlist's patents, in which Smart Modular was participating. Netlist had

disclosed prior art references to Smart Modular before, and Smart Modular had disclosed these prior art references to the Patent Office in the prosecution of its '295 application. The list of references provided by Netlist to Smart Modular in February 2012 contained a Japanese patent application by Takeda ("the Takeda reference"), which is highly material to the patentability of Smart Modular's claims in the '295 patent. Contrary to its prior conduct, Smart Modular did not disclose these prior art references to the Examiner of the '295 application. Several months later, the Examiner issued a notice of allowance for the '295 patent, unaware of the material Takeda reference.

i. On information and belief, Smart Modular's conduct after filing the suit demonstrated that the purpose of the lawsuit was not to enforce a valid patent, but to use a fraudulently obtained patent to harm Netlist's business, drain Netlist's resources and dissuade Netlist's customers from adopting HyperCloud®. Disregarding its obligation to offer a RAND license, Smart Modular moved for a preliminary injunction against HyperCloud®. Smart Modular's motion alleged irreparable harm, a position that was contradicted by Smart Modular's own witnesses. Before and during the litigation, Smart Modular also issued a series of deceptive press releases designed to further harm Netlist's business. On information and belief, Smart Modular's agents have also admitted that its purpose in suing Netlist was for reasons other than the enforcement of a valid patent. During the Smart Modular lawsuit, Smart Worldwide's CEO, Iain MacKenzie told Netlist's CEO, Chuck Hong, that MacKenzie "had no beef" against Netlist and that he had been "railroaded" into filing suit against Netlist by other companies.

11. Based on the conduct alleged above and more fully below, Netlist seeks damages, injunctive relief and attorneys' fees as relief from Defendants' unlawful conduct.

/////

/////

B. The Netlist Patents and Defendants' Infringement

12. On August 16, 2011, the United States Patent and Trademark Office duly and lawfully issued United States Patent No. 8,001,434 ("the '434 patent"), entitled "Memory board with self-testing capability" to Dr. Hyun Lee, Jayesh R. Bhakta, and Soonju Choi. A true and correct copy of the '434 patent is attached to this First Amended Complaint as Exhibit A. Netlist owns the entire right, title and interest in and to the '434 patent. The application for the '434 patent was filed on April 13, 2009.

13. On October 30, 2012, the United States Patent and Trademark Office duly and lawfully issued United States Patent No. 8,301,833 ("the '833 patent"), entitled "Non-volatile memory module" to Chi-she Chen, Jeffrey C. Solomon, Scott Milton, and Jayesh Bhakta. A true and correct copy of the '833 patent is attached to this First Amended Complaint as Exhibit B. Netlist owns the entire right, title and interest in and to the '833 patent. The application for the '833 patent was filed on September 29, 2008.

14. On January 22, 2013, the United States Patent and Trademark Office duly and lawfully issued United States Patent No. 8,359,501 ("the '501 patent"), entitled "Memory board with self-testing capability" to Dr. Hyun Lee, Jayesh R. Bhakta, and Soonju Choi. A true and correct copy of the '501 patent is attached to this First Amended Complaint as Exhibit C. Netlist owns the entire right, title and interest in and to the '501 patent. The application for the '501 patent was filed on July 14, 2011.

15. On August 20, 2013, the United States Patent and Trademark Office duly and lawfully issued United States Patent No. 8,516,185 ("the '185 patent"), entitled "System and method utilizing distributed byte-wise buffers on a memory module" to Dr. Hyun Lee and Jayesh R. Bhakta. A true and correct copy of the '185 patent is attached to this First Amended Complaint as Exhibit D. Netlist owns

the entire right, title and interest in and to the '185 patent. The application for the '185 patent was filed on April 15, 2010.

16. On August 20, 2013, the United States Patent and Trademark Office duly and lawfully issued United States Patent No. 8,516,187 ("the '187 patent"), entitled "Data transfer scheme for non-volatile memory module" to Chi-she Chen, Jeffrey C. Solomon, Scott Milton, and Jayesh Bhakta. A true and correct copy of the '187 patent is attached to this First Amended Complaint as Exhibit E. Netlist owns the entire right, title and interest in and to the '187 patent. The application for the '187 patent was filed on June 28, 2012.

17. Where appropriate, Netlist refers to the '434 patent, the '833 patent, the '501 patent, '185 patent, and the '187 patent collectively as the "Netlist Asserted Patents."

18. On or about August 12, 2013, Smart Storage, at the direction of Smart Worldwide, publicly announced that it had begun sampling to select customers its ULLtraDIMM low-latency memory module, which it claims is the result of a close partnership between Smart Storage and Diablo. The ULLtraDIMM memory module employs, among other things, a distributed buffer architecture that infringes the Netlist Asserted Patents, as alleged more fully below.

19. On information and belief, each Defendant has been and is involved in the manufacture, testing, distribution, marketing, offer for sale, and/or sale of the infringing ULLtraDIMM product, and therefore each Defendant is jointly and severally liable for infringement of the Netlist Asserted Patents, as alleged more fully below.

C. Smart Modular and Smart Worldwide's Deceptive and Anticompetitive Conduct

20. This action arises, in part, out of the deceptive, anticompetitive and abusive litigation and related conduct against Netlist by Smart Modular and Smart Worldwide. On information and belief, Smart Worldwide controls the actions of

Smart Modular, its subsidiary, and therefore stands in the shoes of Smart Modular with respect to the conduct alleged herein. On information and belief, Smart Modular and Smart Worldwide's campaign of deceptive and anticompetitive behavior includes the following:

a. At the direction of Smart Worldwide, Smart Modular obtained the '295 patent through inequitable conduct at the United States Patent & Trademark Office ("Patent Office") by deliberately withholding known, material prior art from the Patent Examiner;

b. At the direction of Smart Worldwide, Smart Modular filed a sham patent infringement lawsuit against Netlist, knowing that the '295 patent was both invalid and unenforceable due to Smart Modular's inequitable conduct at the Patent Office, and without conducting a legally sufficient inquiry into the basis for its infringement claims against Netlist;

c. At the direction of Smart Worldwide, Smart Modular claimed in that lawsuit that the '295 patent allegedly covers an essential feature in an industry standard promulgated by the JEDEC standard-setting body, but Smart Modular failed for years to disclose its purportedly relevant '295 patent application, despite a clear duty to disclose that patent application to JEDEC;

d. At the direction of Smart Worldwide, Smart Modular sought a preliminary injunction in the litigation against Netlist's flagship memory product, HyperCloud®, forcing Netlist to expend large sums of money to defeat that baseless motion;

e. At the direction of Smart Worldwide, Smart Modular continually publicized its litigation efforts through misleading and deceptive press releases designed to harm Netlist's business; and

f. At the direction of Smart Worldwide, Smart Modular took these steps of filing a sham patent infringement lawsuit and issuing misleading and

deceptive press releases in an effort to drive Netlist out of business, all prior to the announcement of the ULLtraDIMM product.

21. On information and belief, Smart Modular and Smart Worldwide have colluded with other industry members in some or all of the activities described above, with the goal of furthering Smart Modular and Smart Worldwide's anticompetitive scheme against Netlist. On more than one occasion, agents of Smart Modular and/or Smart Worldwide informed Netlist that Smart Modular had deeper pockets than Netlist, and that Smart Modular was using the litigation to put Netlist out of business.

a. Smart Modular and Smart Worldwide's Fraud on the Patent Office

22. At the direction of Smart Worldwide, Smart Modular breached its duty of candor and good faith to the Patent Office by deliberately withholding known, material prior art from the Patent Office when the '295 patent was being examined by the Patent Office.

23. Smart Modular filed the '295 application on January 5, 2004. During the prosecution of the '295 application, the applicants and/or their counsel became aware of the Takeda reference when Netlist provided that prior art reference to Smart Modular. The Takeda reference is material prior art to the claims of the '295 patent, and the '295 applicants understood the materiality of the Takeda reference. However, to obtain the '295 patent, the applicants and/or their counsel deliberately withheld the Takeda reference from the Patent Examiner.

24. The Takeda reference is "but-for" material to the patentability of the claims of the '295 patent. On December 7, 2012, the Patent Office ordered reexamination of the '295 patent and simultaneously rejected all asserted claims of the '295 patent as being invalid based on the Takeda reference in view of JEDEC Standard 21-C. This rejection demonstrates that the asserted claims of the '295

patent would not have been allowed to issue if the Examiner had known about the Takeda reference during the original prosecution.

25. On information and belief, the applicants and/or their counsel acted with the specific intent to deceive the Patent Office during prosecution of the '295 patent. On October 11, 2010, after more than six years of failing to get any claims of the '295 application allowed, Smart Modular filed a continuation application from the '295 patent application ("the '295 continuation") to provoke a patent interference with Netlist's United States Patent No. 7,619,912 ("the Netlist '912 patent"). To attempt to provoke an interference with the Netlist '912 patent, Smart Modular copied the claims of the '912 patent into the '295 continuation. Therefore, Smart Modular claimed that the '295 application disclosed the same subject matter as claimed in the Netlist '912 patent.

26. On October 20, 2010, Smart Modular filed a request for *inter partes* reexamination of the Netlist '912 patent. Michael Heafey of Orrick, Herrington & Sutcliffe LLP filed that request on behalf of Smart Modular. The Patent Office later consolidated that reexamination with two other reexaminations of the Netlist '912 patent filed by Google and Inphi. The three reexaminations were merged into a single proceeding ("the Netlist '912 reexamination").

27. In the Netlist '912 reexamination, all parties filed Information Disclosure Statements ("IDS") at various times disclosing prior art references the parties believed to be material to the patentability of the Netlist '912 patent. On June 6, 2011, Smart Modular's counsel filed an IDS ("the First IDS") in the '295 application listing prior art references, including the prior art references that had been disclosed in the Netlist '912 reexamination. Smart Modular therefore regarded the prior art references disclosed in the Netlist '912 reexamination as being material to the patentability of the claims in the '295 application.

28. On February 6, 2012, Netlist filed an information disclosure statement ("the Netlist IDS") in the Netlist '912 reexamination disclosing the Takeda

reference, among other things. Netlist attached a Japanese copy, an English abstract, and an English machine translation of the Takeda reference to the Netlist IDS. On that same day, Netlist's reexamination counsel mailed a copy of the Netlist IDS, together with the Japanese copy, English abstract, and English translation of the Takeda reference, to Smart Modular's reexamination counsel Mr. Heafey.

29. On February 7, 2012, Smart Modular filed a Second Supplemental IDS ("the Second IDS") in the '295 application, in which Smart Modular disclosed more prior art references submitted in the Netlist '912 reexamination. In the Second IDS, Smart Modular disclosed prior art references cited by both Smart Modular and Netlist during the Netlist '912 reexamination. Thus, Smart Modular regarded the prior art cited in the Netlist '912 reexamination to be relevant to the '295 patent application.

30. Despite Smart Modular's practice of disclosing in the '295 application essentially all prior art cited in the Netlist '912 reexamination, Smart Modular never disclosed the Takeda reference during the prosecution of the '295 application. Smart Modular and its agents, including its '295 patent prosecution counsel Mr. Durant, had a legal obligation to disclose that reference in the co-pending prosecution of the '295 application.

31. At all relevant times to the '295 application, the following persons had a duty of candor and good faith toward the Patent Office: (i) the named inventors, (ii) every attorney or agent who prepared or prosecuted the '295 patent application, and (iii) every other person who was substantively involved in the preparation or prosecution of the application and who was associated with the inventor, with the assignee, or with anyone to whom there was an obligation to assign the application.

32. Despite their actual knowledge of the Takeda reference at least as early as February 6, 2012, and despite many opportunities to disclose the Takeda reference in the '295 application, Smart Modular deliberately withheld the Takeda

reference from the Patent Office. As a result, the Patent Office issued the '295 patent six months after the Takeda reference was provided to Smart Modular.

33. Based on these facts, the single most reasonable inference is that Smart Modular and/or its counsel made a deliberate decision not to disclose the material Takeda reference in their possession to wrongfully obtain issuance of the '295 patent claims and file this patent infringement lawsuit against Netlist.

b. Smart Modular and Smart Worldwide's Fraud on JEDEC

34. At the direction of Smart Worldwide, Smart Modular also engaged in a pattern of misleading silence and misrepresentations about the purported relevance of the '295 patent application to the development of the JEDEC LRDIMM Standard (LRDIMM is short for Load Reduced Dual In-Line Memory Module), intentionally ignoring a clear duty to disclose its patent interests while voting to approve the LRDIMM Standard and otherwise participating in JEDEC standard-setting activities. The JEDEC Patent Policy requires its members to disclose potentially standard-essential patents and patent applications and to offer licenses to all interested licensees on reasonable and non-discriminatory ("RAND") terms.

35. During the eight-plus years that the '295 application was pending at the Patent Office, Smart Modular participated in the JEDEC standard-setting body and voted multiple times to approve the LRDIMM Standard. The entire time, Smart Modular knew about the purported relevance of the '295 patent to the LRDIMM Standard and secretly amended its '295 claims at the Patent Office without ever disclosing the '295 application to JEDEC, in clear violation of the JEDEC Patent Policy.

36. On August 27, 2012, one week after the '295 patent issued and years after voting for the relevant JEDEC standard, Smart Modular finally submitted a "License Assurance/ Disclosure Form" to JEDEC identifying the '295 patent and promising to license the patent on RAND terms. On that form, Smart Modular identified the "relevant JEDEC Standard" as "JESD82-xx, JC40.4, JC45.4, MB

Specification, JESD82-20A," and promised that "a license will be offered to applicants desiring to utilize the license for the purpose of implementing the JEDEC Standard under reasonable terms that are demonstrably free of any unfair discrimination." Despite its promise, Smart Modular has never offered a RAND license to Netlist, a maker of JEDEC-compliant HyperCloud® products.

c. Smart Modular and Smart Worldwide's Sham Litigation Against Netlist

37. On September 10, 2012, at the direction of Smart Worldwide, Smart Modular filed suit against Netlist in the United States District Court for the Eastern District of California, alleging infringement of the '295 patent. On information and belief, Smart Modular failed to perform a proper investigation to find a reasonable basis to confirm its allegation that HyperCloud® infringes the '295 patent. Weeks later, Smart Modular filed a motion for preliminary injunction against Netlist's flagship products, the HyperCloud® products, ignoring its promise to license the '295 patent on RAND terms. Smart Modular then made a series of misleading statements to the press to make sure that its assertions of patent infringement and its motion for preliminary injunction would create uncertainty and doubt about the viability of Netlist's HyperCloud® products and cause maximum damage to Netlist's business.

38. On information and belief, Smart Modular and Smart Worldwide have proceeded in bad faith with knowledge that the '295 patent is not infringed and is invalid and unenforceable. On information and belief, Smart Modular's lawsuit was not intended to enforce valid patent rights. Instead, it was motivated by the desire to put a competitor out of business. Smart Modular's litigation is knowingly and objectively baseless because (a) the '295 patent is invalid and unenforceable because of Smart Modular's inequitable conduct at the Patent Office and its misleading silence at JEDEC, (b) Smart Modular failed to perform a legally sufficient investigation into infringement of the '295 patent, and (c) Smart Modular

sought a preliminary injunction despite its promise to offer RAND licenses for the '295 patent at JEDEC. On information and belief, Smart Modular and Smart Worldwide have invoked the legal process as an exclusionary, discriminatory, predatory and illegitimate method of harming Netlist's business. The Eastern District of California court has denied Smart Modular's frivolous motion for preliminary injunction and has stayed Smart Modular's lawsuit pending the completion of the reexamination of the '295 patent.

39. On information and belief, Smart Modular and Smart Worldwide's motive is to harm competition and to put a competitor out of business. Their conduct in suing Netlist and seeking a preliminary injunction using the '295 patent, which is tainted with Smart Modular's misconduct at JEDEC and deception at the Patent Office, is fraud on the market and the public.

d. Smart Modular and Smart Worldwide's Deceptive Statements to the Public

40. At the direction of Smart Worldwide, Smart Modular further has engaged in anticompetitive behavior by waging a public relations campaign against Netlist since filing its lawsuit against Netlist, using even administrative orders from the Court to issue press releases designed to harm Netlist's business and cause uncertainty in the market about Netlist's products.

41. For example, on September 13, 2012, Smart Modular issued a press release at the direction of Smart Worldwide announcing the commencement of the lawsuit and stating that Smart Modular has "asked the court for permission to file an immediate motion for preliminary injunction to prevent Netlist from selling HyperCloud® products in order to stop the alleged continuing infringement by Netlist." On September 25, 2012, when the Court granted Smart Modular's *ex parte* request to file a brief exceeding a page limit by three pages, Smart Modular issued a press release stating that the Court had "Authorized Filing a Motion for Immediate Preliminary Injunction." Not surprisingly, given Smart Modular's

deceptive statements to the market, Netlist's stock price and market capitalization dropped by about half in the months after Smart Modular's lawsuit was filed. Netlist's sales of HyperCloud® products also declined dramatically as a result of Smart Modular's sham litigation, it baseless preliminary injunction motion, and its misleading statements to the public.

e. Smart Modular and Smart Worldwide's Collusion With Other Industry Members

42. Smart Modular and/or Smart Worldwide also have made statements to Netlist indicating that their anticompetitive and deceptive campaign was perpetrated by Smart Modular and/or Smart Worldwide with the help of, and possibly at the behest of, other industry members. Smart Modular and/or Smart Worldwide also have made statements to Netlist that Smart Modular and Smart Worldwide's true intent is to put Netlist out of business by outspending Netlist in the litigation.

D. Diablo's Unlawful Conduct

43. Diablo's unlawful conduct has taken several forms, including patent infringement, trademark infringement, misappropriation of Netlist trade secrets and unfair competition.

a. Diablo's Misappropriation of Netlist's Trade Secrets

44. In or about early 2008, Diablo approached Netlist expressing interest in licensing Netlist's patented technology. On or about March 13, 2008, Netlist and Diablo entered into a Mutual Non-Disclosure Agreement ("NDA") for purposes of discussing a collaboration whereby Diablo would develop a chipset for Netlist based on Netlist's trade secret technology. Section 2 of the NDA provides:

> Receiving Party shall (i) hold all Confidential Information in strict confidence and not disclose such Confidential Information to any third parties, (ii) protect the Confidential Information in the same manner as Receiving Party protects its own confidential information of a similar type, but in no case with less than reasonable care, and (iii) not use any

> Confidential Information for any purpose except for the Business Purpose. Receiving Party shall not disclose the Confidential Information to any third party other than persons in the direct employ of Receiving Party who have a need to have access to and knowledge of the Confidential Information solely for the Business Purpose. Each Party shall take appropriate measures by instruction and agreement prior to disclosure to such employees to assure against unauthorized use or disclosure.

45. Beginning in April 2008 and thereafter, Netlist disclosed the crown jewels of Netlist's proprietary and trade secret technology to Diablo, including but not limited to the following documents and related information: (1) the Netlist DxD/LRD Opportunity Review; (2) the Netlist Synch Mode Operation Proposal; (3) the Netlist LRD/DxD Chip – Preliminary Specification, ASIC Controller; and (4) the Netlist LRD/DxD – Preliminary Specification, Isolation Device. Each of these documents is labeled "Netlist Confidential," and each contains information that (a) derives independent economic value from not being generally known to the public and (b) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy.

46. On or about September 10, 2008, Netlist and Diablo entered into a Development And Supply Agreement, which was later amended by a Settlement Agreement and Amendment to Development And Supply Agreement dated January 12, 2010 (collectively the "Supply Agreement") for the purpose of having Diablo design and manufacture certain products for Netlist, at Netlist's direction and using Netlist's proprietary and trade secret technology.

47. Section 8(a) of the Supply Agreement provides in part as follows: "Each party shall treat as confidential all Confidential Information of the other party, shall not use such Confidential Information except as set forth in this Agreement, and shall use reasonable efforts not to disclose such Confidential Information to any third party." Section 8(a) further provides: "For purposes of

clarification, the Netlist Technology is the Confidential Information of Netlist and may not be used for any purpose other than as set forth in this Agreement, including without limitation use of such Netlist Technology to develop a chip competitive to the Netlist Chipset."

48. In connection with the Supply Agreement and the NDA, Netlist disclosed the Netlist trade secrets to Diablo, including but not limited to the documents and related information described above. Netlist provided these documents and other proprietary and trade secret know-how to Diablo pursuant to and in reliance upon the protections of the Supply Agreement and the NDA.

49. Notwithstanding Diablo's obligations to maintain Netlist's proprietary and trade secret information in confidence, Diablo intentionally misappropriated Netlist's trade secret information in designing and developing ULLtraDIMM. ULLtraDIMM contains trade secret technology that was provided by Netlist to Diablo in confidence under the protections of the Supply Agreement and the NDA.

50. Diablo's wrongful conduct in misusing and disclosing to the public Netlist's confidential and proprietary information constitutes trade secret misappropriation, as alleged more fully below.

b. Diablo's Willful Infringement of Netlist's Trademark

51. Netlist is the owner of United States Trademark Registration No. 4,120,406 for the HyperCloud® name. Netlist's trademark registration for HyperCloud® is attached as Exhibit F.

52. On information and belief, Diablo has infringed and is infringing the HyperCloud® trademark by offering a website (http://www.diablo-technologies.com/products/vtblr/) that improperly incorporates the HyperCloud® name in connection with Diablo offering what purports to be its own product (the "Infringing Mark"). Diablo told potential customers for its own product that "Diablo has delivered the world class silicon implementation necessary to address the rapidly growing memory requirements of virtualization, high-performance

computing and database applications via the HyperCloud architecture." (Emphasis added.) Diablo also told potential customers that "HyperCloud breaks traditional memory speed and capacity limitations faced in DDR3 based server platforms, and the VT-Berlinetta chipset plays an integral part of this solution." Diablo even admitted that HyperCloud® was proprietary and patented: "Via proprietary and Patented design techniques, the VT-Berlinetta chipset allows interoperability with RDIMMs in the same channel. This permits end users to re-use existing install bases, and simply ADD HyperCloud DIMMs based on the VT-Berlinetta chipsets." (Emphasis added.)

53. As a result of Diablo's use of the Infringing Mark, consumers and potential consumers of the products offered by Diablo through its website, as well as through other channels of commerce, are likely to be misled into believing that the goods and services that Diablo provides to the public originate from, or are approved or otherwise sponsored by, Netlist.

54. Diablo's use of the Infringing Mark is deceptive and an attempt to unfairly trade on the public recognition of Netlist's registered trademark and the goodwill acquired by Netlist with respect to its HyperCloud® trademark.

c. Diablo's Failure To Name the Netlist Inventors on Diablo's Patent

55. On September 14, 2009, during the collaboration between Netlist and Diablo, Diablo filed patent application number 12/559,185 ("the '185 application"). The '185 application was published on March 18, 2010 as Publication No. US 2010/0070690 A1. Netlist was not aware of the '185 application at the time it was filed. The '185 application issued as United States Patent No. 8,452,917 ("the '917 patent") on May 28, 2013 and names as inventors Maher Amer and Michael Lewis Takefman. The '917 patent is attached as Exhibit G to this First Amended Complaint.

56. On information and belief, Diablo improperly used Netlist's proprietary technology to file the '185 application and to obtain the '917 patent.

The '917 patent contains details of Netlist's proprietary technology that Diablo and the named inventors of the '917 patent learned from Netlist under the protections of the NDA and the Supply Agreement.

57. The true inventor of the Netlist proprietary technology contained in the '917 patent is Dr. Hyun Lee. Dr. Lee conceived of the technology claimed in multiple claims of the '917 patent. Diablo failed to name Dr. Lee as an inventor when it filed the '185 application with the Patent Office, nor did Diablo name Dr. Lee at any time during the prosecution of the '185 application. Thus, the '917 patent issued without the proper inventors being named on the patent.

COUNT ONE

Infringement of the '434 Patent

(Against All Defendants)

58. Netlist incorporates by reference the preceding allegations of its Complaint.

59. Netlist is the sole holder of the entire right, title and interest in the '434 patent.

60. Defendants have infringed and continue to infringe, directly and indirectly by contributing to and inducing the infringement of others, at least claim 1 of the '434 patent by making, using, importing, selling, and/or offering to sell memory modules that are covered by the claims of the '434 patent, including but not limited to ULLtraDIMM memory modules.

61. On information and belief, Defendants have been on notice of and aware of the '434 patent since at least August 22, 2013.

62. On information and belief, Defendants have actively induced, and continue to actively induce, the infringement of one or more claims of the '434 patent by, among other things, instructing their customers to operate the accused products in a manner that infringes the claims of the '434 patent. On information and belief, Defendants intend that their customers will use the accused memory

modules in a manner that infringes the '434 patent. On information and belief, use of the accused memory modules in their intended manner results in infringement of the '434 patent.

63. On information and belief, Defendants have contributorily infringed and continue to contributorily infringe the '434 patent by offering to sell and/or selling within the United States and/or importing into the United States one or more components of a machine, manufacture, or combination covered by the '434 patent that constitute a material part of the invention, which is not a staple article or commodity of commerce suitable for substantial noninfringing use. On information and belief, Defendants know that the component and/or apparatus is especially made or especially adapted for use in infringing the '434 patent.

64. On information and belief, Defendants have been on notice of and aware of the '434 patent since at least August 22, 2013 and have nevertheless continued to infringe, despite an objectively high likelihood that their actions constitute infringement of the '434 patent. Accordingly, Defendants' infringement has been and continues to be willful.

65. The filing of this complaint also constitutes notice to Defendants of the '434 patent under 35 U.S.C. § 287. Furthermore, on information and belief, Defendants have knowledge of Netlist's patent portfolio, because Defendants Smart Modular and Smart Worldwide have sought Patent Office reexamination of multiple Netlist patents since at least October 2010.

66. As a result of Defendants' infringement of the '434 patent, Netlist is suffering irreparable harm for which it has no adequate remedy at law. Unless enjoined by this Court, this infringement will continue and will result in further irreparable harm to Netlist.

67. Netlist is entitled to recover damages from Defendants not less than a reasonable royalty adequate to compensate for the infringement.

/////

COUNT TWO

Infringement of the '833 Patent

(Against All Defendants)

68. Netlist incorporates by reference the preceding allegations of its Complaint.

69. Netlist is the sole holder of the entire right, title and interest in the '833 patent.

70. Defendants have infringed and continue to infringe, directly and indirectly by contributing to and inducing the infringement of others, at least claim 15 of the '833 patent by making, using, importing, selling, and/or offering to sell memory modules that are covered by the claims of the '833 patent, including but not limited to ULLtraDIMM memory modules.

71. On information and belief, Defendants have been on notice of and aware of the '833 patent since at least August 22, 2013.

72. On information and belief, Defendants have actively induced, and continue to actively induce, the infringement of one or more claims of the '833 patent by, among other things, instructing their customers to operate the accused products in a manner that infringes the claims of the '833 patent. On information and belief, Defendants intend that their customers will use the accused memory modules in a manner that infringes the '833 patent. On information and belief, use of the accused memory modules in their intended manner results in infringement of the '833 patent.

73. On information and belief, Defendants have contributorily infringed and continue to contributorily infringe the '833 patent by offering to sell and/or selling within the United States and/or importing into the United States one or more components of a machine, manufacture, or combination covered by the '833 patent that constitute a material part of the invention, which is not a staple article or commodity of commerce suitable for substantial noninfringing use. On information

and belief, Defendants know that the component and/or apparatus is especially made or especially adapted for use in infringing the '833 patent.

74. On information and belief, Defendants have been on notice of and aware of the '833 Patent since at least August 22, 2013 and have nevertheless continued to infringe, despite an objectively high likelihood that their actions constitute infringement of the '833 patent. Accordingly, Defendants' infringement has been and continues to be willful.

75. The filing of this complaint also constitutes notice to Defendants of the '833 patent under 35 U.S.C. § 287. Furthermore, on information and belief, Defendants have knowledge of Netlist's patent portfolio, because Defendants Smart Modular and Smart Worldwide have sought Patent Office reexamination of multiple Netlist patents since at least October 2010.

76. As a result of Defendants' infringement of the '833 patent, Netlist is suffering irreparable harm for which it has no adequate remedy at law. Unless enjoined by this Court, this infringement will continue and will result in further irreparable harm to Netlist.

77. Netlist is entitled to recover damages from Defendants not less than a reasonable royalty adequate to compensate for the infringement.

COUNT THREE

Infringement of the '501 Patent

(Against All Defendants)

78. Netlist incorporates by reference the preceding allegations of its Complaint.

79. Netlist is the sole holder of the entire right, title and interest in the '501 patent.

80. Defendants have infringed and continue to infringe, directly and indirectly by contributing to and inducing the infringement of others, at least claim 1 of the '501 patent by making, using, importing, selling, and/or offering to sell

memory modules that are covered by the claims of the '501 patent, including but not limited to ULLtraDIMM memory modules.

81. On information and belief, Defendants have been on notice of and aware of the '501 patent since at least August 22, 2013.

82. On information and belief, Defendants have actively induced, and continue to actively induce, the infringement of one or more claims of the '501 patent by, among other things, instructing their customers to operate the accused products in a manner that infringes the claims of the '501 patent. On information and belief, Defendants intend that their customers will use the accused memory modules in a manner that infringes the '501 patent. On information and belief, use of the accused memory modules in their intended manner results in infringement of the '501 patent.

83. On information and belief, Defendants have contributorily infringed and continue to contributorily infringe the '501 patent by offering to sell and/or selling within the United States and/or importing into the United States one or more components of a machine, manufacture, or combination covered by the '501 patent that constitute a material part of the invention, which is not a staple article or commodity of commerce suitable for substantial noninfringing use. On information and belief, Defendants know that the component and/or apparatus is especially made or especially adapted for use in infringing the '501 patent.

84. On information and belief, Defendants have been on notice of and aware of the '501 patent since at least August 22, 2013 and have nevertheless continued to infringe, despite an objectively high likelihood that their actions constitute infringement of the '501 patent. Accordingly, Defendants' infringement has been and continues to be willful.

85. The filing of this complaint also constitutes notice to Defendants of the '501 patent under 35 U.S.C. § 287. Furthermore, on information and belief, Defendants have knowledge of Netlist's patent portfolio, because Defendants Smart

Modular and Smart Worldwide have sought Patent Office reexamination of multiple Netlist patents since at least October 2010.

86. As a result of Defendants' infringement of the '501 patent, Netlist is suffering irreparable harm for which it has no adequate remedy at law. Unless enjoined by this Court, this infringement will continue and will result in further irreparable harm to Netlist.

87. Netlist is entitled to recover damages from Defendants not less than a reasonable royalty adequate to compensate for the infringement.

COUNT FOUR

Infringement of the '185 Patent

(Against All Defendants)

88. Netlist incorporates by reference the preceding allegations of its Complaint.

89. Netlist is the sole holder of the entire right, title and interest in the '185 patent.

90. Defendants have infringed and continue to infringe, directly and indirectly by contributing to and inducing the infringement of others, at least claim 1 of the '185 patent by making, using, importing, selling, and/or offering to sell memory modules that are covered by the claims of the '185 patent, including but not limited to ULLtraDIMM memory modules.

91. On information and belief, Defendants have been on notice of and aware of the '185 Patent since at least August 22, 2013.

92. On information and belief, Defendants have actively induced, and continue to actively induce, the infringement of one or more claims of the '185 patent by, among other things, instructing their customers to operate the accused products in a manner that infringes the claims of the '185 patent. On information and belief, Defendants intend that their customers will use the accused memory modules in a manner that infringes the '185 patent. On information and belief, use

of the accused memory modules in their intended manner results in infringement of the '185 patent.

93. On information and belief, Defendants have contributorily infringed and continue to contributorily infringe the '185 patent by offering to sell and/or selling within the United States and/or importing into the United States one or more components of a machine, manufacture, or combination covered by the '185 patent that constitute a material part of the invention, which is not a staple article or commodity of commerce suitable for substantial noninfringing use. On information and belief, Defendants know that the component and/or apparatus is especially made or especially adapted for use in infringing the '185 patent.

94. On information and belief, Defendants have been on notice of and aware of the '185 patent since at least August 22, 2013 and have nevertheless continued to infringe, despite an objectively high likelihood that their actions constitute infringement of the '185 patent. Accordingly, Defendants' infringement has been and continues to be willful.

95. The filing of this complaint also constitutes notice to Defendants of the '185 patent under 35 U.S.C. § 287. Furthermore, on information and belief, Defendants have knowledge of Netlist's patent portfolio, because Defendants Smart Modular and Smart Worldwide have sought Patent Office reexamination of multiple Netlist patents since at least October 2010.

96. As a result of Defendants' infringement of the '185 patent, Netlist is suffering irreparable harm for which it has no adequate remedy at law. Unless enjoined by this Court, this infringement will continue and will result in further irreparable harm to Netlist.

97. Netlist is entitled to recover damages from Defendants not less than a reasonable royalty adequate to compensate for the infringement.

/////

/////

COUNT FIVE

Infringement of the '187 Patent

(Against All Defendants)

98. Netlist incorporates by reference the preceding allegations of its Complaint.

99. Netlist is the sole holder of the entire right, title and interest in the '187 patent.

100. Defendants have infringed and continue to infringe, directly and indirectly by contributing to and inducing the infringement of others, at least Claim 26 of the '187 patent by making, using, importing, selling, and/or offering to sell memory modules that are covered by the claims of the '187 patent, including but not limited to ULLtraDIMM memory modules.

101. On information and belief, Defendants have been on notice of and aware of the '187 patent since at least August 22, 2013.

102. On information and belief, Defendants have actively induced, and continue to actively induce, the infringement of one or more claims of the '187 patent by, among other things, instructing their customers to operate the accused products in a manner that infringes the claims of the '187 patent. On information and belief, Defendants intend that their customers will use the accused memory modules in a manner that infringes the '187 patent. On information and belief, use of the accused memory modules in their intended manner results in infringement of the '187 patent.

103. On information and belief, Defendants have contributorily infringed and continue to contributorily infringe the '187 patent by offering to sell and/or selling within the United States and/or importing into the United States one or more components of a machine, manufacture, or combination covered by the '187 patent that constitute a material part of the invention, which is not a staple article or commodity of commerce suitable for substantial noninfringing use. On information

and belief, Defendants know that the component and/or apparatus is especially made or especially adapted for use in infringing the '187 patent.

104. On information and belief, Defendants have been on notice of and aware of the '187 patent since at least August 22, 2013 and have nevertheless continued to infringe, despite an objectively high likelihood that their actions constitute infringement of the '187 patent. Accordingly, Defendants' infringement has been and continues to be willful.

105. The filing of this complaint also constitutes notice to Defendants of the '187 patent under 35 U.S.C. § 287. Furthermore, on information and belief, Defendants have knowledge of Netlist's patent portfolio, because Defendants Smart Modular and Smart Worldwide have sought Patent Office reexamination of multiple Netlist patents since at least October 2010.

106. As a result of Defendants' infringement of the '187 patent, Netlist is suffering irreparable harm for which it has no adequate remedy at law. Unless enjoined by this Court, this infringement will continue and will result in further irreparable harm to Netlist.

107. Netlist is entitled to recover damages from Defendants not less than a reasonable royalty adequate to compensate for the infringement.

COUNT SIX

Attempted Monopolization Under 15 U.S.C. § 2

(Against Defendants Smart Modular and Smart Worldwide)

108. As alleged above, Smart Modular and Smart Worldwide have willfully pursued a plan to exclude competition and obtain monopoly power in a relevant market with power to exercise control over prices.

109. The relevant market is a technology market consisting of rank multiplication (the "Relevant Technology"). Rank multiplication is embodied in the JEDEC LRDIMM Standard. Smart Modular and Smart Worldwide allege that the '295 patent covers the Relevant Technology.

110. The relevant geographic market for the Relevant Technology is worldwide. The United States constitutes a relevant geographic submarket in which it is proper to assess the anticompetitive effects of Smart Modular and Smart Worldwide's improper assertion of the '295 patent.

111. Consumers in the market of the Relevant Technology include manufacturers of products that incorporate the Relevant Technology and end users of those products, who necessarily deploy the Relevant Technology when they operate those products.

112. Because of the growing adoption of the LRDIMM Standard, there are no readily-available or reasonably-interchangeable substitutes for the Relevant Technology. Netlist and other manufacturers have further made substantial investments in products incorporating the Relevant Technology (the "Relevant Products"), and consumers have come to rely on the Relevant Technology.

113. At the direction of Smart Worldwide, Smart Modular is interpreting the claims of the '295 patent in a manner that would potentially preclude other companies from competing in the production and sale of the Relevant Products. Smart Modular and Smart Worldwide have sought to give credibility to the '295 patent by touting the '295 patent as essential to compliance with the LRDIMM Standard.

114. The detrimental impact of Smart Modular and Smart Worldwide's unlawful conduct and the specific injury to Netlist is evident. As a result of Smart Modular and Smart Worldwide's unlawful conduct, there have been or will be adverse competitive effects in the relevant market. Smart Modular and Smart Worldwide have sought to capitalize on the disruptive effects of, and resulting uncertainty generated by, Smart Modular's claim of patent infringement against Netlist. The inevitable impact of Smart Modular's inequitable conduct at the Patent Office, its misuse of an invalid and unenforceable patent, and its decision to file a motion for preliminary injunction against Netlist's Relevant Products, and its

misleading media campaign against Netlist have been to impose on consumers of the Relevant Technology the burden or threat of an elevated price for the Relevant Technology as a direct result of bad-faith patent enforcement.

115. By engaging in this unlawful conduct, Smart Modular and Smart Worldwide have further threatened to deny consumers the competitive benefits of future standard-setting efforts. Willingness to participate in future standard-setting activities, including efforts to extend LRDIMM, has been or will be chilled.

116. Smart Modular and Smart Worldwide have pursued a scheme with specific intent to monopolize the Relevant Technology and have engaged in unreasonable, anticompetitive and exclusionary conduct through activities that have included, as alleged above: (i) securing the '295 patent through inequitable conduct at the Patent Office and seeking to enforce the '295 patent; (ii) seeking to enforce the patent against JEDEC-compliant products knowing that the patent is invalid and unenforceable; (iii) asserting patent claims to give themselves control over the Relevant Technology by threatening, instituting and perpetuating objectively baseless litigation in bad faith; and (iv) filing a motion for preliminary injunction attempting to halt the sale of Netlist's HyperCloud® products, which Smart Modular and Smart Worldwide allege to be the primary competition with Smart Modular's own LRDIMM products.

117. Based on the facts alleged above, there was and is a dangerous probability that Smart Modular and Smart Worldwide will succeed in obtaining monopoly power in the Relevant Technology unless restrained from engaging in further unlawful conduct.

118. Smart Modular and Smart Worldwide's unlawful conduct has caused injury to competition and to Netlist as creators and consumers of the Relevant Technology and, unless enjoined and restrained, will cause further injury to competition and to Netlist.

/////

119. Netlist has been injured in its business and has been threatened with further injury, including the payment of attorneys' fees and other defense costs associated with Smart Modular's patent infringement claims, as a direct and proximate result of Smart Modular and Smart Worldwide's conduct.

COUNT SEVEN

***Walker Process* Violation Under 15 U.S.C. § 2**

Bad Faith Enforcement of Fraudulently-Obtained Patent

(Against Defendants Smart Modular and Smart Worldwide)

120. Netlist incorporates by reference the preceding allegations of its Complaint.

121. This claim seeks damages and injunctive relief against Smart Modular and Smart Worldwide for a *Walker Process* violation by reason of Smart Modular and Smart Worldwide's fraud on the Patent Office and their resulting bad faith enforcement of a fraudulently-obtained patent in violation of Section 2 of the Sherman Act, 15 U.S.C. § 2.

122. This intentional failure to disclose known and material prior art to the Patent Office constitutes fraud on the Patent Office that is actionable under Section 2 of the Sherman Antitrust Act.

123. By reason of its fraud on the Patent Office, Smart Modular and Smart Worldwide have a dangerous probability of achieving monopoly power in the international and United States markets for rank multiplication technology.

124. Smart Modular and Smart Worldwide's deceitful conduct has had a substantial effect on interstate (and foreign) commerce and it will continue to have such an effect as long as Smart Modular and Smart Worldwide continue to assert that the '295 patent is valid, enforceable and necessary to practice the LRDIMM Standard.

125. As a direct and proximate result of Smart Modular and Smart Worldwide's unlawful attempted monopolization, Netlist has been injured in its

business and property in an amount that has yet to be determined, but will be established at trial.

126. Unless Smart Modular and Smart Worldwide are enjoined by this Court, Smart Modular and Smart Worldwide's unlawful conduct will continue and Netlist will continue to sustain injury and damages.

COUNT EIGHT

Lanham Act Violation—15 U.S.C. § 1114(a)

(Against Defendant Diablo)

127. Netlist incorporates by reference the preceding allegations of its Complaint.

128. Netlist owns all right, title and interest in and to the HyperCloud® registered trademark, as alleged above.

129. Diablo is improperly and willfully infringing the HyperCloud® registered trademark in interstate commerce through the advertising, promotion and sale of goods under the Infringing Mark.

130. Diablo's use in interstate commerce of the Infringing Mark that are confusingly similar to the HyperCloud® registered trademark is likely to cause confusion or deception of consumers as to the source, origin, or sponsorship of Diablo's goods in violation of Section 32 of the Lanham Act, 15 U.S.C. § 1114. Customers are likely to purchase Diablo's goods believing them to be those of Netlist or otherwise affiliated or sponsored by Netlist.

131. Diablo's conduct constitutes trademark infringement in violation of Section 32 of the Lanham Act, 15 U.S.C. § 1114.

132. As a direct and proximate result of Diablo's conduct, Netlist has been, and is likely to be, substantially injured.

133. Netlist has no adequate remedy at law, and unless enjoined by this Court, Diablo will continue to engage in such acts of trademark infringement, to the irreparable damage and injury of Netlist.

134. On information and belief, Diablo has engaged in the above-referenced acts of trademark infringement with full knowledge of Netlist's exclusive rights in the HyperCloud® registered trademark, and Diablo continues in such acts of intentional infringement, thus making this case exceptional and entitling Netlist to an award of treble its actual damages, plus attorneys' fees and costs in bringing and maintaining this action, pursuant to Section 35(b) of the Lanham Act, 15 U.S.C. § 1117(b).

COUNT NINE

Lanham Act Violation—15 U.S.C. 1125(a)

(Against Defendant Diablo)

135. Netlist incorporates by reference the preceding allegations of its Complaint.

136. Netlist owns all right, title and interest in and to the HyperCloud® registered trademark, as alleged above.

137. Diablo's infringement, as alleged above, constitutes the use of false designations of origin or false or misleading descriptions and representations of fact, which are likely to cause confusion or mistake, or to deceive consumers, as to the affiliation, connection or association of Diablo and the Infringing Mark and with Netlist, or as to the origin, sponsorship, or approval by Netlist of the goods and commercial activities offered and sold by Diablo, in violation of Section 43(a) of the Lanham Act, 15 U.S.C. § 1125(a).

138. Diablo's infringement, as alleged above, has been made in commercial advertising or promotion and misrepresents the nature and characteristics of Diablo's goods, in violation of 15 U.S.C. § 1125(a).

139. On information and belief, Diablo has undertaken its infringing activities with the willful intent of deceiving and misleading the public into falsely believing that the goods offered in conjunction with the Infringing Mark are

associated with Netlist and therefore are deserving of the public's trust and business.

140. As a result of Diablo's wrongful and willful acts, as set forth herein, Netlist has suffered and will continue to suffer irreparable injury which is not capable of monetary determination.

141. Netlist has no adequate remedy at law.

142. Diablo has engaged in acts of unfair competition, false representation and designation with knowledge of the exclusive rights of Netlist to its registered trademark, and Diablo continues in such acts of unfair competition, intentional false representation and designation, thus entitling Netlist to an award of its actual damages, Diablo's profits, plus attorneys' fees and costs in bringing and maintaining this action, pursuant to Section 35(a) of the Lanham Act, 15 U.S.C. § 1117(a).

COUNT TEN

Correction of Inventorship (35 U.S.C. § 256) and Ownership (35 U.S.C. § 262)

(Against Defendant Diablo)

143. Netlist incorporates by reference the preceding allegations of its Complaint.

144. On information and belief, Diablo improperly used Netlist's proprietary and trade secret technology to file a patent application, which issued as the '917 patent, as alleged above.

145. The Netlist Inventors communicated the conception of the '917 patent to the named inventors of the '917 patent. This communication occurred before the date that Diablo claims to have conceived of the alleged inventions disclosed in the '917 patent. The named inventors of the '917 patent derived the subject matter claimed in the '917 patent from the Netlist Inventors.

/////

/////

146. The Netlist Inventors were not named as inventors of the '917 patent. Each of the Netlist Inventors was unaware of Diablo's patent-related activities and had no deceptive intent in not being named as inventors.

147. Pursuant to 35 U.S.C. § 256, Netlist requests that the Court order the Patent Office to correct the inventorship of the '917 patent to identify Dr. Hyun Lee as a co-inventor of that patent. Netlist also seeks a declaration that Netlist is a joint owner of the '917 patent under 35 U.S.C. § 262.

COUNT ELEVEN

Misappropriation of Trade Secrets—Civil Code § 3426 et seq.

(Against Defendant Diablo)

148. Netlist incorporates by reference the preceding allegations of its Complaint.

149. At all relevant times, Netlist was in possession of valuable trade secrets as alleged above, including but not limited to information concerning Netlist's proprietary memory technology.

150. Netlist's trade secrets, as alleged above, have tremendous economic value to Netlist.

151. As alleged above, Netlist has made reasonable efforts to ensure that its trade secrets remain secret.

152. On information and belief, Diablo has misappropriated and continues to misappropriate Netlist's trade secrets by engaging in the conduct alleged above. Diablo's misappropriation of Netlist's trade secrets has caused, and will continue to cause, substantial injury to Netlist consisting of, among other things, harm to its competitive business, loss of key technology useful in generating new products and loss of valuable business opportunities.

153. As a proximate result of Diablo's misappropriation of Netlist's trade secrets, Netlist has suffered damages in an amount to be determined at trial.

/////

154. On information and belief, Diablo's acts were willful and malicious, and Diablo misappropriated Netlist's trade secrets with a deliberate intent to injure Netlist's business and improve its own, thereby depriving Netlist of property and legal rights and otherwise causing Netlist injury. Accordingly, Netlist is entitled to an award of exemplary damages and attorneys' fees under Civil Code sections 3426.3 and 3426.4.

155. Diablo's wrongful conduct in disclosing and using Netlist's trade secrets, unless and until enjoined by order of this Court, will continue in the future and will result in irreparable injury to Netlist. Among other things, Netlist is severely threatened with the loss of valuable and profitable business opportunities if Diablo's actions are not enjoined.

156. Netlist has no adequate remedy at law for the injuries it is currently suffering and will continue to suffer in the future. Furthermore, unless Diablo is enjoined by order of this Court, Diablo will continue to make use of the trade secrets misappropriated from Netlist.

COUNT TWELVE

Breach of Contract

(Against Defendant Diablo)

157. Netlist incorporates by reference the preceding allegations of its Complaint.

158. Diablo's wrongful conduct, as alleged above, constitutes a material breach of the NDA and Supply Agreement between Netlist and Diablo.

159. Diablo's breach of contract has caused and is causing damage to Netlist, in an amount to be proven at trial.

160. Netlist is entitled to specific performance of certain provisions of the NDA and Supply Agreement, including but not limited to provisions requiring the return of Netlist's confidential information and the transfer of all right, title and interest in the '917 patent to Netlist.

COUNT THIRTEEN

Unfair Competition—California Bus. & Prof. Code § 17200

(Against All Defendants)

161. Netlist incorporates by reference the preceding allegations of its Complaint.

162. California Business & Professions Code § 17200 prohibits acts of "unfair competition," including any ""unlawful, unfair or fraudulent business act or practice."

163. As alleged above, Defendants have engaged in unfair competition or unlawful, unfair, or fraudulent business practices in violation of the Unfair Competition law when they engaged in any one or more of the following:

a. Smart Modular and Smart Worldwide disparaged Netlist's business and products by falsely and misleadingly alleging that Netlist's products infringe the '295 patent and by seeking to enjoin Netlist from making and selling products to consumers using a wrongfully obtained patent;

b. Smart Modular and Smart Worldwide falsely asserted, in litigation and in public press releases, that Netlist infringed the '295 patent, which Smart Modular and Smart Worldwide knew was invalid and unenforceable;

c. Defendants engaged in collusion with other industry participants with the specific intent of harming Netlist's business; and

d. Diablo engaged in acts of trade secret misappropriation and trademark infringement with the specific intent of harming Netlist's business.

164. Defendants' deceptive, fraudulent, negligent, and/or unlawful business practices are ongoing and are specifically designed to injure Netlist.

165. Defendants have acted in bad faith while engaging in these deceptive trade practices.

166. Netlist has been harmed by Defendants' unfair competition and deceptive trade practices. This harm will continue or can reasonably be expected to

continue as long as Defendants' acts of unfair competition are permitted to continue.

COUNT FOURTEEN

Common Law Fraud

(Against Defendants Smart Modular and Smart Worldwide)

167. Netlist incorporates by reference the preceding allegations of its Complaint.

168. Netlist has been a member of JEDEC since 2000 and a member of several JEDEC committees, including JC-40 and JC-45.

169. At all times relevant to this claim, Smart Modular, at the direction of Smart Worldwide, was a member of JEDEC and a member of several JEDEC committees, including JC-40 and JC-45.

170. Despite its membership in several JEDEC committees and its active participation in JEDEC voting activities, Smart Modular and Smart Worldwide did not disclose the '295 application to the relevant JEDEC committees during the eight-plus years the '295 application was pending in the Patent Office. The '295 application was filed in the Patent Office on January 5, 2004. At no time between January 5, 2004 and August 27, 2012 did Smart Modular or Smart Worldwide disclose the '295 application to JEDEC, as they were required to do under the JEDEC Patent Policy. As a result, specifications of several relevant JEDEC standards were drafted, finalized and published, and the JEDEC committees had no opportunity to consider reasonable workarounds or technical alternatives to the technology purportedly disclosed in the '295 application.

171. Despite Smart Modular and Smart Worldwide's failure to timely disclose the '295 application and despite its failure to offer a license to Netlist on reasonable and nondiscriminatory terms, Smart Modular is asserting the '295 against Netlist, a JEDEC member and maker of JEDEC-compliant products.

/////

172. Despite their duty to disclose relevant patents and its knowledge of the proposed LRDIMM Standard, Smart Modular and Smart Worldwide intentionally failed to disclose the application for the '295 patent. Smart Modular and Smart Worldwide failed to disclose the '295 application because they intended to induce the adoption of the LRDIMM Standard and its inclusion in various products.

173. As a result of Smart Modular and Smart Worldwide's fraudulent failure to disclose the '295 application and subsequent attempts to assert the patent, Netlist has incurred damages and will continue to be damaged in the future.

PRAYER FOR RELIEF

WHEREFORE, Netlist prays for the following relief:

i) Judgment that the Netlist Asserted Patents are each valid, enforceable and infringed by Defendants;

ii) A preliminary and permanent injunction enjoining Defendants, their officers, agents, servants, employees, subsidiaries and affiliated companies, and those persons acting in active concert or participation therewith, from engaging in the aforesaid unlawful acts of patent infringement;

iii) An award of damages arising out of each of Defendants' acts of patent infringement, together with pre-judgment and post-judgment interest;

iv) Judgment that damages be trebled in accordance with 35 U.S.C. § 284;

v) An order declaring this case exceptional and awarding Netlist its attorneys' fees and costs incurred in this action, pursuant to 35 U.S.C. § 285;

vi) An award of damages adequate to compensate Netlist for the harm it has sustained as a result of Smart Modular and Smart Worldwide's unlawful and anticompetitive conduct;

vii) An award of treble damages under Clayton Act § 4 against Smart Modular and Smart Worldwide for their Sherman Act violations;

viii) An award of attorneys' fees against Smart Modular and Smart Worldwide under Clayton Act §§ 4 and 16;

ix) Judgment that Diablo has willfully violated Section 32 of the Federal Trademark Act, 15 U.S.C. § 1114, by infringing the federally registered HyperCloud® trademark;

x) Judgment that Diablo has violated Section 43(a) of the Federal Trademark Act, 15 U.S.C. § 1125(a)(1)(A), by engaging in federal unfair competition;

xi) An order directing the Patent Office to correct the inventorship of the '917 patent to identify the Netlist Inventors as inventors of the '917 patent, and judgment declaring that Netlist is a joint owner of the '917 patent;

xii) An award of damages to compensate Netlist for Defendant Diablo's acts of trade secret misappropriation;

xiii) A preliminary and permanent injunction enjoining Diablo, its officers, agents, servants, employees, subsidiaries and affiliated companies, and those persons acting in active concert or participation therewith, from engaging in the aforesaid unlawful acts of trade secret misappropriation;

xiv) An award of damages to compensate Netlist for Defendant Diablo's breach of contract;

xv) Specific performance of certain provisions of the NDA and Supply Agreement, including but not limited to provisions requiring the return of Netlist's confidential information and the transfer of all right, title and interest in the '917 patent to Netlist;

xvi) An order awarding Netlist damages and Diablo's profits pursuant to 15 U.S.C. § 1117(a), trebled pursuant to 15 U.S.C. § 1117(b), and statutory damages pursuant to 15 U.S.C. § 1117(d);

xvii) An order under California Business and Professions Code section 17203 that Defendants and their successors, agents, representatives, employees and all persons who act in concert with Defendants, be permanently enjoined from committing any acts of unfair competition;

xviii) Such other and further equitable or legal relief as the Court or a jury deems proper.

DEMAND FOR JURY TRIAL

In accordance with Fed. R. Civ. P. 38(b), Netlist demands a trial by jury on all issues so triable.

Dated: August 23, 2013

DLA PIPER LLP (US)

By: _____
SEAN C. CUNNINGHAM
STANLEY J. PANIKOWSKI
ERIN P. GIBSON
RAJIV DHARNIDHARKA
RYAN W. COBB

McANDREWS, HELD & MALLOY
THOMAS J. WIMBISCUS
GREGORY C. SCHODDE
WAYNE H. BRADLEY

Attorneys for Plaintiff NETLIST, INC.

US008001434B1

(12) **United States Patent**
Lee et al.

(10) **Patent No.: US 8,001,434 B1**
(45) **Date of Patent: Aug. 16, 2011**

(54) **MEMORY BOARD WITH SELF-TESTING CAPABILITY**

(75) Inventors: **Hyun Lee**, Ladera Ranch, CA (US); **Jayesh R. Bhakta**, Cerritos, CA (US); **Soonju Choi**, Irvine, CA (US)

(73) Assignee: **NETLIST, Inc.**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 131 days.

(21) Appl. No.: **12/422,925**

(22) Filed: **Apr. 13, 2009**

Related U.S. Application Data

(60) Provisional application No. 61/044,801, filed on Apr. 14, 2008, provisional application No. 61/044,825, filed on Apr. 14, 2008, provisional application No. 61/044,839, filed on Apr. 14, 2008.

(51) **Int. Cl.**
G01R 31/28 (2006.01)
G11C 29/00 (2006.01)

(52) **U.S. Cl.** **714/733**; 714/719

(58) **Field of Classification Search** 714/718, 714/719, 736, 819, 733, 734, 30; 365/200, 365/201, 230.03, 189.011
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,660,675	A		5/1972	Andrews, Jr.
3,757,235	A		9/1973	McCormick et al.
4,305,091	A		12/1981	Cooper
4,586,168	A		4/1986	Adlhoch et al.
4,701,845	A		10/1987	Andreasen et al.
4,752,741	A		6/1988	Kim et al.
4,782,487	A		11/1988	Smelser
4,837,743	A		6/1989	Chiu et al.
4,885,799	A		12/1989	Van Horn
4,903,266	A		2/1990	Hack
4,910,597	A		3/1990	Harada et al.
4,942,556	A		7/1990	Sasaki et al.
4,987,321	A		1/1991	Toohey
5,033,048	A	*	7/1991	Pierce et al. 714/719
5,051,997	A	*	9/1991	Sakashita et al. 714/732
5,138,619	A		8/1992	Fasang et al.
5,173,906	A		12/1992	Dreibelbis et al.
5,222,066	A		6/1993	Grula et al.
5,241,503	A		8/1993	Cheng

(Continued)

OTHER PUBLICATIONS

Der-Chang et al. "*A parallel built-in self-diagnostic method for embedded memoryarrays*", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Apr. 2002, vol. 21, Issue 4, pp. 449-465.

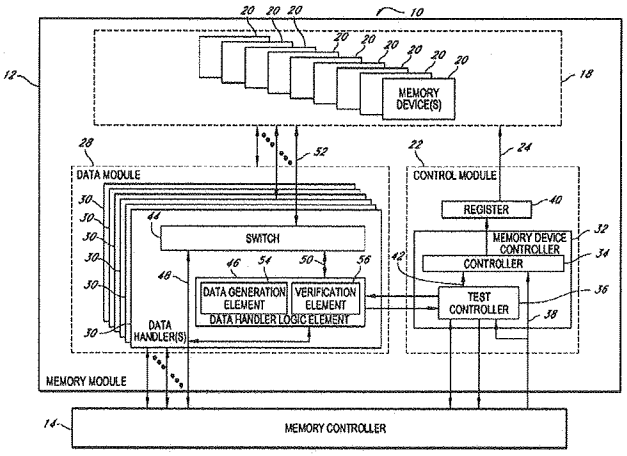
(Continued)

Primary Examiner — Phung M Chung
(74) *Attorney, Agent, or Firm* — Knobbe Martens Olson & Bear LLP

(57) **ABSTRACT**

A self-testing memory module includes a printed circuit board configured to be operatively coupled to a memory controller of a computer system and includes a plurality of memory devices on the printed circuit board, each memory device of the plurality of memory devices comprising data, address, and control ports. The memory module also includes a control module configured to generate address and control signals for testing the memory devices. The memory module includes a data module comprising a plurality of data handlers. Each data handler is operable independently from each of the other data handlers of the plurality of data handlers. Each data handler is operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and is configured to generate data for writing to the corresponding plurality of data ports.

35 Claims, 5 Drawing Sheets



Exh A

U.S. PATENT DOCUMENTS

5,304,856	A	4/1994	Rainal
5,337,254	A	8/1994	Knee et al.
5,359,235	A	10/1994	Coyle et al.
5,394,037	A	2/1995	Josephson et al.
5,430,335	A	7/1995	Tanoi
5,525,917	A	6/1996	Wong et al.
5,841,296	A	11/1998	Churcher et al.
5,914,543	A	6/1999	Scherpenberg et al.
6,000,048	A	12/1999	Krishna et al.
6,044,481	A	3/2000	Kornachuk et al.
6,070,217	A	5/2000	Connolly et al.
6,169,696	B1	1/2001	Bissey
6,194,959	B1	2/2001	Kamoshida et al.
6,216,240	B1	4/2001	Won et al.
6,467,056	B1	10/2002	Satou et al.
6,560,740	B1	5/2003	Zuraski, Jr. et al.
6,614,702	B2	9/2003	Lee
6,721,150	B1	4/2004	Guerrero, Jr.
6,812,869	B1	11/2004	Rahman et al.
6,829,728	B2	12/2004	Cheng et al.
6,918,072	B2	7/2005	Cowels et al.
6,928,024	B2	8/2005	Pfeiffer et al.
6,928,593	B1	8/2005	Halbert et al.
6,930,509	B2	8/2005	Banik
6,934,900	B1	8/2005	Cheng et al.
7,036,064	B1	4/2006	Kebichi et al.
7,053,470	B1	5/2006	Sellers et al.
7,062,696	B2	6/2006	Barry et al.
7,184,915	B2	2/2007	Hansquine et al.
7,190,210	B2	3/2007	Azrai et al.
7,203,873	B1	4/2007	Adams
7,253,652	B2	8/2007	Azimi et al.
7,284,166	B2	10/2007	Zappa et al.
2001/0048342	A1	12/2001	Yoshida et al.
2002/0000847	A1	1/2002	Taguchi
2002/0131535	A1	9/2002	Huber
2002/0140523	A1	10/2002	Park et al.
2003/0098742	A1	5/2003	Nakagawa et al.
2003/0126346	A1	7/2003	Kuo
2003/0197797	A1	10/2003	Segura
2003/0218491	A1	11/2003	Nagasue
2004/0155702	A1	8/2004	Danielsson
2004/0199843	A1	10/2004	Hansquine et al.
2005/0093620	A1	5/2005	Ho et al.
2005/0127989	A1	6/2005	Miyagi
2005/0289423	A1	12/2005	Yabuta
2006/0082383	A1	4/2006	Choi
2006/0107156	A1	5/2006	Lee et al.
2006/0140015	A1	6/2006	Kasamsetty
2006/0144015	A1	7/2006	Cash et al.
2006/0147217	A1	7/2006	Hahin et al.
2006/0192653	A1	8/2006	Atkinson et al.
2006/0242458	A1	10/2006	Feldman et al.
2006/0262586	A1	11/2006	Solomon et al.
2006/0271748	A1	11/2006	Jain et al.
2007/0030814	A1	2/2007	Shin et al.
2007/0058471	A1	3/2007	Rajan et al.
2007/0079199	A1	4/2007	Chorn et al.
2007/0109707	A1	5/2007	Honda
2007/0152743	A1	7/2007	Keeth et al.
2007/0204075	A1	8/2007	Rajan et al.
2007/0223296	A1	9/2007	Miller et al.

OTHER PUBLICATIONS

Bayard "On the LTI properties of adaptive feed forward systems with tapdelay-line regressors", IEEE Transactions on Signal Processing, May 1999, vol. 47, Issue 5, pp. 1288-1296.

Mutoh et al. "EMI Noise controlling methods suitable for electric vehicle drive systems", Industrial Electronics Society, 2004, IECON 2004, Nov. 2-6, 2004, vol. 1, pp. 963-968.

Sekiguchi et al "Low-noise, high-speed datat transmission using a ringing-canceling output buffer", IEEE Journal of Solid-State Circuits, Dec. 1995, vol. 30, Issue 12, pp. 1569-1574.

* cited by examiner

FIG. 1

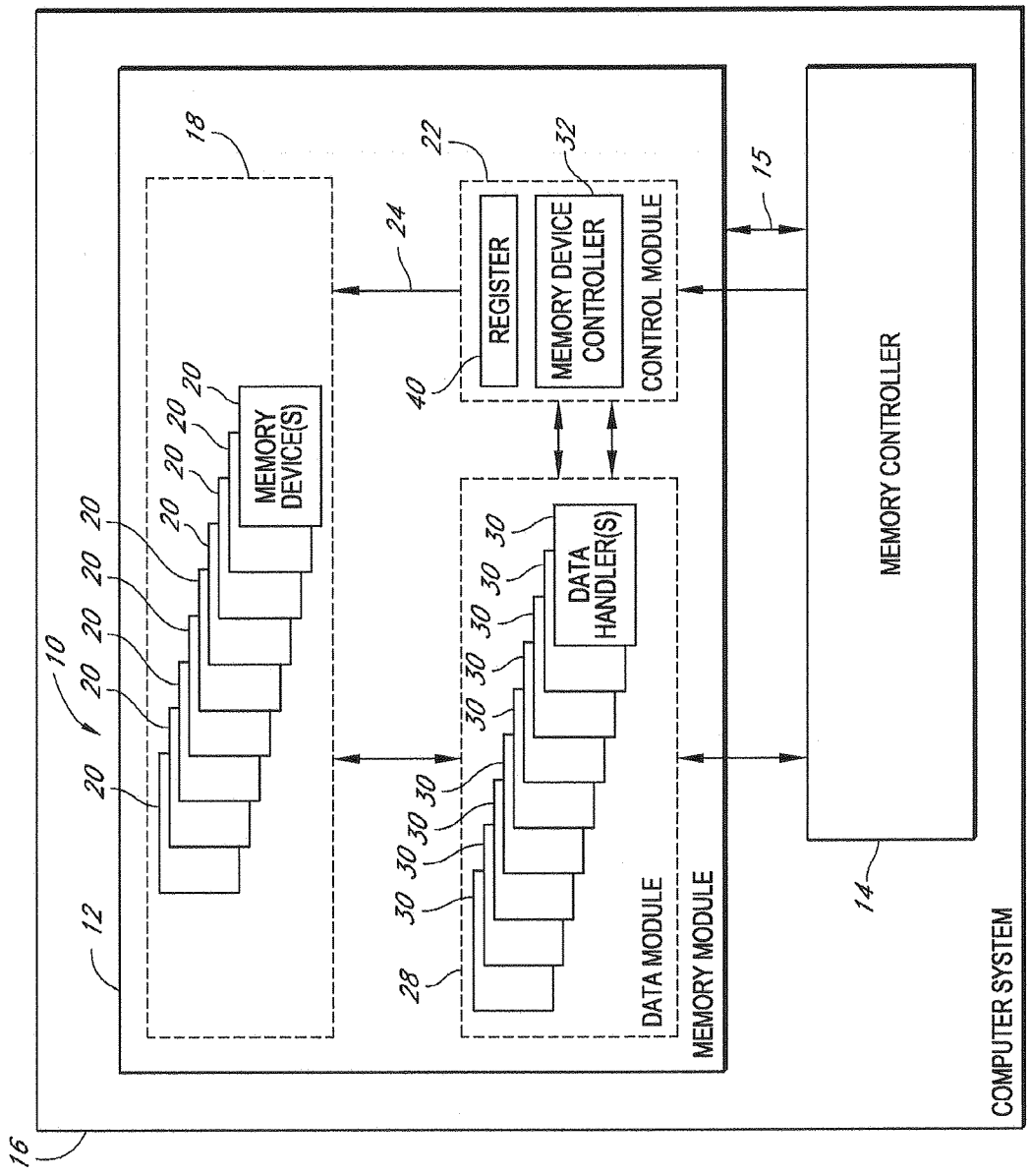
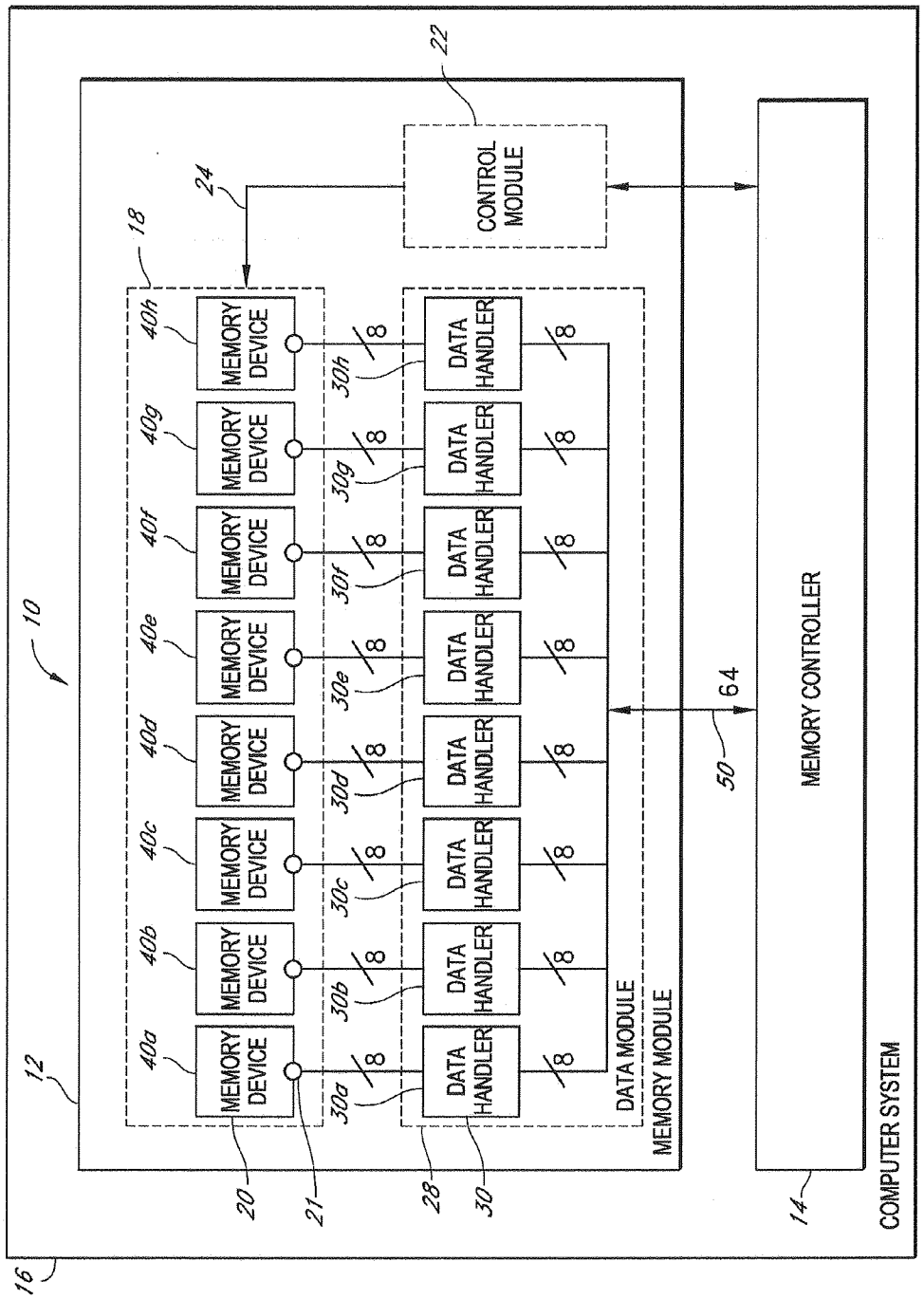

COMPUTER SYSTEM
MEMORY CONTROLLER
MEMORY MODULE
DATA MODULE
CONTROL MODULE
REGISTER
MEMORY DEVICE CONTROLLER
DATA HANDLER(S)
MEMORY DEVICE(S)
10
12
14
15
16
18
20
22
24
28
30
32
40

FIG. 2


10
12
16
18
20
21
22
24
28
30
50
64
14
CONTROL MODULE
MEMORY CONTROLLER
COMPUTER SYSTEM
MEMORY MODULE
DATA MODULE
MEMORY DEVICE
DATA HANDLER
40a
40b
40c
40d
40e
40f
40g
40h
30a
30b
30c
30d
30e
30f
30g
30h
8

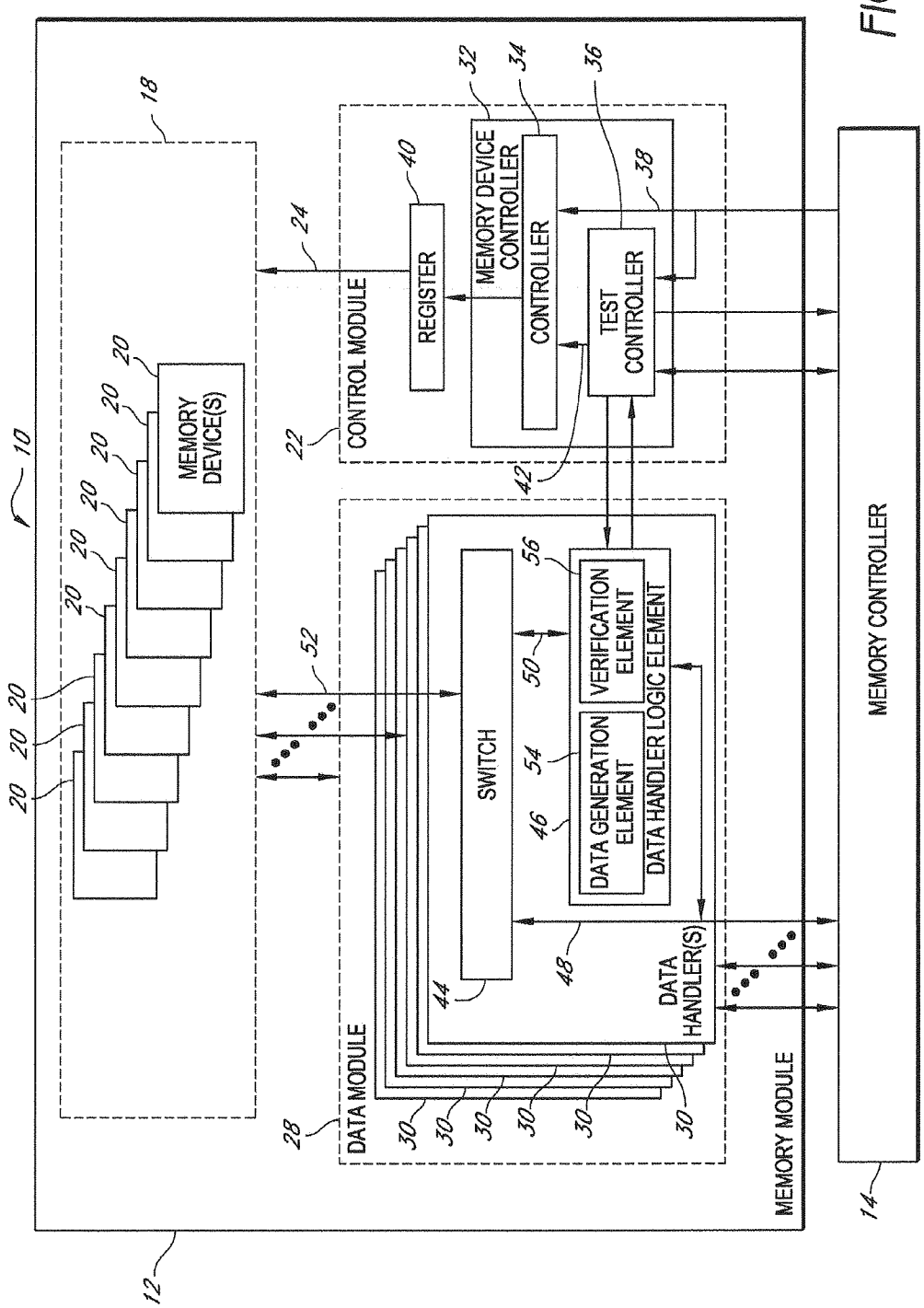

10
12
14
18
20
MEMORY DEVICE(S)
22
CONTROL MODULE
24
REGISTER
40
MEMORY DEVICE CONTROLLER
32
CONTROLLER
34
TEST CONTROLLER
36
38
42
MEMORY CONTROLLER
MEMORY MODULE
28
DATA MODULE
30
52
SWITCH
44
50
VERIFICATION ELEMENT
56
54
DATA GENERATION ELEMENT
46
DATA HANDLER LOGIC ELEMENT
48
DATA HANDLER(S)

FIG. 3

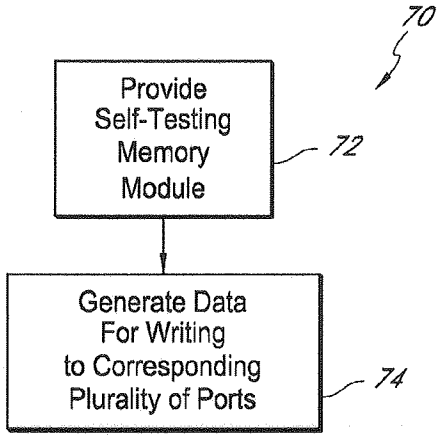

70
Provide Self-Testing Memory Module
72
Generate Data For Writing to Corresponding Plurality of Ports
74

FIG. 4

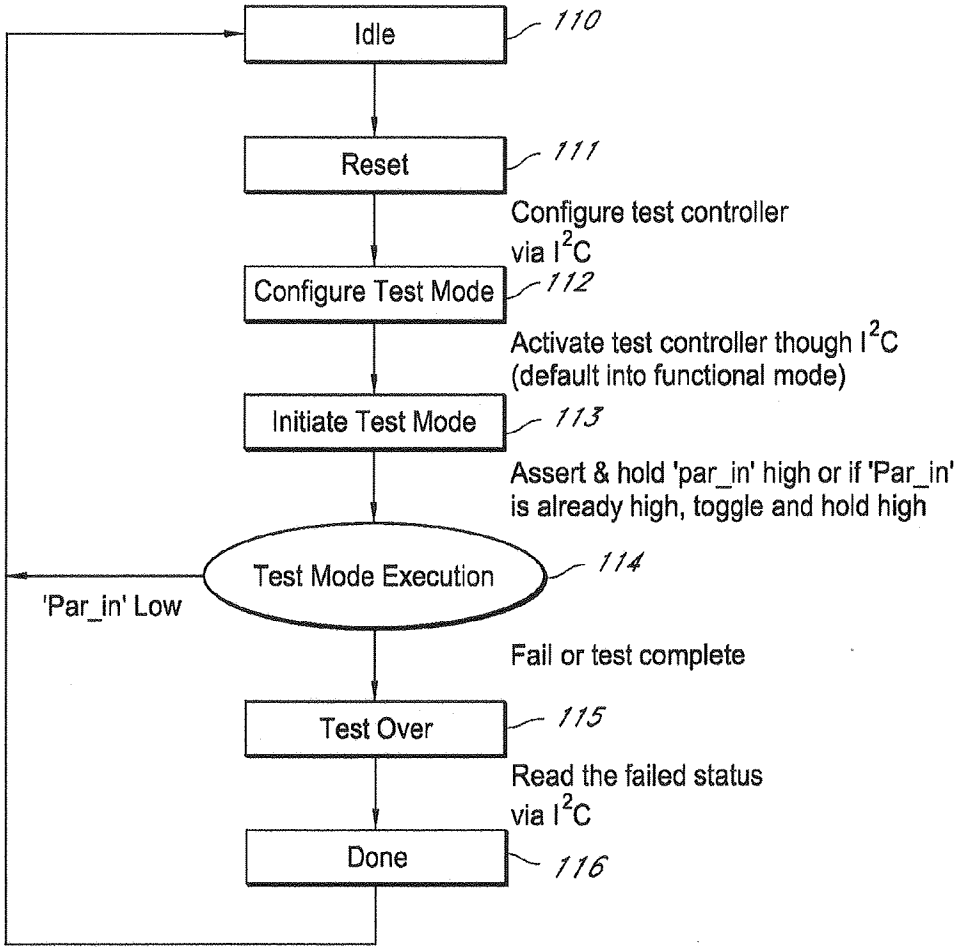

Idle
110
Reset
111
Configure test controller via I2C
Configure Test Mode
112
Activate test controller though I2C (default into functional mode)
Initiate Test Mode
113
Assert & hold 'par_in' high or if 'Par_in' is already high, toggle and hold high
Test Mode Execution
114
'Par_in' Low
Fail or test complete
Test Over
115
Read the failed status via I2C
Done
116

FIG. 5

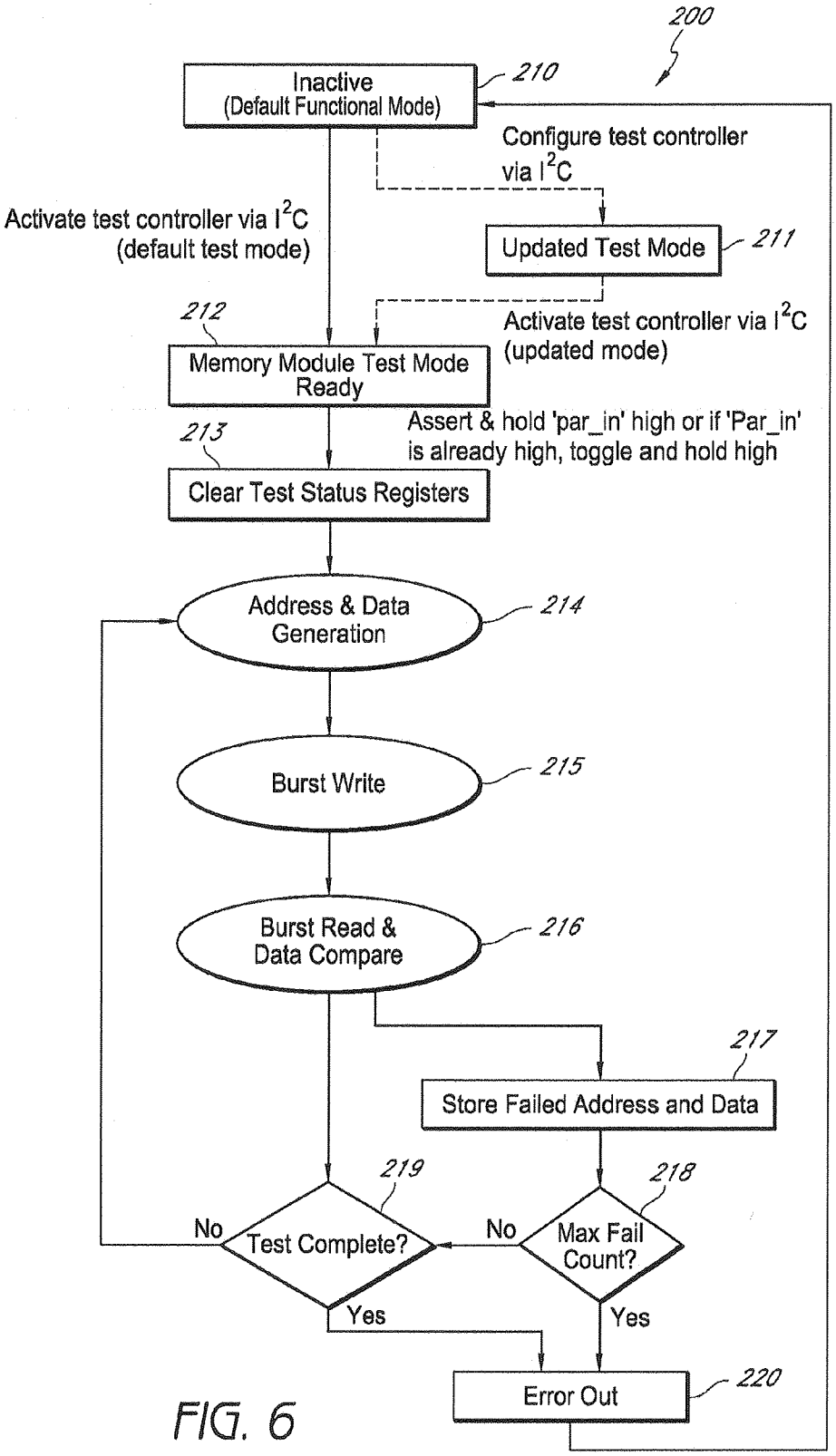

200
Inactive
(Default Functional Mode)
210
Configure test controller via I2C
Activate test controller via I2C (default test mode)
Updated Test Mode
211
212
Activate test controller via I2C (updated mode)
Memory Module Test Mode Ready
Assert & hold 'par_in' high or if 'Par_in' is already high, toggle and hold high
213
Clear Test Status Registers
Address & Data Generation
214
Burst Write
215
Burst Read & Data Compare
216
217
Store Failed Address and Data
219
218
No
Test Complete?
No
Max Fail Count?
Yes
Yes
Error Out
220

FIG. 6

MEMORY BOARD WITH SELF-TESTING CAPABILITY

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims the benefit of priority from U.S. Provisional Application No. 61/044,801, filed Apr. 14, 2008, from U.S. Provisional Application No. 61/044,825, filed Apr. 14, 2008, and from U.S. Provisional Application No. 61/044,839, filed Apr. 14, 2008, each of which are incorporated in their entirety by reference herein. This application is related to U.S. patent application Ser. No. 12/422,912, filed on even date herewith and entitled "Self-Adjusting Damper", and to U.S. application Ser. No. 12/422,853, filed on even data herewith and entitled "Circuit Providing Load Isolation and Noise Reduction", both of which are incorporated in their entirety by reference herein.

BACKGROUND

1. Field

The present invention relates to self-testing electronic modules and, more particularly, to self-testing electronic memory modules.

2. Description of the Related Art

The failure of memory components in an electronic system may result in the loss of valid data. Therefore, it is important to ensure proper memory operation in an electronic system. Memory integrated circuits ("memory chips") often go through a series of tests at various stages of system manufacture. Once memory chips are deployed in a system, they also generally go through a system level memory test each time the system is booted. In addition, memory chips may undergo a parity checking process during normal system operation.

There are typically at least three test phases which memories undergo during system manufacture. Each phase generally tests for memory defects and for the correct operation of the input/output interface. The first test phase is typically conducted by the memory chip manufacturer and generally involves checking for bit failures, correct memory access speed, etc. The second test phase is typically done by memory module manufacturers and generally involves testing the signal quality, the noise susceptibility, and the operational speed of the memory module as a single unit. The second test phase may also include checking for bit failures in individual memory chips. The third phase is usually carried out by the system manufacturer. During the third phase, the interaction of the memory subsystem with other components in the system is tested. During the third phase, the individual memory module operation is also tested again and the memory array is checked for defects. Because of the significant amount of testing that memories undergo during the manufacturing process, there is generally substantial test cost and test time associated with ensuring the proper memory operation. This test cost and test time translate into an increase in system cost and a decrease in system performance.

There are a number of memory test methodologies that employ either external test hardware, embedded self-test logic ("MBIST"), or both. However, the usefulness of these test methodologies is limited due to the high cost and other limitations associated with them. For instance, external test hardware such as automatic test equipment ("ATE") is very expensive. Moreover, the development time and cost associated with implementing MBIST is relatively high. These costs and limitations are especially significant when testing dynamic random access memory ("DRAM"). For example,

technological developments, such as increases in DRAM speed, may require manufacturers to upgrade ATE machines relatively frequently. In addition, MBIST in DRAM chips generally cannot be fully utilized for system level testing of memory boards.

Because of the increasing cost, complexity, and time involved with fully testing DRAM chips, DRAM manufacturers often provide "effectively tested" ("ETT") DRAM chips to memory module manufacturers at a lower price rather than providing fully tested DRAM chips. Memory module manufacturers often prefer the ETT DRAM chips mainly due to their greater availability. Memory module manufacturers who receive ETT DRAM chips then have to assume a part of the responsibility of validating the DRAM chips, adding to the complexity of the memory module test process.

SUMMARY

In certain embodiments, a self-testing memory module is provided comprising a printed circuit board configured to be operatively coupled to a memory controller of a computer system. The memory module may further comprise a plurality of memory devices on the printed circuit board. Each memory device of the plurality of memory devices comprises data, address, and control ports. In certain embodiments, the memory module includes a control module configured to generate address and control signals for testing the memory devices and a data module comprising a plurality of data handlers. Each data handler in certain embodiments is operable independently from each of the other data handlers of the plurality of data handlers. Each data handler may be operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and configured to generate data for writing to the corresponding plurality of data ports.

A self-testing memory module is provided in certain embodiments comprising a printed circuit board configured to be operatively coupled to a memory controller of a computer system. The memory module may further comprise a plurality of memory devices on the printed circuit board. Each memory device of the plurality of memory devices comprises data, address, and control ports in some embodiments. The memory module of certain embodiments includes a control module configured to generate address and control signals for testing the memory devices. In certain embodiments, the memory module comprises a data module comprising at least one data handler operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and configured to generate cyclic data for writing to the corresponding plurality of data ports.

In certain embodiments, a method of self-testing a memory module includes providing a self-testing memory module. The memory module comprises a printed circuit board configured to be operatively coupled to a memory controller of a computer system. The memory module further comprises a plurality of memory devices on the printed circuit board. Each memory device of the plurality of memory devices comprising data, address, and control ports. In certain embodiments, the memory module comprises a control module configured to generate address and control signals for testing the memory devices. The memory module may further comprise a data module comprising a plurality of data handlers. Each data handler may be operable independently from each of the other data handlers of the plurality of data handlers and operatively coupled to a corresponding plurality of the data ports. The method of certain embodiments further

includes generating, by each of the data handlers, data for writing to the corresponding plurality of data ports.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of an example self-testing memory module in accordance with certain embodiments described herein.

FIG. **2** is a block diagram of an example self-testing memory module including eight memory devices and eight data handlers in accordance with certain embodiments described herein.

FIG. **3** is a block diagram of an example data handler module and an example control module in accordance with certain embodiments described herein.

FIG. **4** is a flowchart of an example method of self-testing a memory module in accordance with certain embodiments described herein.

FIG. **5** is a flow diagram illustrating the self-testing operation of an example memory module in accordance with certain embodiments described herein.

FIG. **6** is a flow diagram illustrating the self-testing operation of an example memory module in accordance with certain embodiments described herein.

DETAILED DESCRIPTION

Certain embodiments described herein present a self-testing electronic system, such as for example, a self-testing electronic memory module. Some embodiments described herein present a self-testing memory module that is populated with ETT DRAM chips. Certain embodiments described herein present a self-testing registered dual in-line memory module ("RDIMM"). Some embodiments described herein present a self-testing RDIMM that does not require any additional pins other than the pins on the standard JEDEC RDIMM connector. For example, the memory module may utilize the address and control signals (e.g., address and control signals generated by the memory module to test the memory module) along with a test signal to enable and execute a self testing function.

Certain embodiments described herein present a self-testing memory module that includes a control module and a data module which can generate memory addresses and data according to the JEDEC standard memory protocol.

Some embodiments described herein presents a self-testing memory module that can be configured through an I^2C interface and that allows test results to be read out through the I^2C interface. Certain embodiments described herein present a self-testing memory module that allows a test function to be configured, controlled, and/or executed without substantial system memory controller involvement. Some embodiments present a self-testing memory module that can be tested without any external test equipment. For example, certain embodiments present a self-testing memory module that can be tested without any system driven test procedure. Various embodiments described herein present a self-testing memory module that can be self-tested at a target system speed. Certain embodiments described herein present a self-testing memory module that can generate data (DQ) and data-strobe (DQS) signals with wave characteristics that resemble the wave characteristics of DQ and DQS signals from a system memory controller.

MBIST is commonly used to test memories embedded in application specific integrated circuits ("ASICs") or system on chip integrated circuits ("SoCs") such as, for example, advanced memory buffers ("AMBs"). The MBIST implementation usually includes three distinct functional blocks: the address/control generator, the data generator/checker (sometimes referred to as a signature analyzer), and the test interface controller/register ("JTAG"). The test instructions and test patterns are generally loaded to the address/control generator and the data generator/checker through the JTAG interface. The test results are generally read out through the JTAG interface.

In memory module applications, there are obstacles associated with using MBIST. One of these obstacles is the large number of signals used to interface the three MBIST functional blocks. The inter-block timing constraints can present another obstacle. In addition, the ability to control the test is limited. For example, synchronizing the address/or control signals with the data signal can be difficult. The relatively large amount of information to be gathered and stored while testing the memory presents yet another obstacle.

Unlike on an ASIC, in which the three MBIST functional blocks are in a single chip, the MBIST functional blocks on a memory module generally would be segregated into multiple chips on the memory module due to physical and electrical limitations and requirements. This makes implementing MBIST on a memory module difficult because, while there is virtually no limit on the number of available interface signals among the three MBIST functional blocks in an ASIC, the memory module can support only a limited amount of interface signals between memory chips. In addition, because the MBIST functional blocks are spread out to multiple chips on a memory module, the inter-block signal delay is generally much longer on a memory module than on an ASIC. It is also generally not feasible to implement cross-checking logic that operates the three MBIST functional blocks in lock-operational block. This is due to the limitation on the number of interface signals and to the relatively long inter-block signal delay time on a memory module versus on an ASIC.

In some cases such as where a memory module includes an AMB, the self-test logic (MBIST) implemented in the AMB includes command and address generation logic in addition to a data generator and checker. Each of these functional blocks may be implemented on a single physical AMB device (e.g., a single integrated circuit package). However, because all of the self-test command, address and data signals are combined in one physical area of the memory module, a memory module (e.g., DIMM) level routing problem can occur, making it difficult to route the self-test signals on the memory module and resulting in performance degradation and/or implementation difficulty. In addition, the data width of self-test logic of the memory module will be limited to the data width of the AMB (e.g., to the data width of the self-test logic implemented on the AMB, to the number of available ports on the AMB, etc.). As such, implementing memory module test logic on an AMB is not flexible (e.g., to changes in the data width of the memory module) and generally supports only memory modules having certain predetermined, fixed data widths.

Finally, in most cases, since an ASIC cannot be repaired, the ASIC MBIST is generally only capable of detecting and reporting the pass/fail status of memory tests. For memory module test results, on the other hand, it is generally advantageous to include both addresses of the memory locations where failures occur and the data patterns that were read back from the failed memory locations. This type of reporting can help to facilitate the repair of the memory module by, for example, allowing for the identification and replacement of failed components.

FIG. **1** is a block diagram of an example self-testing memory module **10** in accordance with certain embodiments

described herein. The memory module **10** includes a printed circuit board **12** configured to be operatively coupled to a memory controller **14** of a computer system **16**. The memory module **10** further includes a plurality of memory devices **18** on the printed circuit board (PCB) **12**, each memory device **20** of the plurality of memory devices **18** comprising data, address, and control ports. The memory module **10** comprises a control module **22** configured to generate address and control signals **24** for testing the memory devices **18**. The memory module **10** also includes a data module **28** comprising a plurality of data handlers **30**. Each data handler **30** is operable independently from each of the other data handlers **30** of the plurality of data handlers **28** and is operatively coupled to a corresponding plurality of the data ports of one or more of the plurality of memory devices **18**. For example, each of the data handlers **30** may be operatively coupled to (e.g., logically and/or electrically coupled to) the corresponding plurality of data ports. Each data handler **30** is further configured to generate data for writing to the corresponding plurality of data ports. The memory module **10** may further include an I^2C interface **15** in certain embodiments.

As described more fully below, in certain embodiments the data module **28** generates test data patterns to write to the plurality of memory devices **18** of the memory module **12** and checks the data patterns read or received back from the plurality of memory devices **18** for agreement with corresponding data patterns that are expected to be read back from the plurality of memory devices **18**. For example, in one embodiment, the data module **28** generates cyclic patterns to write to the plurality of memory devices **18**. In some embodiments, the data module **18** also isolates the data path from the system board of the computer system **16** to the plurality of memory devices **18** while the memory module **10** is not accessed by the computer system **16**. For example, the data module **28** may isolate the data path from the system board to the plurality of memory devices **18** when the memory module **10** is in a self-testing mode. The control module **22** may include, for example, a dual input register (e.g., the memory device controller **32** described more fully below) for registering address and control signals coming from either self-testing logic (e.g., from the test controller **36** described more fully below) or from the memory controller **14** on the system board. In some embodiments, during testing, the control module **22** generates address and control signals **24** associated with memory locations to be tested and the data module **28** generates corresponding test data patterns and provides them to the appropriate memory devices **20**. For example, the data module **28** may receive a write command from the control module **22** and provide data to be written to certain locations in the memory devices **20** during a write operation. The data module **28** may then receive a read command to read back the data from those locations and check the read data for agreement with the expected data. If there is a mismatch between the read data and the expected data, the data module **28** may, for example, store the failure information (e.g., the failed data word) and inform the control module **22** about the failure. The control module may save the address of the memory location where the failure occurred.

In certain embodiments, the memory module **10** is configured to be operated in a test mode in which the control module **22** selectively inputs the address and control signals to the address and control ports of the plurality of memory devices **18**. Moreover, in the test mode, each of the data handlers **30** write the data generated by the data handler **30** to the corresponding plurality of data ports by selectively inputting data signals to the data ports of the plurality of memory devices **18**. The data module **28** and/or the control module **22** of certain embodiments are configured to test the plurality of memory devices **18** at the normal operating speed of the memory devices **20**. For example, the data module **28** and/or the control module **22** are configured to provide memory signals (e.g., data, address and control signals) according the operating specification of the memory devices **20**. In some embodiments, the control module **22** and the data module **28** produce memory addresses, control and/or data signals according to the JEDEC standard memory protocol. In some embodiments, for example, the control module **22** and the data module **28** generate the memory interface signals with proper edge relationships based on the JEDEC standard. In certain embodiments, the test speed, for example, may be defined by the speed of the clock (e.g., the system clock). The address sequences and/or the data patterns of certain embodiments may be programmable either through the I^2C interface or they may be defaulted to pre-defined values.

In certain embodiments, data module **28** is configured to generate data signals with programmable slew rates and/or with variable peak values. In one embodiment, for example, the data module **28** is also able to generate data (DQ) and data-strobe (DQS) signals with programmable slew rates and programmable peak values so that the characteristics of the signals generated by the data module **28** generally correspond to the characteristics DQ and DQS signals generated by the system memory controller **14**.

In some embodiments, the data module **28** and/or the control module **22** are configured to test the plurality of memory devices **18** under non-normal conditions. For example, the data module **28** and/or control module **22** may be configured to provide signals having frequencies which are higher or lower than the normal operating frequencies of the memory devices **20**.

In certain embodiments, the memory module **10** has a memory capacity of 512-MB, 1-GB, 2-GB, 4-GB, or 8-GB. Other memory capacities are also compatible with certain embodiments described herein. In addition, memory modules **100** having widths of 4 bytes, 8 bytes, 16 bytes, 32 bytes, or 32 bits, 64 bits, 128 bits, 256 bits, as well as other widths (in bytes or in bits), are compatible with embodiments described herein. In certain embodiments, the PCB **12** has an industry-standard form factor. For example, the PCB **12** can have a low profile (LP) form factor with a height of 30 millimeters and a width of 133.35 millimeters. In certain other embodiments, the PCB **12** has a very high profile (VHP) form factor with a height of 50 millimeters or more. In certain other embodiments, the PCB **12** has a very low profile (VLP) form factor with a height of 18.3 millimeters. Other form factors including, but not limited to, small-outline (SO-DIMM), unbuffered (UDIMM), registered (RDIMM), fully-buffered (FBDIMM), mini-DIMM, mini-RDIMM, VLP mini-DIMM, micro-DIMM, and SRAM DIMM are also compatible with certain embodiments described herein. For example, in other embodiments, certain non-DIMM form factors are possible such as, for example, single in-line memory module (SIMM), multi-media card (MMC), and small computer system interface (SCSI).

In certain embodiments, the plurality of memory devices **18** of the memory module **10** may be arranged as ranks, each rank of memory generally having a bit width. In certain embodiments, each rank may comprise an independent set of memory devices **20** of the plurality of memory devices **18** that can be accessed by the memory controller **14** to access the full bit-width of the memory bus of the memory module **10**. For example, a memory module **10** in which each rank of the memory module is 64 bits wide is described as having an "x 64" organization. Similarly, a memory module **10** having

72-bit-wide ranks is described as having an "x 72" organization. The number of memory devices **20** and corresponding memory capacity of a memory module **10** can be increased by increasing the number of memory devices **20** per rank or by increasing the number of ranks. For example, a memory module with four ranks with each rank having N 512-MB memory devices **20** has double the memory capacity of a memory module with two ranks with each rank having N 512-MB memory devices **20** and four times the memory capacity of a memory module with one rank with each rank having N 512-MB memory devices **20**. During operation, the ranks of a memory module **10** may be selected or activated by control signals that are received from a component of the system (e.g., a system memory controller **14** or a local memory controller of the memory module **10**). Examples of such control signals include, but are not limited to, rank-select signals, also called chip-select signals. In certain other embodiments, the memory module **10** comprises only one rank of memory devices **20**.

As discussed, the PCB **12** may include at least one connector (not shown) configured to operatively couple the memory module **10** to the memory controller **14** of the computer system **16**. The computer system **16** may include a host computer system. For example, the memory module is electrically coupled, logically coupled, or both, with the memory controller **14**. Examples of host computer systems **108** include, but are not limited to, blade servers, IU servers, personal computers (PCs), data storage systems and other applications in which space is constrained or limited. The memory controller **14** may comprise a disk controller of the computer system **16**, for example. The memory controller **14** may be mounted on a system board of the host computer **16**. The connector can comprise a plurality of edge connections which fit into a corresponding slot connector of the host system **16**. The connector of certain embodiments provides a conduit for power voltage as well as data, address, and control signals between the memory module **10** and the host system **16**. For example, the connector can comprise a standard DDR2, DDR3, and other future generation edge connectors. Additionally, in certain embodiments, more than one memory module **10** is coupled to the host system **16**.

The plurality of memory devices **18** on the PCB **12** may include one or more volatile memory components. For example, the plurality of memory devices **18** of certain embodiments comprises two or more dynamic random-access memory (DRAM) elements **20**. Types of DRAM devices **20** compatible with certain embodiments described herein include, but are not limited to, DDR, DDR2, DDR3, and synchronous DRAM (SDRAM). The memory devices **18** may comprise other types of memory elements such as static random-access memory (SRAM). In addition, volatile memory devices **20** having bit widths of 4, 8, 16, 32, as well as other bit widths, are compatible with certain embodiments described herein. Memory devices **20** compatible with certain embodiments described herein have packaging which include, but are not limited to, thin small-outline package (TSOP), ball-grid-array (BGA), fine-pitch BGA (FBGA), micro-BGA (μBGA), mini-BGA (mBGA), and chip-scale packaging (CSP). The plurality of memory devices **18** may further include one or more non-volatile memory devices **20**, such as, for example, flash memories. The plurality of memory devices **18** of certain embodiments may include both volatile and non-volatile memory devices **20**. For example, the plurality of memory devices **18** may include one or more of DRAM, SRAM, and/or flash memory devices in some embodiments.

Each data handler **30** is operable independently from each of the other data handlers **30** of the plurality of data handlers **28**. For example, each data handler **30** is configured to write to and/or read from the corresponding plurality of data ports of one or more of the memory devices **20** without being in communication any of the other data handlers **30** or other data ports of the memory devices **20**. As such, each data handler **30** can be used to generally independently test a portion of the memory space of the memory module **10**. For example, each data handler **30** may be used to independently test one memory device **20** of the memory module **10**. In such a configuration, the corresponding plurality of data ports of each data handler **30** may comprise each data port of the corresponding memory device **20**. In other embodiments, each data handler **30** may be used to test a segment of one memory device **20**, more than one memory device **20**, segments more than one memory device **20**, or any combination or sub-combination thereof. Because each of the data handlers **30** is operable independently of each of the other data handlers **30**, the data handlers **30** are generally modular. As such, modifications in the configuration of the memory module **10** (e.g., changes in the bit-width of the memory bus, changes in the number of memory devices **20**, etc.) may be less complicated to accommodate than in other types of self-testing memory modules **10**. For example, where a new memory device **20** or set of memory devices **20** is added to the memory module **10**, the change may be generally accommodated by adding a corresponding data handler **30**. The change may be accommodated without having to implement a major reorganization of the memory module **10** or the self-testing logic of the memory module **10**, for example.

Each data handler **30** is further configured to generate data for writing to the corresponding plurality of data ports. FIG. **2** is a block diagram of an example self-testing memory module **10** including eight memory devices **20** (e.g., memory devices **40***a*-**40***h*) and a data module **28** comprising eight data handlers **30** (e.g., data handlers **30***a*-**30***h*) in accordance with certain embodiments described herein. Each of the memory devices **20** includes an eight bit output data word and eight corresponding data ports. In addition, the system memory bus **50** between the memory controller **14** and the example memory module **10** is 64 bits wide and each of the data handlers **30** receives an eight bit segment of the system memory bus **50**. Each of the data handlers **30** is operatively coupled to a corresponding plurality of data ports **21** of a corresponding one of the memory devices **20**. As such, the data handlers **30** may be operatively coupled (e.g., electrically and/or logically coupled or connected) to the eight data ports **21** of one of the corresponding memory devices **20**. For example, the data handler **30***a* may be operatively coupled to the eight data ports **21** of the memory device **40***a* of FIG. **2**.

The configuration shown in FIG. **2** is for the purposes of illustration and is not intended to be limiting. For example, while the example memory module **10** of FIG. **2** includes an equal number of memory devices **20** and data handlers **30**, other configurations are possible. In some configurations there are more memory devices **20** than data handlers **30** or vice versa. Moreover, the one or more data handlers **30** may be operatively coupled to a subset of the data ports **21** of one the memory devices **20** instead of all of the data ports **21** of one of the memory devices **20**. In other embodiments, one or more data handlers **30** may be operatively coupled to a subset or all of the data ports **21** of more than one of the memory devices **20**. For example, in one embodiment, each of the data handlers **30** are operatively coupled to all of the data ports of two memory devices **20**.

In certain embodiments, the plurality of data handlers **28** comprises at least two physically separate components mounted on the PCB **12**. For example, the plurality of data handlers **28** may include at least two physically separate integrated circuit packages. The physically separate integrated circuit packages are mounted on different portions of the PCB **12** in some embodiments. For example, each of the eight data handlers **30***a*-**30***h* shown in FIG. **2** may include physically separate integrated circuit packages mounted on different portions of the PCB **12**. While eight data handlers **30** are shown in FIG. **2**, other numbers of data handlers **30** are possible including fewer or more than eight.

In certain embodiments, each of the plurality of data handlers **30** is positioned on the PCB **12** proximate to the corresponding plurality of data ports. For example, each data handler **30** of certain embodiments is positioned closer to the corresponding plurality of data ports **21** than the data handler **30** is to the other data ports **21** of the plurality of memory devices **18**. For example, the data handler **30***a* is positioned closer to the corresponding plurality of data ports **21** of the memory device **40***a* than to the other data ports **21** of the other memory devices **40***b*-**40***h*.

FIG. **3** is a block diagram an example data module **28** and an example control module **22** in accordance with certain embodiments described herein. The control module **22** can be configured to generate address and control signals **24** for testing the plurality of memory devices **18**. In some embodiments, the control module **22** includes a control mixer element **32**. The control mixer element **32** may include a memory device controller **34** (e.g., a DRAM controller) and a test controller **36**. In certain embodiments, the control mixer element **32** generally controls the address and the control signals for the self-testing function.

In certain embodiments, the memory device controller **34** generally pre-processes address and control information before it sends the information to a register **40**. In one embodiment, the memory device controller **34** receives signals **38** (e.g., address and control signals) from the system memory controller **14** and signals **42** (e.g., address and controls signals) from the test controller **36**. The control module **22** of certain embodiments is configured to selectively input to the address and control ports of the plurality memory devices **18** either the address and control signals **38** from the system memory controller **14** or the address and control signals **42** from the control module **22** (e.g., from the test controller **36**). For example, the memory device controller **34** may send either the signals **38** from the system memory controller **14** or, alternatively, the signals **42** from the test controller **36**, to the register **40** depending on whether the memory module **10** is in normal (non-test) mode or in a test mode, respectively. In one embodiment, the memory device controller **34** generates the address and control signals for memory device (e.g., DRAM device) operations. The test controller **36** controls the generation of the address and control signal sequences to be used during the self-testing operation of the memory module **10** and also communicates with the data module **28**. The control module **22** may be implemented in the control register of the memory module **10** in certain embodiments. In various embodiments, the control module **22** includes discrete logic, one or more application-specific integrated circuit (ASICs), one or more microprocessors, one or more field-programmable gate arrays (FPGAs), or one or more computer-programmable logic device (CPLDs).

The data module **28** and the subcomponents thereof (e.g., the data handlers **30**) may be in communication with one or more of the memory devices **20**, the control module **22**, and the memory controller **14**. In certain embodiments, the data module **28** comprises a plurality of data handlers **30**. In other embodiments the data module **28** includes at least one data handler **30**. Each of the data handlers **30** of certain embodiments comprises a switch **44**. For example, the switch **44** may include a data multiplexer/demultiplexer ("data mux/demux"). The switch **44** may provide a bi-directional data multiplexer function. In certain embodiments, the switch **44** is configured to selectively input to the corresponding plurality of data ports either data signals **48** from the system memory controller **14** or data signals **50** from the data handler logic element **46**. The switch **44** of certain embodiments may further be configured to receive data signals **52** (e.g., during a read operation) from the plurality of memory devices **18** and to propagate the data signals **52** to the data handler logic element **46** and/or the memory controller **14**. In some embodiments, for example, the switch **44** selectively inputs the data signals **48** to be written to the plurality of memory devices **18** from the system memory controller **14** when the memory module **10** is a normal (non-test mode) mode and, alternatively, inputs the data signals **50** from the data handler logic element **46** during a test mode. While the switch **44** is shown as being included in the data handler **30** in the example of FIG. **3**, other configurations are possible. For example, in other embodiments the switch **44** may be logically and/or physically separated from the data handler module **28** and/or the data handlers **30**.

Each of the data handlers **30** of certain embodiments further includes a data handler logic element **46**. The data handler logic element **46** of certain embodiments comprises a data generation element **54** and a verification element **56**. The data generation element **54** may be configured to generate data signals (e.g., patterns of data signals) for writing to the corresponding plurality of data ports, for example. The data signals and/or patterns of data signals may be based on information (e.g., programming or configuration information) the data handler logic element **46** receives from the control module **22**, for example. The data may be cyclic data in some embodiments or non-cyclic data in other embodiments. For example, the cyclic data may comprise at least one predetermined pattern of data which repeats or is cycled two or more times. In various embodiments, the data comprises one or more incrementing patterns or decrementing patterns, for example. In other embodiments, the data comprises a pattern which alternates each bit on successive memory writes. For example, a memory write comprising one or more hexadecimal "A" characters (each corresponding to a four-bit binary word of "1010") may be followed by a memory write comprising one or more hexadecimal "5" characters (each corresponding to a four-bit binary word of "0101"). The data may be generated in a variety of ways. In one embodiment, the data is generated based on a current write address value. For example, in one example configuration, on a first write cycle, hexadecimal "A's" are generated and written to even address locations and hexadecimal "5's" are generated and written to odd address locations, and on a second write cycle, "5's" are written to even addresses and "A's" are written to odd addresses, and this pattern repeats in subsequent cycles. The data may be generated based previously written data (e.g., inverting each of the bits of a previously written data word) in some embodiments. In general, any manner of generating a cyclic or otherwise deterministic data pattern may be compatible with embodiments described herein. In other embodiments, random or pseudorandom data may be generated and written to the corresponding plurality of data ports. For example, a linear feedback shift register (LFSR) may be used in some embodiments. In addition, the data patterns may be programmable. For example, the data patterns may be pro-

grammable based on information received by the data generation element **54** from the memory controller **14** (e.g., through the control module **22**), or from the control module **22**.

The plurality of data handlers **30** are further configured to read data from the corresponding plurality of data ports. For example, the verification element **56** may be configured to receive data from the corresponding plurality of data ports (e.g., through the switch **44** during a test mode). The verification element **56** may further be configured to check for failures in the operation of the plurality memory devices **18** by verifying that data read from the corresponding plurality of data ports corresponds to the data generated by the data handler **30** and written to the corresponding plurality of data ports.

In certain embodiments, the verification element **56** is configured to perform the verification without storing a copy of the data written to the corresponding plurality of data ports or accessing a stored copy beyond the data read from the plurality of memory devices **18**. For example, the verification element **56** does not store or access a copy of the data that is written to the corresponding plurality of data ports except for the data stored and read back from the plurality of memory devices **18**. As such, the memory module **10** of certain embodiments advantageously does not require separate memory for storing duplicate copies of test data that is written to the plurality of memory devices **18** for later comparison. For example, the verification element **56** may calculate comparison data and may compares the comparison data to the data read from the corresponding plurality of data ports. In certain embodiments, the comparison data comprises data which expresses the data or values expected to be received from the plurality of memory devices **18** if the write, store, and read processes of the data using the data module **28** and the plurality of memory devices **18** are performed correctly or as expected. The calculation may be performed simultaneously or substantially simultaneously with receiving the data read from the corresponding plurality of data ports in certain embodiments. In other embodiments, the calculation is performed either before or after receiving the data. Other configurations are possible. For example, in one embodiment, the verification element **56** does store a separate copy of the data written to the corresponding plurality of data ports upon writing the data and compares the separate copy to the read data received from the plurality of memory devices **18**.

The verification element **56** of certain embodiments calculates the comparison data based on the cyclic data. For example, in one embodiment, the verification element **56** calculates the comparison data in substantially the same manner that the data generation element **54** generates the data as described above (e.g., based on a current write address, using an LFSR, etc.). As such, the comparison data of certain embodiments is substantially a repeat of the data written. In one example embodiment, the data handler **30** is configured (e.g., is programmed by the control module **22**) to write an alternating series of "A's" and "5's" to the data ports of the corresponding plurality of data ports as described herein. For example, the data handler **30** may be configured to write one or more "A's" to the first address location of an N-word memory device **20** including the corresponding plurality of data ports. The data handler **30** may then write one or more "5's" to the second address location, one or more "A's" to the third address location and so on until the data handler **30** has written to all N memory locations. The verification element **56** of the example embodiment then calculates the comparison data based on the cyclic data written to the corresponding plurality of data ports. For example, the verification element **56** calculates a comparison word including one or more "A's", "5's," and "A's," respectively, to be compared to the data read from the first, second, and third address locations based on the cyclic data (e.g., based on the known cycle of the data). In some embodiments, the comparison data is calculated based on a current read address.

In certain embodiments, data associated with failures in the operation of the plurality of memory devices **18** are stored in the data module **28**. For example, data read from the corresponding plurality of data ports which do not correspond to (e.g., match) the comparison data calculated by the verification element **56** may be stored in the data module. Moreover, in some embodiments, memory addresses associated with the failures in the operation of the plurality of memory devices **18** are stored in the control module **22**. For example, the data handler **30** may communicate data failures (e.g., when data read from the corresponding plurality of data ports does not correspond to calculated comparison data) to the control module **22** which may then store the addresses corresponding to the data failure. In certain embodiments, the memory module **12** is configured to report failures (e.g., the failed data, the address corresponding to the memory location of the failed data, and/or expected data) via the I^2C interface **15** to the memory controller **14**. In addition, in certain embodiments the test controller **36** and/or the data handlers **30** may be updated through the I^2C interface **15** with new data patterns and/or with alternative memory access sequences to conduct AC tests (e.g., tests of the power, current, I/O speed, etc.).

In various embodiments, the components of the data module **28** (e.g., the switch **44**, the data handlers **30**, the data handler logic element **46**, the data generation element **54**, and/or verification element **56**) may include discrete logic, one or more application-specific integrated circuits (ASICs) one or more microprocessors, one or more field-programmable gate arrays (FPGAs), or one or more computer-programmable logic devices (CPLDs). Additionally, one or more of the various functional blocks (e.g., the switch **44**) of the data module **28** of FIG. **3** may not be included. In some embodiments, additional functional blocks may be included. Moreover, some of the functional blocks are described as separate functional blocks for illustration purposes and may comprise one physical component. For example, in one embodiment, each of the data handlers **30** and the corresponding switch **44**, data generation element **54**, and verification element **56** comprise one physical component (e.g., are included in one integrated circuit package). In another embodiment, the data module **28** comprises one physical component.

Referring again to FIG. **1**, a self-testing memory module **10** of certain embodiments comprises a printed circuit board (PCB) **12** and is configured to be operatively coupled to a memory controller **14** of a computer system **16**. The memory module **10** further includes a plurality of memory devices **18** on the printed circuit board **12** where each memory device **20** of the plurality of memory devices **18** comprising data, address, and control ports. The memory module **10** further comprises a control module **22** configured to generate address and control signals for testing the plurality of memory devices **18**. In certain embodiments, the memory module **10** further comprises a data module **28** comprising at least one data handler **30** and operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices **20**. The data handler **30** is configured to generate cyclic data for writing to the corresponding plurality of data ports. In certain embodiments, there may be one data handler **30**, for example. In other embodiments, there may be more than one

data handler **30**. The elements of the memory module **10** may be compatible with any of the embodiments described herein.

FIG. **4** is a flowchart of an example method **70** of self-testing a memory module **10** in accordance with certain embodiments described herein. While the method **70** is described herein by reference to the memory module **10**, other memory modules, electronic systems or subsystems, and/or circuits are also compatible with the embodiments described herein. The method **70** of certain embodiments comprises providing a self-testing memory module **10** at operational block **72**. The memory module **10** may comprise a printed circuit board (PCB) **12** configured to be operatively coupled to a memory controller **14** of a computer system **16**. The memory module **10** may further comprise a plurality of memory devices **18** on the printed circuit board **12**. Each memory device **20** of the plurality of memory devices **18** may comprise data, address, and control ports. The memory module **10** may comprise a control module **22** configured to generate address and control signals for testing the plurality of memory devices **18**. The memory module **10** may further comprise a data module **28** comprising a plurality of data handlers **30**. In certain embodiments, the data module **28** comprises at least one data handler **30**. Each data handler **30** of certain embodiments is operable independently from each of the other data handlers **30** of the plurality of data handlers **28** and is operatively coupled to a corresponding plurality of the data ports. At operational block **74**, the method **70** further comprises generating, by each of the data handlers **30**, data for writing to the corresponding plurality of data ports.

FIG. **5** is a flow diagram **100** illustrating self-testing operation of an example self-testing memory module **10** in accordance with certain embodiments described herein. At operational block **110** the memory module **10** enters an idle state after power up or, in some embodiments, when the memory board of the computer system **16** is connected to a mother board or to a test board. In one embodiment, at Operational block **110**, the contents of the register **40** are undefined and the controllers (e.g., the test controller **36** and the memory device controller **34**) are in unknown state. After power up, the memory module **10** is reset (e.g., a reset command is executed) at operational block **111**. At operational block **111**, the logic of the data handler **30** (e.g., data generation element **54** and/or verification element **56**) and the logic of the control mixer **32** are set to default values and/or states. For example, the memory module **10** may default to a non-test mode (e.g., normal operational mode) and the switches **44** of the data handlers **30** and the memory device controller **34** are not configured in a test mode. At operational block **112**, the test mode is configured. For example, the computer system **16** (e.g., through the memory controller **14**) may configure the test mode. The configuration may be through the I^2C interface **15**, for example, and may include configuring the test controller **36** for test mode. For example, configuring the test mode may include switching (e.g., by configuring the test controller **36**) the mode of the memory module **10** from a normal operation mode to the test mode.

At operational block **113**, the test mode is initiated. Initiating the test mode may comprise configuring a particular test case (e.g., particular test data or data patterns, particular write and/or read address sequences, etc.). The initiation may be achieved by activating the test controller **36** to initiate the test mode. For example, at operational block **113**, the data handlers **30** and the control mixer **32** are readied (e.g., via a tester) beginning the self-testing of the memory module **10**. For example, the tester may comprise an ATM tester, a server, a specialized tester. The tester may ready the data handlers **30** and the control mixer through the I^2C interface **15** or through the control signals **38**, for example. For example, the data handlers **30** are configured to input data generated by the data handlers **30** to the corresponding plurality of data ports of the plurality of memory devices **18** and the control mixer **32** is configured to input address and control signals from the test controller **36** to the plurality of memory devices **18**. At operational block **113** the test controller **36** updates each of the data handlers **30** (e.g., with new data patterns, write signal characteristics, etc.).

In one embodiment, the input signal on a pin (e.g., a parity-in pin) of the memory module **10** is asserted or toggled (e.g., by the memory controller **14**). For example, if the parity-in signal ("Par-in") is un-asserted (e.g., set to a "low" value), it is asserted (e.g., set to a "high" value) and held in the asserted state. Alternatively, if the parity-in signal is already in the asserted state, it may be toggled and then held in the asserted state. The memory module **10** executes the test mode (e.g., writes and reads test data patterns) at operational block **114**. The memory module **10** continues executing the test mode at operational block **114** until the self-test is complete or until the memory module **10** detects a certain number of memory failures such that a failure count exceeds a preset number.

If the failure count exceeds the preset number, the memory module **10** enters operational block **115** and "errors out" of the test mode. If the self-test is completed at operational block **114** and the failure count does not exceed the preset number, the memory module **10** enters operational block **115**. At operational block **115**, the memory module **10** sends out a test completion indication signal (e.g., through the test controller **36** to the memory controller **14**). At operational block **116**, the fail status of the test may be read out through, for example, the I^2C interface **15**. For example, one or more address values associated with failed memory locations, and/or the data read from those locations, and/or expected data may be read. In other cases, where the test fails, the memory module **10** may report the failure to the memory controller **14** through any available signal (e.g., bidirectional data or data strobe signal) between the memory module **10** and the memory controller **14**.

In some embodiments, the memory module **10** is generally interruptible. For example, the memory module **10** may exit the self-test and return to operational block **110** if the signal going into the parity-in pin is, for example, de-asserted or removed at operational block **155**. In other embodiments, the interrupt signal or condition may be different and may not be a parity-in signal. For example, in one embodiment, the memory module **10** may exit the self-test and return to operational block **110** if a timer having a pre-determined count expires. When the self-testing is interrupted, the memory module **10** will return to operational block **110** and the bit failure information in the control mixer **32** and the data handler **30** becomes invalid.

FIG. **6** is a flow diagram **200** illustrating the operation of an example self-testing memory module **10** in accordance with certain embodiments described herein. For example, one or more of the operational blocks of the flow diagram **200** of FIG. **6** may correspond to one or more of the operational blocks of FIG. **5**. At operational block **210**, the memory module **10** may be generally inactive. For example, the testing logic (e.g., the control module **22** and/or the data module **28**) may be generally inactive and the memory module **10** may default to a functional (e.g., non-test) mode. In certain embodiments, the memory module **10** executes one or more tasks which configure the self-test mode of the self-testing memory module using an I^2C interface **15**. For example, the memory module **200** may configure the test controller **36** through the I^2C interface **15** (e.g., as described with respect to

FIG. **5**). In other embodiments, the test mode is configured using the control signals **38**. For example, the test mode may be updated at operational block **211** in certain embodiments. In certain embodiments, updating the test mode comprises programming the test controller **36** with test configuration information such as test data patterns, test data sequences, test failure conditions, etc. After updating the test mode at operational block **211**, the test controller **36** may then be activated through the I^2C interface **15** (e.g., as described with respect to FIG. **5**) in certain embodiments. Alternatively, as shown, the test controller **36** may be activated through the I^2C interface **15** without first updating the test mode.

At operational block **212**, the memory module test mode is ready for execution. In one embodiment, the input signal on a pin (e.g., a parity-in pin) of the memory module **10** is asserted or toggled (e.g., by the memory controller **14**). For example, if the parity-in signal is low, it is asserted (e.g., set to a "high" value) and held in the asserted state. Alternatively, if the parity-in signal is already in the asserted state, it may be toggled and then held in the asserted state. At operational block **113**, a set of status registers (e.g., one or more registers of the control module **22** or the data module **28**) are cleared.

The memory module **200** generates address and data signals for testing the plurality of memory devices **18** at operational block **214**. For example, the data generation elements **54** of the data handlers **30** may generate the data for writing to the plurality of memory devices **18** as described herein. In addition, the control module **22** may generate address and control signals for testing the plurality of memory devices **18** as described herein. At operational block **215**, the memory module **10** performs a burst write function. For example, in certain embodiments, the memory module **10** writes multiple memory locations in the memory module **10** that are to be tested at operational block **215** in a relatively short amount of time (or in a burst). In certain embodiments, the memory module **10** can perform multiple write bursts at operational block **215**. Next, at operational block **216**, the memory module **10** performs a read & compare function. For example, in certain embodiments, the memory module **10** reads back certain memory locations at operational block **216** that have been written at operational block **215** and compares the values with certain expected data. For example, a verification element **56** of each of the data handlers **30** may calculate the expected data and/or compare the values as described herein. In some embodiments, the memory module **200** reads from multiple memory locations in the memory module **10** in a relatively short amount of time (or in a burst). In certain embodiments, the memory module **10** can execute multiple read bursts at operational block **216** as it compares the read data with expected data. In some embodiments, the memory module **200** (e.g., the data handlers **30** of the memory module **200**) stores any failed read data and the associated expected data at operational block **217**. In some embodiments, the memory module sends one or more failure indications to the memory controller **14** over one or more available signals between memory module **10** and the memory controller **14**. For example, the I^2C interface **15** or one or more bi-directional data pins (e.g., one or more of the data pins **48**) may be used.

The memory module **200** stores the addresses of the memory locations associated with the failed read data using the test controller **32** at operational block **217**. In other embodiments the failed read data, expected data, and the addresses of the memory locations associated with the failed read data may be stored using the test controller **32**. For example, in one embodiment, the test controller **32** stores the failed read data, expected data, and the addresses of the memory locations associated with the failed read data at operational block **217**. In some embodiments, the test controller **32** is generally inaccessible through the I^2C interface **15** during self-test operation. For example, the test controller is inaccessible through the I^2C interface **15** until a test failure occurs at operational block **217** or until the test completes at operational block **219**. In some embodiments, when the test is complete, the test results can be read out of the memory module **10** (e.g., through the I^2C interface **15**). In some cases, the test can be interrupted using the I^2C interface **15** or using the parity-in signal (e.g., by de-asserting the parity-in signal).

Although certain preferred embodiments and examples are discussed above, it is understood that the inventive subject matter extends beyond the specifically disclosed embodiments to other alternative embodiments and/or uses of the invention and obvious modifications and equivalents thereof. It is intended that the scope of the inventions disclosed herein should not be limited by the particular disclosed embodiments. Thus, for example, in any method or process disclosed herein, the acts or operations making up the method/process may be performed in any suitable sequence and are not necessarily limited to any particular disclosed sequence. Various aspects and advantages of the embodiments have been described where appropriate. It is to be understood that not necessarily all such aspects or advantages may be achieved in accordance with any particular embodiment. Thus, for example, it should be recognized that the various embodiments may be carried out in a manner that achieves or optimizes one advantage or group of advantages as taught herein without necessarily achieving other aspects or advantages as may be taught or suggested herein.

What is claimed is:

1. A self-testing memory module, comprising:

a printed circuit board configured to be operatively coupled to a memory controller of a computer system;

a plurality of memory devices on the printed circuit board, each memory device of the plurality of memory devices comprising data, address, and control ports; and

a circuit comprising:

a control module configured to generate address and control signals for testing the memory devices; and

a data module comprising a plurality of data handlers, each data handler operable independently from each of the other data handlers of the plurality of data handlers and operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and configured to generate data for writing to the corresponding plurality of data ports, wherein the circuit is configured to test the memory devices using the address and control signals generated by the control module and the data generated by the plurality of data handlers.

2. The self-testing memory module of claim **1**, wherein the plurality of data handlers comprise at least two physically separate components mounted on the printed circuit board.

3. The self-testing memory module of claim **2**, wherein the plurality of data handlers comprise at least two physically separate integrated circuit packages.

4. The self-testing memory module of claim **3**, wherein the physically separate integrated circuit packages are mounted on different portions of the printed circuit board.

5. The self-testing memory module of claim **2**, wherein each of the plurality of data handlers is positioned on the printed circuit board proximate to the corresponding plurality of data ports.

6. The self-testing memory module of claim **5**, wherein each of the plurality of data handlers is positioned closer to

the corresponding plurality of data ports than to the other data ports of the plurality of memory devices.

7. The self-testing memory module of claim 6, wherein each of the data handlers is further configured to read from the corresponding plurality of data ports and further comprises a verification element for checking for failures in the operation of the memory devices by verifying that data read from the corresponding plurality of data ports corresponds to the data generated by the data handler for writing to the corresponding plurality of data ports.

8. The self-testing memory module of claim 7, wherein the verification element is configured to verify that the data read from the corresponding plurality of data ports corresponds to the data generated by the data handler for writing to the corresponding plurality of data ports without storing a copy of the data generated by the data handler for writing to the corresponding plurality of data ports.

9. The self-testing memory module of claim 7, wherein the verification element calculates comparison data and compares the comparison data to the data read from the corresponding plurality of data ports.

10. The self-testing memory module of claim 9, wherein the data for writing to the corresponding plurality of data ports comprises cyclic data.

11. The self-testing memory module of claim 10, wherein the verification element calculates the comparison data based on the cyclic data.

12. The self-testing memory module of claim 10, wherein the verification element calculates the comparison data based on a current read address.

13. The self-testing memory module of claim 7, wherein data associated with failures in the operation of the memory devices is stored in the data module and wherein memory addresses associated with failures in the operation of the memory devices are stored in the control module.

14. The self-testing memory module of claim 1, each of the plurality of data handlers further configured to write data to the memory devices by selectively inputting to the corresponding plurality of data ports either data signals from the system memory controller or data signals from the data handler.

15. The self-testing memory module of claim 14, wherein each of the plurality of data handlers comprises a switch which performs the selective inputting to the corresponding plurality of data ports.

16. The self-testing memory module of claim 15, wherein the switch comprises a data multiplexer-demultiplexer.

17. The self-testing memory module of claim 1, the control module further configured to selectively input to the address and control ports of the memory devices either the address and control signals from the memory controller or the address and control signals from the control module.

18. The self-testing memory module of claim 17, wherein the control module further comprises a DRAM controller which performs the selective inputting of the address and control ports.

19. The self-testing memory module of claim 1, wherein the memory module is configured to be operated in a test mode in which the control module selectively inputs to the address and control ports of the memory devices the address and control signals from the control module and wherein each of the data handlers write the data generated by the data handler to the corresponding plurality of data ports by selectively inputting to the data ports of the memory devices data signals from the data handler.

20. A self-testing memory module, comprising:

a printed circuit board configured to be operatively coupled to a memory controller of a computer system;

a plurality of memory devices on the printed circuit board, each memory device of the plurality of memory devices comprising data, address, and control ports; and

a circuit comprising:

a control module configured to generate address and control signals for testing the memory devices; and

a data module comprising at least one data handler operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and configured to generate cyclic data for writing to the corresponding plurality of data ports wherein the circuit is configured to test the memo devices using the address and control signals generated by the control module and the cyclic data generated by the at least one data handler.

21. The self-testing memory module of claim 20, wherein the at least one data handler is further configured to read data from the corresponding plurality of data ports and further comprises a verification element for checking for failures in the operation of the memory devices by verifying that data read from the corresponding plurality of data ports corresponds to the data generated by the data handler for writing to the corresponding plurality of data ports.

22. The self-testing memory module of claim 21, wherein the verification element is configured to verify that the data read from the corresponding plurality of data ports corresponds the data generated by the data handler for writing to the corresponding plurality of data ports without storing a copy of the data written to the memory devices.

23. The self-testing memory module of claim 22, wherein the verification element calculates comparison data and compares the comparison data to the data read from the corresponding plurality of data ports.

24. The self-testing memory module of claim 23, wherein the verification element calculates the comparison data based on the cyclic data.

25. The self-testing memory module of claim 23, wherein the verification element calculates the comparison data based on a current read address.

26. The self-testing memory module of claim 21, wherein data associated with failures in the operation of the memory devices is stored in the data module and wherein memory addresses associated with failures in the operation of the memory devices are stored in the control module.

27. The self-testing memory module of claim 20, wherein there are at least two data handlers and each of the data handlers is operable independently from each of the other data handlers.

28. The self-testing memory module of claim 20, wherein the at least one data handler is positioned on the printed circuit board proximate to the corresponding plurality of data ports.

29. A method of self-testing a memory module, comprising:

providing a self-testing memory module, the self-testing memory module comprising:

a printed circuit board configured to be operatively coupled to a memory controller of a computer system;

a plurality of memory devices on the printed circuit board, each memory device of the plurality of memory devices comprising data, address, and control ports; and

a circuit comprising:

a control module configured to generate address and control signals for testing the memory devices; and

a data module comprising a plurality of data handlers, each data handler operable independently from each of the other data handlers of the plurality of data handlers and operatively coupled to a corresponding plurality of the data ports; and

generating, by the control module, address and control signals for testing the memory devices;

generating, by each of the data handlers, data for writing to the corresponding plurality of data ports; and

using the address and control signals and the generated data to test the memory devices.

30. The method of claim **29**, further comprising:

reading, by each of the plurality of data handlers, data from the corresponding plurality of data ports; and

checking for failures in the operation of the memory devices by each of the plurality of data handlers by verifying that data read from the corresponding plurality of data ports corresponds to the generated data for writing to the corresponding plurality of data ports.

31. The method of claim **30**, further comprising verifying that the data read from the corresponding plurality of data ports corresponds to the generated data for writing to the corresponding plurality of data ports without storing a copy of the data written to the corresponding plurality of data ports.

32. The method of claim **31**, further comprising:

calculating comparison data; and

comparing the comparison data to the data read from the corresponding plurality of data ports.

33. The method of claim **32**, wherein the data signals written to the corresponding plurality of data ports comprises cyclic data and calculating comprises calculating the data based on the cyclic data.

34. The method of claim **32**, wherein the data signals written to the corresponding plurality of data ports comprises cyclic data calculating comprises calculating the data based on a current read address.

35. The method of claim **30**, further comprising operating the memory module in a test mode in which the control module selectively inputs to the address and control ports of the memory devices the address and control signals from the control module and wherein each of the data handlers write the generated data to the corresponding plurality of data ports by selectively inputting to the data ports of the memory devices data signals representative of the generated data from the data handler.

* * * * *

US008301833B1

(12) **United States Patent**
Chen et al.

(10) **Patent No.: US 8,301,833 B1**
(45) **Date of Patent: Oct. 30, 2012**

(54) **NON-VOLATILE MEMORY MODULE**

(75) Inventors: **Chi-She Chen**, Walnut, CA (US); **Jeffrey C. Solomon**, Irvine, CA (US); **Scott Milton**, Irvine, CA (US); **Jayesh Bhakta**, Cerritos, CA (US)

(73) Assignee: **Netlist, Inc.**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 638 days.

(21) Appl. No.: **12/240,916**

(22) Filed: **Sep. 29, 2008**

Related U.S. Application Data

(63) Continuation of application No. 12/131,873, filed on Jun. 2, 2008, now abandoned.

(60) Provisional application No. 60/941,586, filed on Jun. 1, 2007.

(51) **Int. Cl.**
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **711/104**; 711/160; 711/161; 711/162; 710/10

(58) **Field of Classification Search** 711/160, 711/161, 162, 104; 710/10
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,420,821	A	12/1983	Hoffman	
4,449,205	A	5/1984	Hoffman	
5,519,663	A	5/1996	Harper, Jr. et al.	
6,158,015	A	12/2000	Klein	
6,336,174	B1	1/2002	Li et al.	
6,336,176	B1	1/2002	Leyda et al.	
6,487,623	B1	11/2002	Emerson et al.	
6,658,507	B1	12/2003	Chan	
6,799,244	B2	9/2004	Tanaka et al.	
7,409,590	B2	8/2008	Moshayedi et al.	
2002/0083368	A1	6/2002	Abe et al.	
2004/0190210	A1	9/2004	Leete	
2007/0192627	A1 *	8/2007	Oshikiri	713/191
2008/0195806	A1 *	8/2008	Cope	711/111

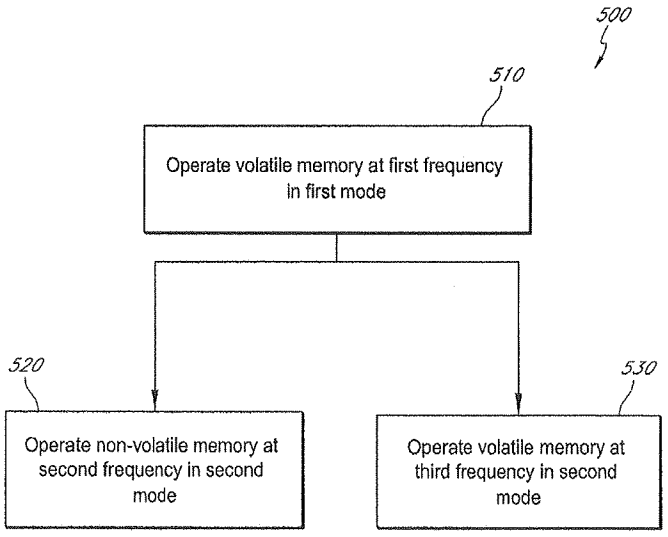
* cited by examiner

Primary Examiner — Midys Rojas
(74) *Attorney, Agent, or Firm* — Nixon Peabody LLP; Khaled Shami

(57) **ABSTRACT**

Certain embodiments described herein include a memory system which can communicate with a host system such as a disk controller of a computer system. The memory system can include volatile and non-volatile memory and a controller which are configured such that the controller backs up the volatile memory using the non-volatile memory in the event of a trigger condition. In order to power the system in the event of a power failure or reduction, the memory system can include a secondary power source which is not a battery and may include, for example, a capacitor or capacitor array. The memory system can be configured such that the operation of the volatile memory is not adversely affected by the non-volatile memory or the controller when the volatile memory is interacting with the host system.

30 Claims, 12 Drawing Sheets



Exh B

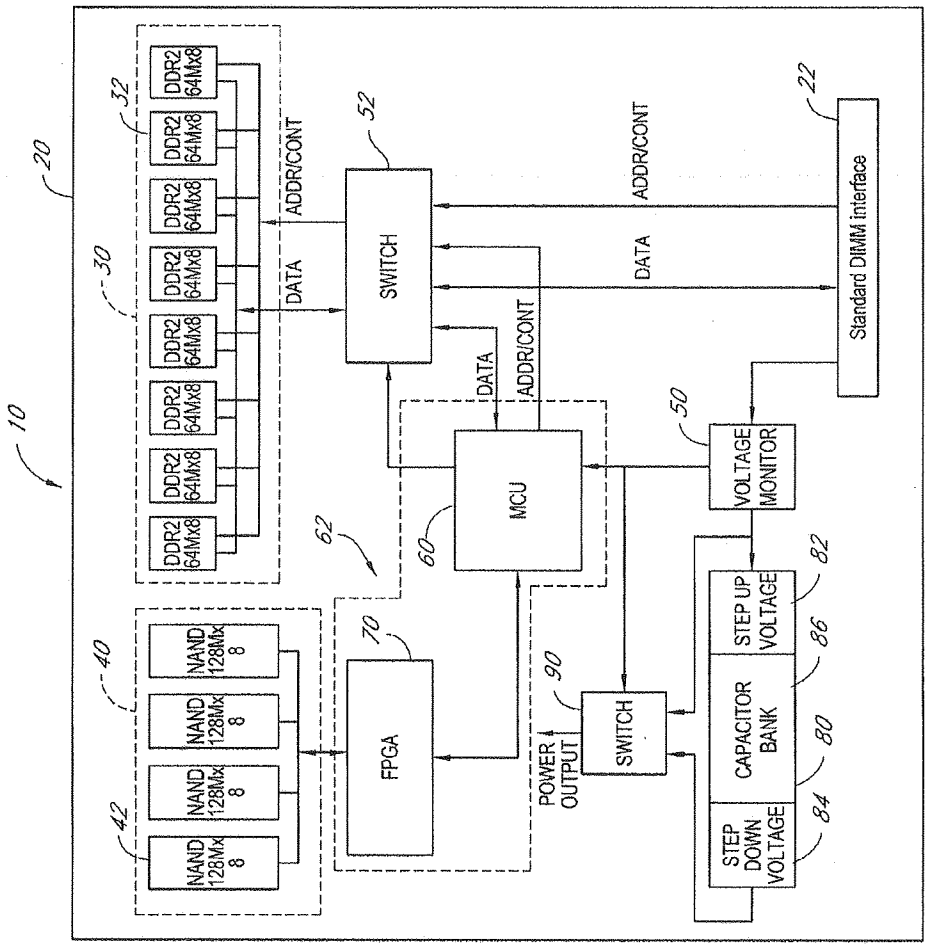

10
20
30
32
DDR2 64Mx8
ADDR/CONT
DATA
SWITCH
52
ADDR/CONT
DATA
22
Standard DIMM interface
DATA
ADDR/CONT
MCU
60
62
50
VOLTAGE MONITOR
40
42
NAND 128Mx 8
FPGA
70
POWER OUTPUT
90
SWITCH
STEP UP VOLTAGE
82
86
CAPACITOR BANK
80
STEP DOWN VOLTAGE
84

FIG. 1

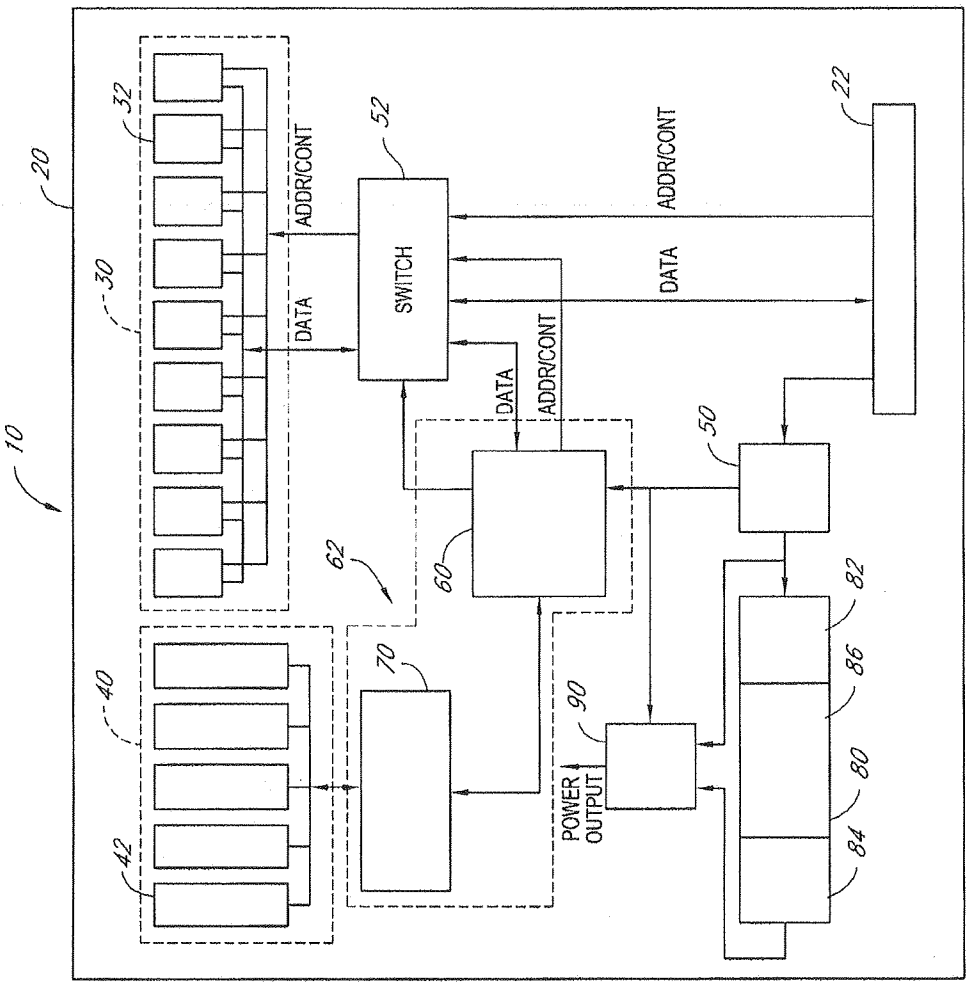

10
20
30
32
40
42
50
52
60
62
70
80
82
84
86
90
22
ADDR/CONT
DATA
SWITCH
ADDR/CONT
DATA
DATA
ADDR/CONT
POWER OUTPUT

FIG. 2

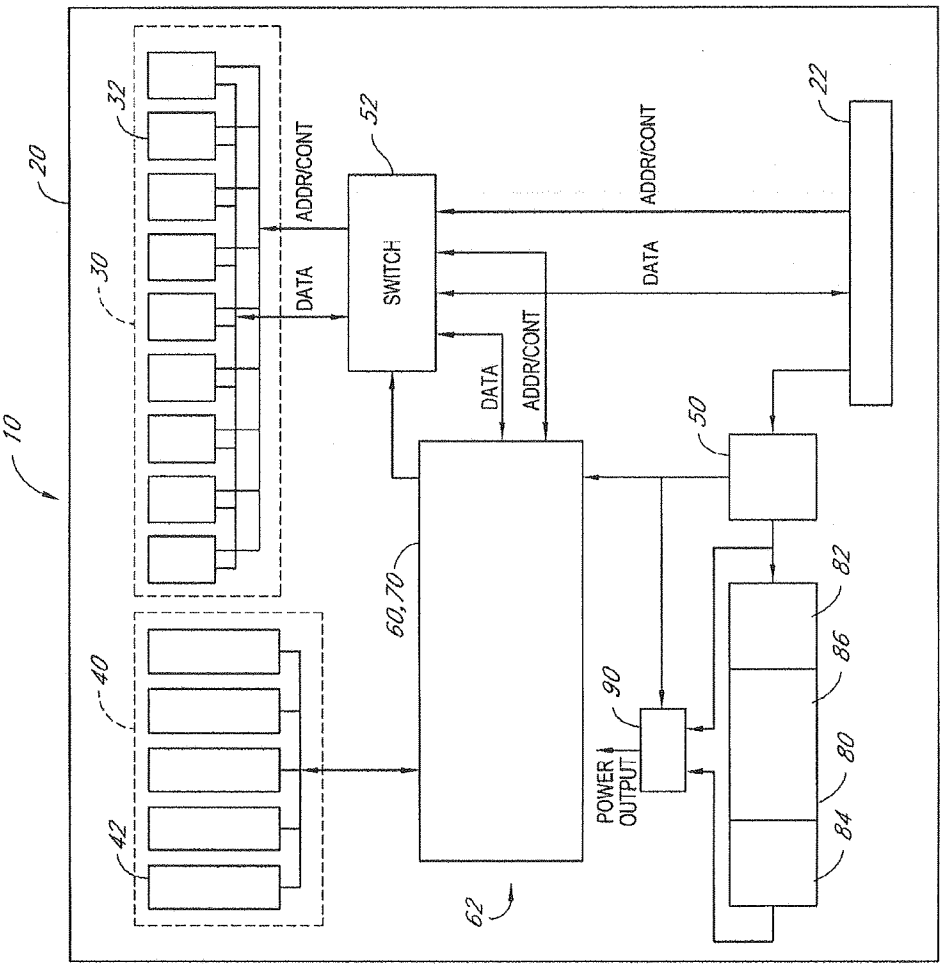

10
20
30
32
40
42
52
SWITCH
ADDR/CONT
DATA
ADDR/CONT
DATA
DATA
ADDR/CONT
22
50
60,70
62
90
POWER OUTPUT
82
86
80
84

FIG. 3

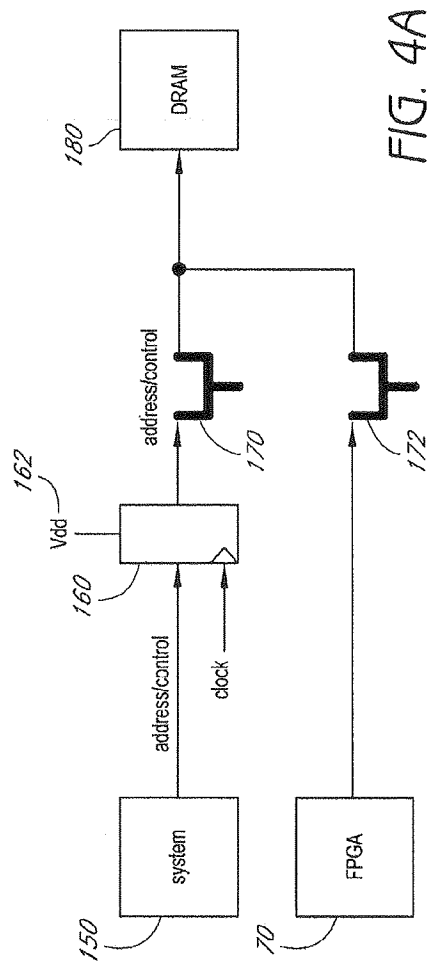

DRAM
180
address/control
170
172
162
Vdd
160
address/control
clock
system
150
FPGA
70

FIG. 4A

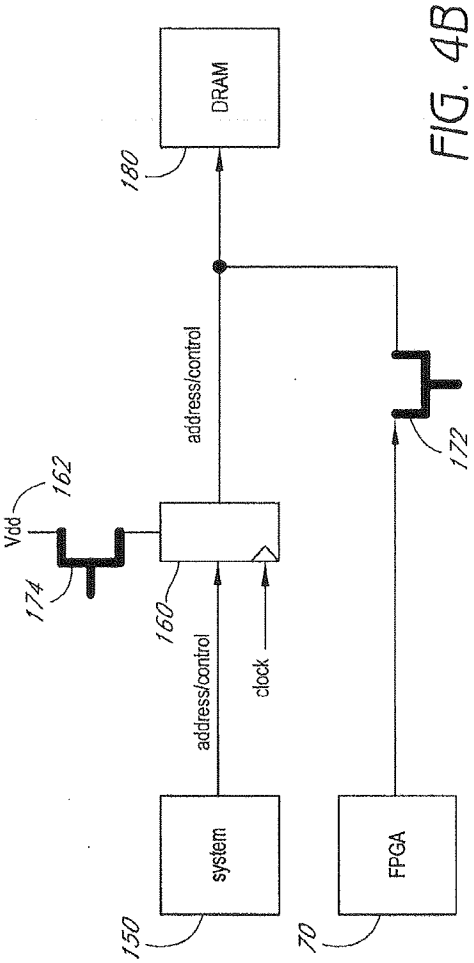

DRAM
180
address/control
172
162
Vdd
174
160
address/control
clock
system
150
FPGA
70

FIG. 4B

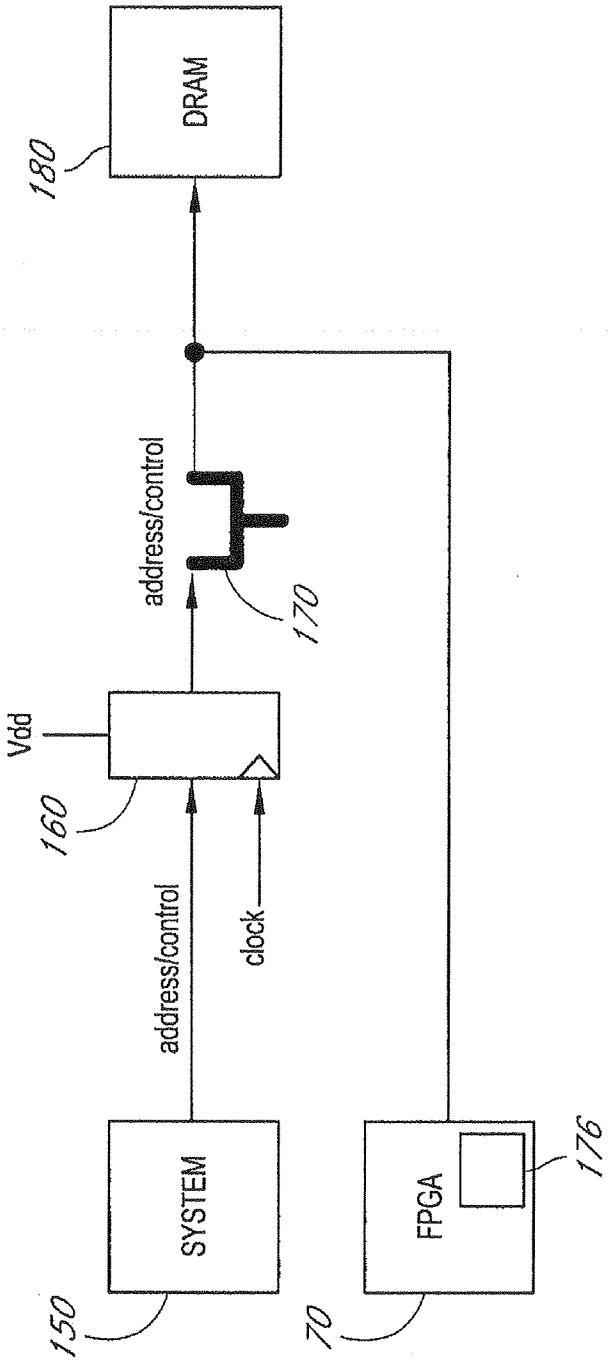

DRAM
180
address/control
170
Vdd
160
address/control
clock
SYSTEM
150
FPGA
176
70

FIG. 4C

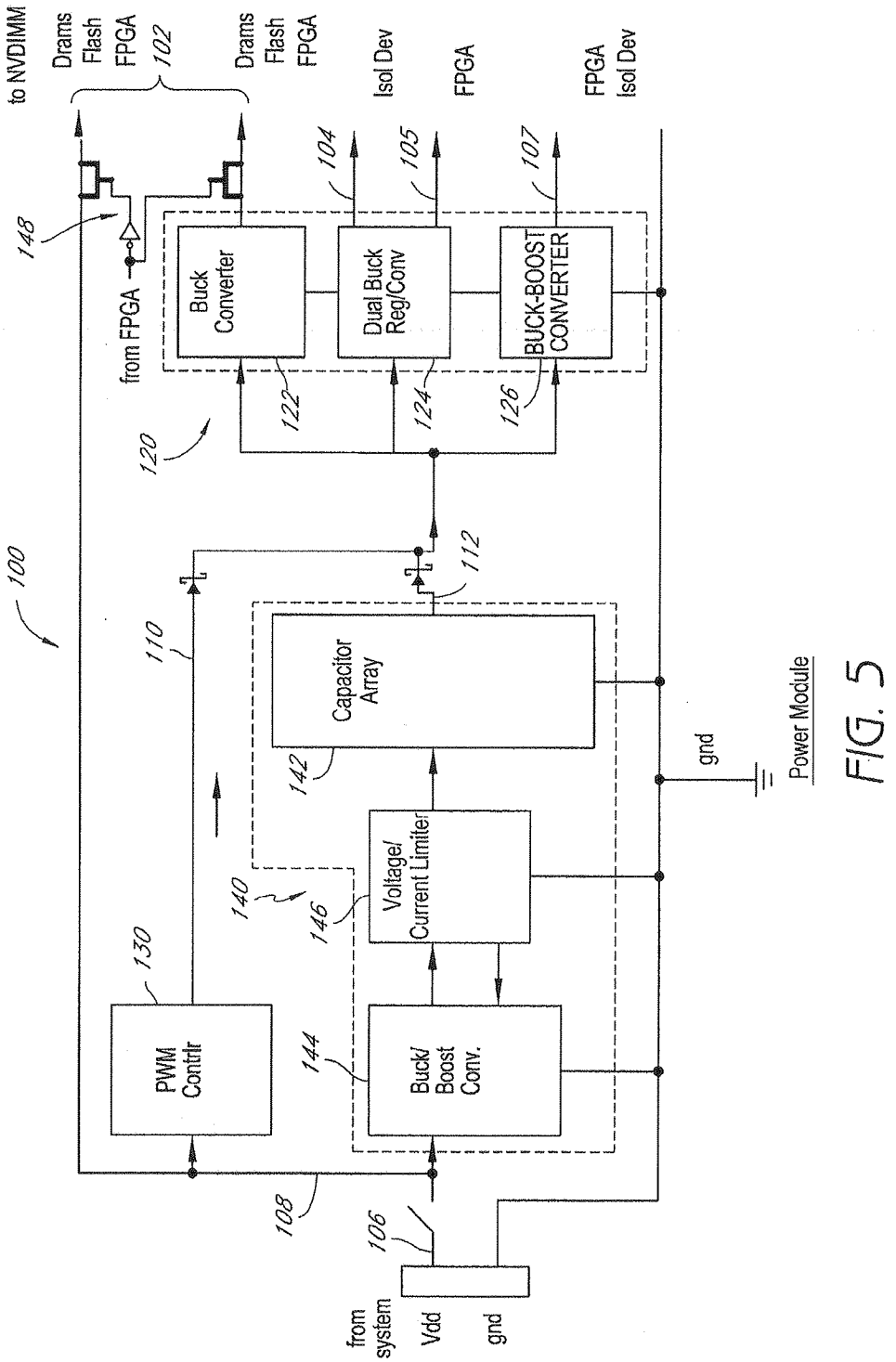

to NVDIMM
Drams Flash FPGA
102
Drams Flash FPGA
Isol Dev
FPGA
FPGA Isol Dev
104
105
107
148
from FPGA
Buck Converter
Dual Buck Reg/Conv
BUCK-BOOST CONVERTER
122
124
126
120
100
112
110
Capacitor Array
142
gnd
Power Module
Voltage/ Current Limiter
140
146
130
PWM Contrlr
144
Buck/ Boost Conv.
108
106
from system
Vdd
gnd

FIG. 5

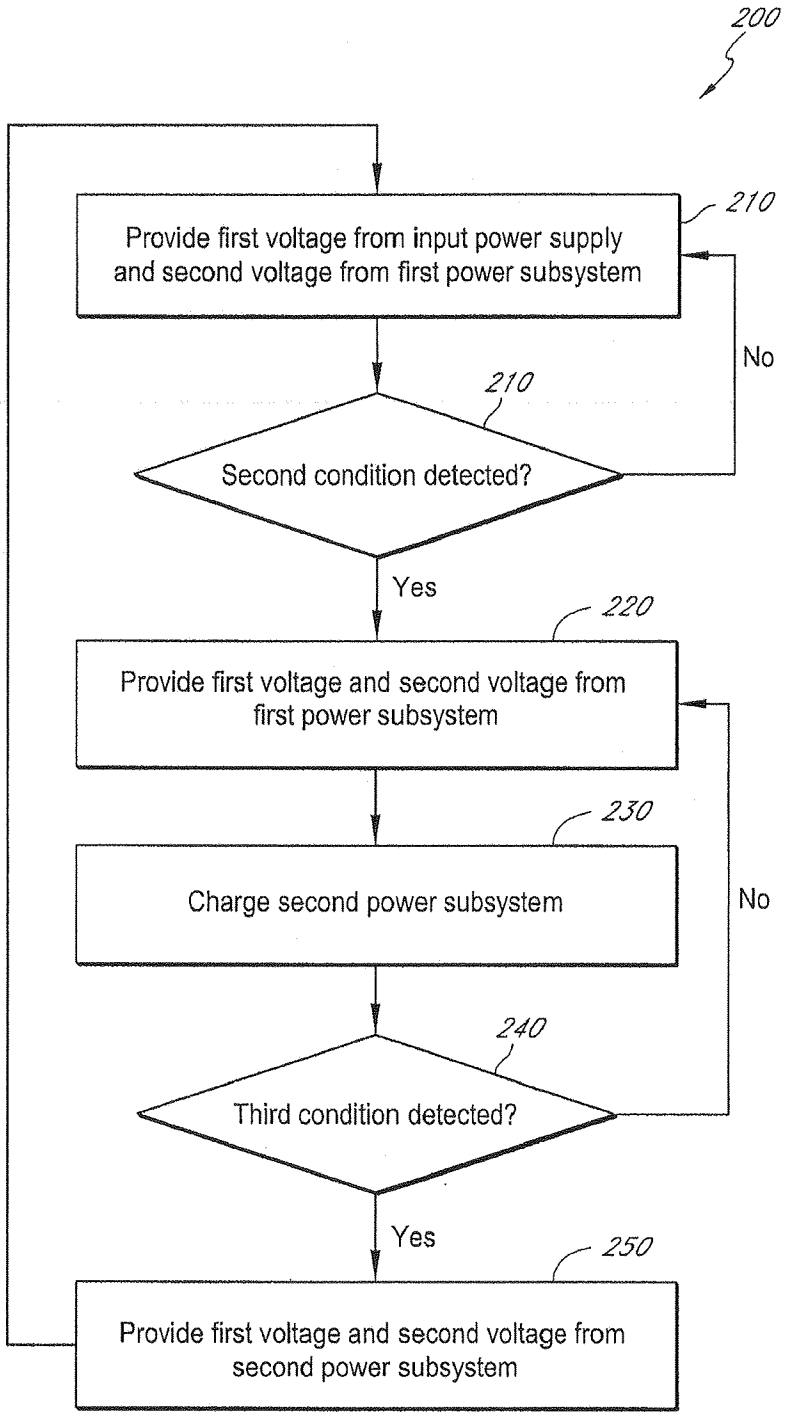

200
Provide first voltage from input power supply and second voltage from first power subsystem
210
No
210
Second condition detected?
Yes
220
Provide first voltage and second voltage from first power subsystem
230
Charge second power subsystem
No
240
Third condition detected?
Yes
250
Provide first voltage and second voltage from second power subsystem

FIG. 6

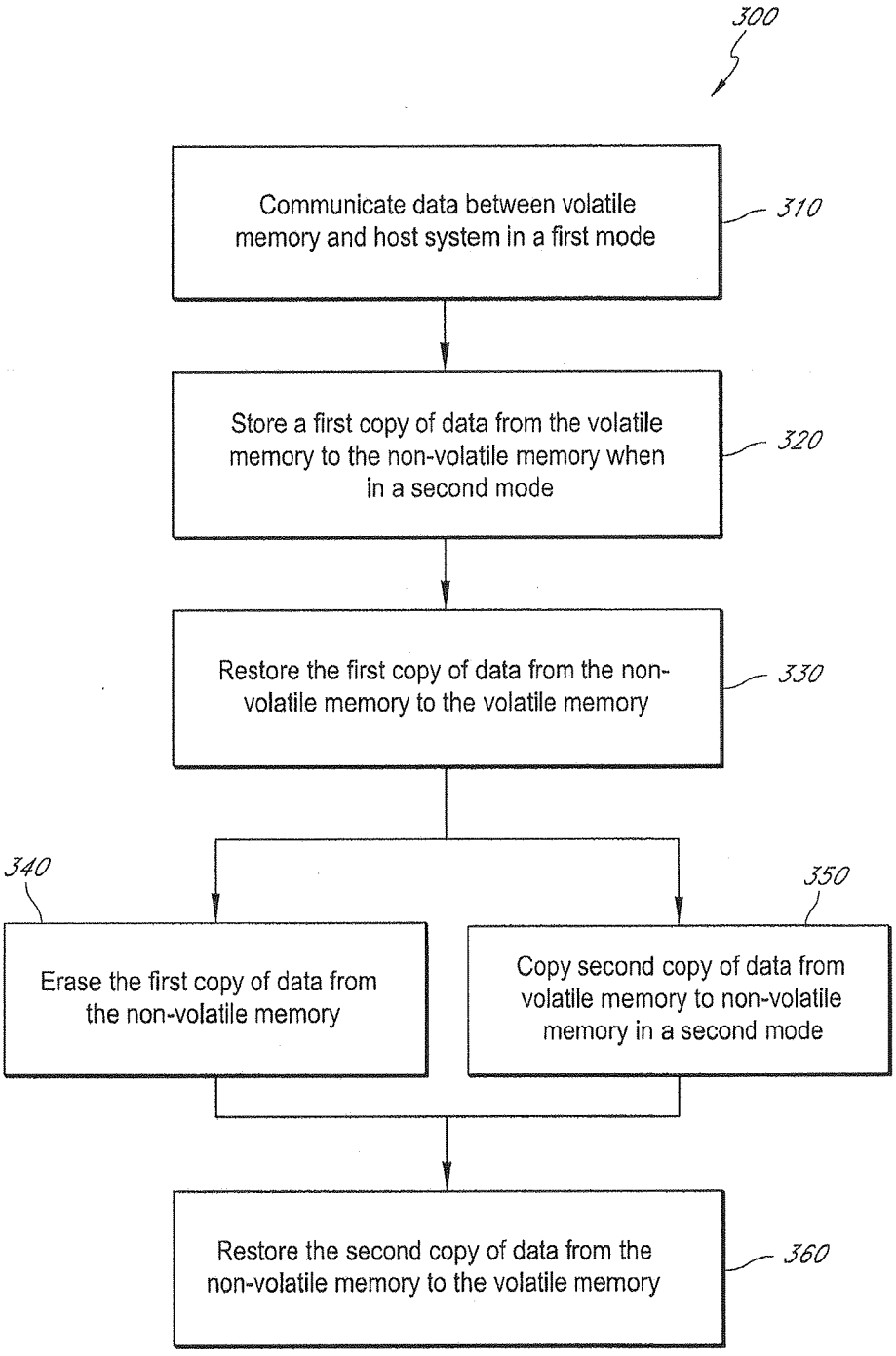

300
Communicate data between volatile memory and host system in a first mode
310
Store a first copy of data from the volatile memory to the non-volatile memory when in a second mode
320
Restore the first copy of data from the non-volatile memory to the volatile memory
330
340
Erase the first copy of data from the non-volatile memory
350
Copy second copy of data from volatile memory to non-volatile memory in a second mode
Restore the second copy of data from the non-volatile memory to the volatile memory
360

FIG. 7

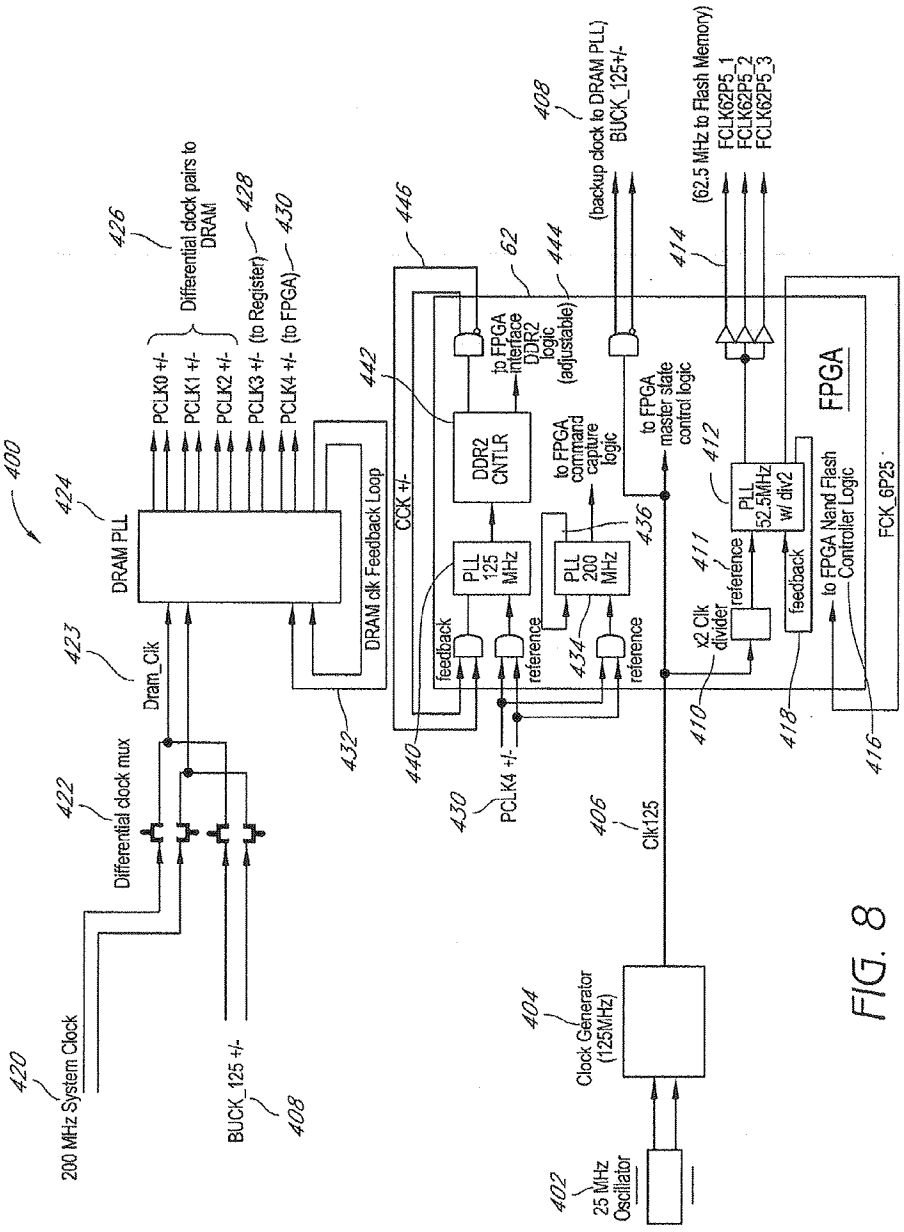

400
424
DRAM PLL
PCLK0 +/-
PCLK1 +/-
PCLK2 +/-
PCLK3 +/- (to Register)
PCLK4 +/- (to FPGA)
426
Differential clock pairs to DRAM
428
430
446
62
444
408
(backup clock to DRAM PLL)
BUCK_125+/-
414
(62.5 MHz to Flash Memory)
FCLK62P5_1
FCLK62P5_2
FCLK62P5_3
442
to FPGA interface DDR2 logic (adjustable)
DDR2 CNTLR
to FPGA command capture logic
to FPGA master state control logic
412
PLL 52.5MHz w/ div2
FPGA
423
Dram_Clk
DRAM clk Feedback Loop
CCK +/-
PLL 125 MHz
PLL 200 MHz
436
411
reference
feedback
reference
reference
434
x2 Clk divider
feedback
to FPGA Nand Flash Controller Logic
FCK_6P25
422
Differential clock mux
432
440
410
418
416
430
PCLK4 +/-
406
Clk125
420
200 MHz System Clock
BUCK_125 +/-
408
404
Clock Generator (125MHz)
402
25 MHz Oscillator

FIG. 8

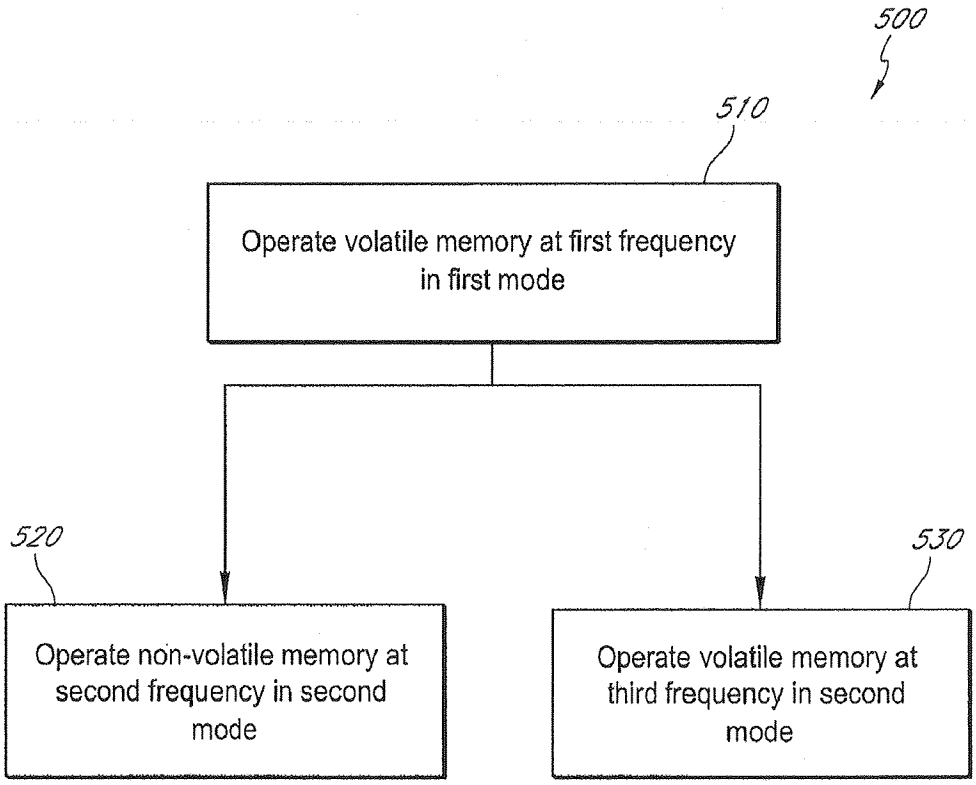

500
510
Operate volatile memory at first frequency in first mode
520
Operate non-volatile memory at second frequency in second mode
530
Operate volatile memory at third frequency in second mode

FIG. 9

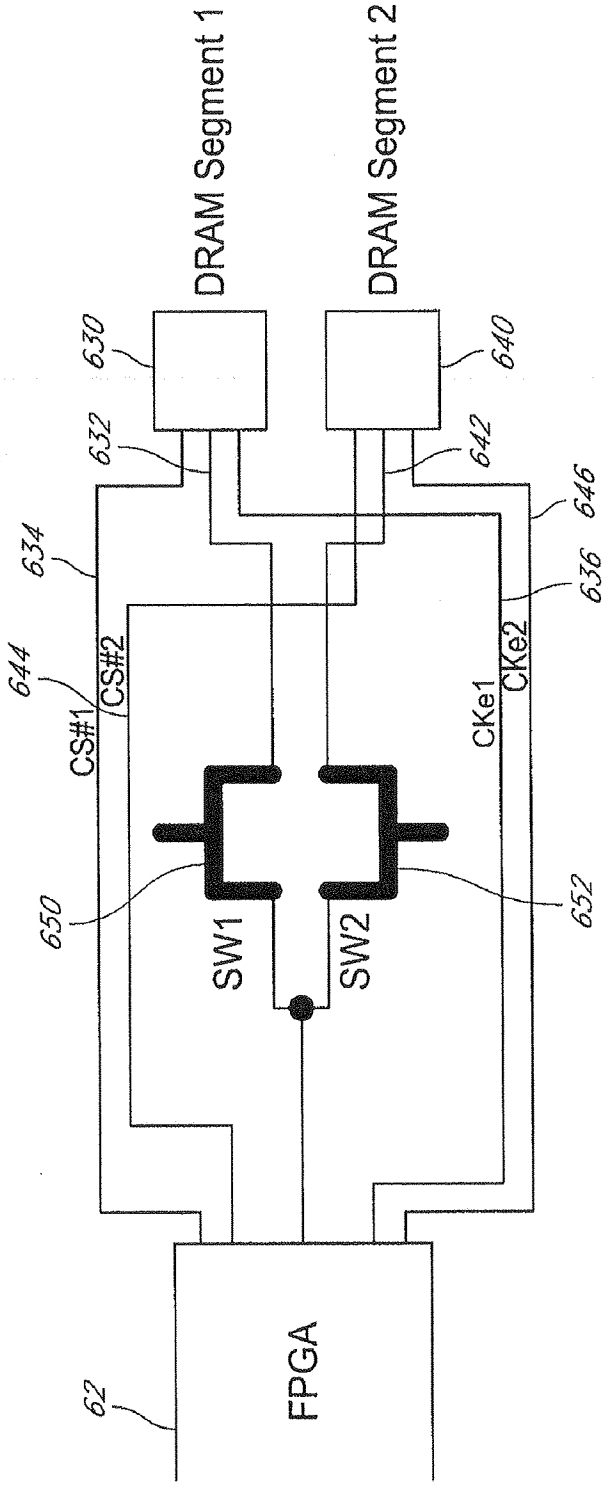

DRAM Segment 1
DRAM Segment 2
630
640
632
642
646
634
636
644
CS#2
CS#1
CKe2
CKe1
650
652
SW1
SW2
62
FPGA

FIG. 10

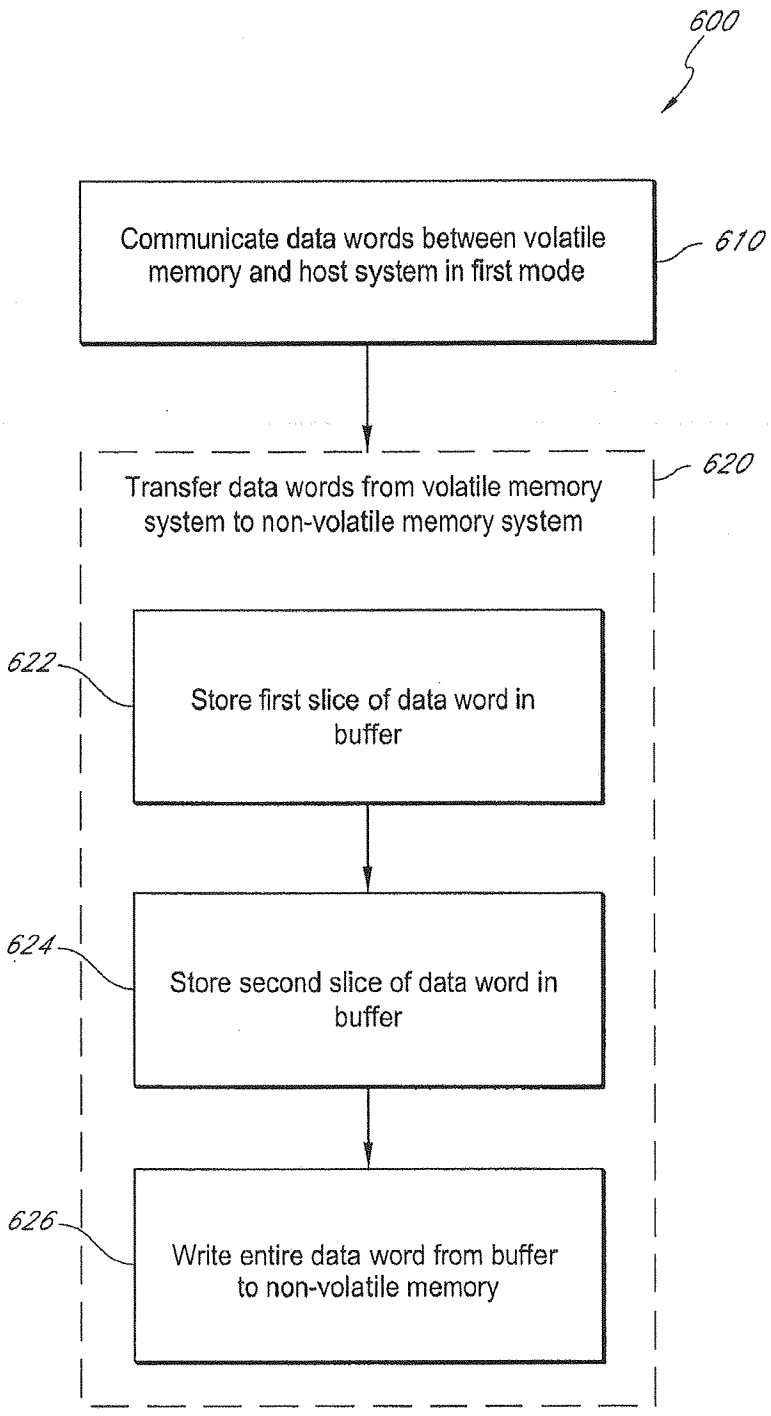

600
Communicate data words between volatile memory and host system in first mode
610
Transfer data words from volatile memory system to non-volatile memory system
620
622
Store first slice of data word in buffer
624
Store second slice of data word in buffer
626
Write entire data word from buffer to non-volatile memory

FIG. 11

NON-VOLATILE MEMORY MODULE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 12/131,873, filed Jun. 2, 2008, which claims the benefit of U.S. Provisional Application No. 60/941,586, filed Jun. 1, 2007. Each of these applications is incorporated in its entirety by reference herein.

BACKGROUND

Certain types of memory modules comprise a plurality of dynamic random-access memory (DRAM) devices mounted on a printed circuit board (PCB). These memory modules are typically mounted in a memory slot or socket of a computer system (e.g., a server system or a personal computer) and are accessed by the computer system to provide volatile memory to the computer system.

Volatile memory generally maintains stored information only when it is powered. Batteries have been used to provide power to volatile memory during power failures or interruptions. However, batteries may require maintenance, may need to be replaced, are not environmentally friendly, and the status of batteries can be difficult to monitor.

Non-volatile memory can generally maintain stored information while power is not applied to the non-volatile memory. In certain circumstances, it can therefore be useful to backup volatile memory using non-volatile memory.

SUMMARY

In certain embodiments, a memory system coupled to a computer system is provided which includes a volatile memory subsystem, a non-volatile memory subsystem, and a controller operatively coupled to the non-volatile memory subsystem. The memory system can also include at least one circuit configured to selectively operatively decouple the controller from the volatile memory subsystem.

In some embodiments, a power module for providing a plurality of voltages to a memory system is described. The power module includes non-volatile and volatile memory, and the plurality of voltages include at least a first voltage and a second voltage. The power module of certain embodiments includes an input providing a third voltage to the power module and a voltage conversion element configured to provide the second voltage to the memory system. The power module also includes a first power element configured to selectively provide a fourth voltage to the conversion element. The power module further includes a second power element configured to selectively provide a fifth voltage to the conversion element. The power module can be configured to selectively provide the first voltage to the memory system either from the conversion element or from the input.

The power module can be configured to be operated in at least three states in certain embodiments. In a first state, the first voltage is provided to the memory system from the input and the fourth voltage is provided to the conversion element from the first power element. In a second, state the fourth voltage is provided to the conversion element from the first power element and the first voltage is provided to the memory system from the conversion element. In a third state, the fifth voltage is provided to the conversion element from the second power element and the first voltage is provided to the memory system from the conversion element.

A method of providing a first voltage and a second voltage to a memory system including volatile and non-volatile memory subsystems is provided in certain embodiments. The method includes, during a first condition, providing the first voltage to the memory system from an input power supply and providing the second voltage to the memory system from a first power subsystem. The method further includes detecting a second condition and, during the second condition, providing the first voltage and the second voltage to the memory system from the first power subsystem. The method also includes charging a second power subsystem and detecting a third condition. During the third condition, the method includes providing the first voltage and the second voltage to the memory system from the second power subsystem.

In certain embodiments, a method is provided for controlling a memory system operatively coupled to a host system and which includes a volatile memory subsystem and a non-volatile memory subsystem. The method can include operating the volatile memory subsystem at a first frequency when the memory system is in a first mode of operation in which data is communicated between the volatile memory subsystem and the host system. In certain embodiments, the method further includes operating the non-volatile memory subsystem at a second frequency when the memory system is in a second mode of operation in which data is communicated between the volatile memory subsystem and the non-volatile memory subsystem. The method can also include operating the volatile memory subsystem at a third frequency when the memory system is in the second mode of operation, the third frequency less than the first frequency.

In certain embodiments, a method is provided for controlling a memory system operatively coupled to a host system. The memory system includes a volatile memory subsystem and a non-volatile memory subsystem. In certain embodiments, the method includes communicating data words between the volatile memory subsystem and the host system when the memory system is in a first mode of operation. The method can further include transferring data words from the volatile memory subsystem to the non-volatile memory subsystem when the memory system is in a second mode of operation. Transferring each data word can include storing a first portion of the data word in a buffer, storing a second portion of the data word in the buffer, and writing the entire data word from the buffer to the non-volatile memory subsystem.

A memory system operatively coupled to a host system is provided in certain embodiments. The memory system can include a volatile memory subsystem and a non-volatile memory subsystem comprising at least 100 percent more storage capacity than does the volatile memory subsystem. The memory system includes a controller operatively coupled to the volatile memory subsystem and operatively coupled to the non-volatile memory subsystem, the controller configured to allow data to be communicated between the volatile memory subsystem and the host system when the memory system is operating in a first state and to allow data to be communicated between the volatile memory subsystem and the non-volatile memory subsystem when the memory system is operating in a second state.

A method of controlling a memory system operatively coupled to a host system is provided in certain embodiments. The memory system includes a volatile memory subsystem and a non-volatile memory subsystem. The method can include communicating data between the volatile memory subsystem and the host system when the memory system is in a first mode of operation. The method of certain embodiments further includes storing a first copy of data from the volatile

memory subsystem to the non-volatile memory subsystem at a first time when the memory system is in a second mode of operation. The method may further include restoring the first copy of data from the non-volatile memory subsystem to the volatile memory subsystem and erasing the first copy of data from the non-volatile memory subsystem. In certain embodiments, the method also includes storing a second copy of data from the volatile memory subsystem to the non-volatile memory subsystem at a second time when the memory system is in the second mode of operation, wherein storing the second copy begins before the first copy is completely erased from the non-volatile memory subsystem.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of an example memory system compatible with certain embodiments described herein.

FIG. **2** is a block diagram of an example memory module with ECC (error-correcting code) having a volatile memory subsystem with nine volatile memory elements and a non-volatile memory subsystem with five non-volatile memory elements in accordance with certain embodiments described herein.

FIG. **3** is a block diagram of an example memory module having a microcontroller unit and logic element integrated into a single device in accordance with certain embodiments described herein.

FIGS. **4**A-**4**C schematically illustrate example embodiments of memory systems having volatile memory subsystems comprising registered dual in-line memory modules in accordance with certain embodiments described herein.

FIG. **5** schematically illustrates an example power module of a memory system in accordance with certain embodiments described herein.

FIG. **6** is a flowchart of an example method of providing a first voltage and a second voltage to a memory system including volatile and non-volatile memory subsystems.

FIG. **7** is a flowchart of an example method of controlling a memory system operatively coupled to a host system and which includes at least 100 percent more storage capacity in non-volatile memory than in volatile memory.

FIG. **8** schematically illustrates an example clock distribution topology of a memory system in accordance with certain embodiments described herein.

FIG. **9** is a flowchart of an example method of controlling a memory system operatively coupled to a host system, the method including operating a volatile memory subsystem at a reduced rate in a back-up mode.

FIG. **10** schematically illustrates an example topology of a connection to transfer data slices from two DRAM segments of a volatile memory subsystem of a memory system to a controller of the memory system.

FIG. **11** is a flowchart of an example method of controlling a memory system operatively coupled to a host system, the method including backing up and/or restoring a volatile memory subsystem in slices.

DETAILED DESCRIPTION

Certain embodiments described herein include a memory system which can communicate with a host system such as a disk controller of a computer system. The memory system can include volatile and non-volatile memory, and a controller. The controller backs up the volatile memory using the non-volatile memory in the event of a trigger condition. Trigger conditions can include, for example, a power failure, power reduction, request by the host system, etc. In order to power the system in the event of a power failure or reduction, the memory system can include a secondary power source which does not comprise a battery and may include, for example, a capacitor or capacitor array.

In certain embodiments, the memory system can be configured such that the operation of the volatile memory is not adversely affected by the non-volatile memory or by the controller when the volatile memory is interacting with the host system. For example, one or more isolation devices may isolate the non-volatile memory and the controller from the volatile memory when the volatile memory is interacting with the host system and may allow communication between the volatile memory and the non-volatile memory when the data of the volatile memory is being restored or backed-up. This configuration generally protects the operation of the volatile memory when isolated while providing backup and restore capability in the event of a trigger condition, such as a power failure.

In certain embodiments described herein, the memory system includes a power module which provides power to the various components of the memory system from different sources based on a state of the memory system in relation to a trigger condition (e.g., a power failure). The power module may switch the source of the power to the various components in order to efficiently provide power in the event of the power failure. For example, when no power failure is detected, the power module may provide power to certain components, such as the volatile memory, from system power while charging a secondary power source (e.g., a capacitor array). In the event of a power failure or other trigger condition, the power module may power the volatile memory elements using the previously charged secondary power source.

In certain embodiments, the power module transitions relatively smoothly from powering the volatile memory with system power to powering it with the secondary power source. For example, the memory system may power volatile memory with a third power source from the time the memory system detects that power failure is likely to occur until the time the memory system detects that the power failure has actually occurred.

In certain embodiments, the volatile memory system can be operated at a reduced frequency during backup and/or restore operations which can improve the efficiency of the system and save power. In some embodiments, during backup and/or restore operations, the volatile memory communicates with the non-volatile memory by writing and/or reading data words in bit-wise slices instead of by writing entire words at once. In certain embodiments, when each slice is being written to or read from the volatile memory the unused slice(s) of volatile memory is not active, which can reduce the power consumption of the system.

In yet other embodiments, the non-volatile memory can include at least 100 percent more storage capacity than the volatile memory. This configuration can allow the memory system to efficiently handle subsequent trigger conditions.

FIG. **1** is a block diagram of an example memory system **10** compatible with certain embodiments described herein. The memory system **10** can be coupled to a host computer system and can include a volatile memory subsystem **30**, a non-volatile memory subsystem **40**, and a controller **62** operatively coupled to the non-volatile memory subsystem **40**. In certain embodiments, the memory system **10** includes at least one circuit **52** configured to selectively operatively decouple the controller **62** from the volatile memory subsystem **30**.

In certain embodiments, the memory system **10** comprises a memory module. The memory system **10** may comprise a printed-circuit board (PCB) **20**. In certain embodiments, the

memory system **10** has a memory capacity of 512-MB, 1-GB, 2-GB, 4-GB, or 8-GB. Other volatile memory capacities are also compatible with certain embodiments described herein. In certain embodiments, the memory system **10** has a non-volatile memory capacity of 512-MB, 1-GB, 2-GB, 4-GB, 8-GB, 16-GB, or 32-GB. Other non-volatile memory capacities are also compatible with certain embodiments described herein. In addition, memory systems **10** having widths of 4 bytes, 8 bytes, 16 bytes, 32 bytes, or 32 bits, 64 bits, 128 bits, 256 bits, as well as other widths (in bytes or in bits), are compatible with embodiments described herein. In certain embodiments, the PCB **20** has an industry-standard form factor. For example, the PCB **20** can have a low profile (LP) form factor with a height of 30 millimeters and a width of 133.35 millimeters. In certain other embodiments, the PCB **20** has a very high profile (VHP) form factor with a height of 50 millimeters or more. In certain other embodiments, the PCB **20** has a very low profile (VLP) form factor with a height of 18.3 millimeters. Other form factors including, but not limited to, small-outline (SO-DIMM), unbuffered (UDIMM), registered (RDIMM), fully-buffered (FBDIMM), mini-DIMM, mini-RDIMM, VLP mini-DIMM, micro-DIMM, and SRAM DIMM are also compatible with certain embodiments described herein. For example, in other embodiments, certain non-DIMM form factors are possible such as, for example, single in-line memory module (SIMM), multi-media card (MMC), and small computer system interface (SCSI).

In certain preferred embodiments, the memory system **10** is in electrical communication with the host system. In other embodiments, the memory system **10** may communicate with a host system using some other type of communication, such as, for example, optical communication. Examples of host systems include, but are not limited to, blade servers, 1U servers, personal computers (PCs), and other applications in which space is constrained or limited. The memory system **10** can be in communication with a disk controller of a computer system, for example. The PCB **20** can comprise an interface **22** that is configured to be in electrical communication with the host system (not shown). For example, the interface **22** can comprise a plurality of edge connections which fit into a corresponding slot connector of the host system. The interface **22** of certain embodiments provides a conduit for power voltage as well as data, address, and control signals between the memory system **10** and the host system. For example, the interface **22** can comprise a standard 240-pin DDR2 edge connector.

The volatile memory subsystem **30** comprises a plurality of volatile memory elements **32** and the non-volatile memory subsystem **40** comprises a plurality of non-volatile memory elements **42**. Certain embodiments described herein advantageously provide non-volatile storage via the non-volatile memory subsystem **40** in addition to high-performance (e.g., high speed) storage via the volatile memory subsystem **30**. In certain embodiments, the first plurality of volatile memory elements **32** comprises two or more dynamic random-access memory (DRAM) elements. Types of DRAM elements **32** compatible with certain embodiments described herein include, but are not limited to, DDR, DDR2, DDR3, and synchronous DRAM (SDRAM). For example, in the block diagram of FIG. **1**, the first memory bank **30** comprises eight 64Mx8 DDR2 SDRAM elements **32**. The volatile memory elements **32** may comprise other types of memory elements such as static random-access memory (SRAM). In addition, volatile memory elements **32** having bit widths of 4, 8, 16, 32, as well as other bit widths, are compatible with certain embodiments described herein. Volatile memory elements **32** compatible with certain embodiments described herein have packaging which include, but are not limited to, thin small-outline package (TSOP), ball-grid-array (BGA), fine-pitch BGA (FBGA), micro-BGA (μBGA), mini-BGA (mBGA), and chip-scale packaging (CSP).

In certain embodiments, the second plurality of non-volatile memory elements **42** comprises one or more flash memory elements. Types of flash memory elements **42** compatible with certain embodiments described herein include, but are not limited to, NOR flash, NAND flash, ONE-NAND flash, and multi-level cell (MLC). For example, in the block diagram of FIG. **1**, the second memory bank **40** comprises 512 MB of flash memory organized as four 128 Mbx8 NAND flash memory elements **42**. In addition, non-volatile memory elements **42** having bit widths of 4, 8, 16, 32, as well as other bit widths, are compatible with certain embodiments described herein. Non-volatile memory elements **42** compatible with certain embodiments described herein have packaging which include, but are not limited to, thin small-outline package (TSOP), ball-grid-array (BGA), fine-pitch BGA (FBGA), micro-BGA (μBGA), mini-BGA (mBGA), and chip-scale packaging (CSP).

FIG. **2** is a block diagram of an example memory module **10** with ECC (error-correcting code) having a volatile memory subsystem **30** with nine volatile memory elements **32** and a non-volatile memory subsystem **40** with five non-volatile memory elements **42** in accordance with certain embodiments described herein. The additional memory element **32** of the first memory bank **30** and the additional memory element **42** of the second memory bank **40** provide the ECC capability. In certain other embodiments, the volatile memory subsystem **30** comprises other numbers of volatile memory elements **32** (e.g., 2, 3, 4, 5, 6, 7, more than 9). In certain embodiments, the non-volatile memory subsystem **40** comprises other numbers of non-volatile memory elements **42** (e.g., 2, 3, more than 5).

Referring to FIG. **1**, in certain embodiments, the logic element **70** comprises a field-programmable gate array (FPGA). In certain embodiments, the logic element **70** comprises an FPGA available from Lattice Semiconductor Corporation which includes an internal flash. In certain other embodiments, the logic element **70** comprises an FPGA available from another vendor. The internal flash can improve the speed of the memory system **10** and save physical space. Other types of logic elements **70** compatible with certain embodiments described herein include, but are not limited to, a programmable-logic device (PLD), an application-specific integrated circuit (ASIC), a custom-designed semiconductor device, a complex programmable logic device (CPLD). In certain embodiments, the logic element **70** is a custom device. In certain embodiments, the logic element **70** comprises various discrete electrical elements, while in certain other embodiments, the logic element **70** comprises one or more integrated circuits. FIG. **3** is a block diagram of an example memory module **10** having a microcontroller unit **60** and logic element **70** integrated into a single controller **62** in accordance with certain embodiments described herein. In certain embodiments, the controller **62** includes one or more other components. For example, in one embodiment, an FPGA without an internal flash is used and the controller **62** includes a separate flash memory component which stores configuration information to program the FPGA.

In certain embodiments, the at least one circuit **52** comprises one or more switches coupled to the volatile memory subsystem **30**, to the controller **62**, and to the host computer (e.g., via the interface **22**, as schematically illustrated by FIGS. **1-3**). The one or more switches are responsive to sig-

nals (e.g., from the controller **62**) to selectively operatively decouple the controller **62** from the volatile memory subsystem **30** and to selectively operatively couple the controller **62** to the volatile memory subsystem **30**. In addition, in certain embodiments, the at least one circuit **52** selectively operatively couples and decouples the volatile memory subsystem **30** and the host system.

In certain embodiments, the volatile memory subsystem **30** can comprise a registered DIMM subsystem comprising one or more registers **160** and a plurality of DRAM elements **180**, as schematically illustrated by FIG. **4**A. In certain such embodiments, the at least one circuit **52** can comprise one or more switches **172** coupled to the controller **62** (e.g., logic element **70**) and to the volatile memory subsystem **30** which can be actuated to couple and decouple the controller **62** to and from the volatile memory subsystem **30**, respectively. The memory system **10** further comprises one or more switches **170** coupled to the one or more registers **160** and to the plurality of DRAM elements **180** as schematically illustrated by FIG. **4**A. The one or more switches **170** can be selectively switched, thereby selectively operatively coupling the volatile memory subsystem **30** to the host system **150**. In certain other embodiments, as schematically illustrated by FIG. **4**B, the one or more switches **174** are also coupled to the one or more registers **160** and to a power source **162** for the one or more registers **160**. The one or more switches **174** can be selectively switched to turn power on or off to the one or more registers **160**, thereby selectively operatively coupling the volatile memory subsystem **30** to the host system **150**. As schematically illustrated by FIG. **4**C, in certain embodiments the at least one circuit **52** comprises a dynamic on-die termination (ODT) **176** circuit of the logic element **70**. For example, the logic element **70** can comprise a dynamic ODT circuit **176** which selectively operatively couples and decouples the logic element **70** to and from the volatile memory subsystem **30**, respectively. In addition, and similar to the example embodiment of FIG. **4**A described above, the one or more switches **170** can be selectively switched, thereby selectively operatively coupling the volatile memory subsystem **30** to the host system **150**.

Certain embodiments described herein utilize the non-volatile memory subsystem **40** as a flash "mirror" to provide backup of the volatile memory subsystem **30** in the event of certain system conditions. For example, the non-volatile memory subsystem **40** may backup the volatile memory subsystem **30** in the event of a trigger condition, such as, for example, a power failure or power reduction or a request from the host system. In one embodiment, the non-volatile memory subsystem **40** holds intermediate data results in a noisy system environment when the host computer system is engaged in a long computation. In certain embodiments, a backup may be performed on a regular basis. For example, in one embodiment, the backup may occur every millisecond in response to a trigger condition. In certain embodiments, the trigger condition occurs when the memory system **10** detects that the system voltage is below a certain threshold voltage. For example, in one embodiment, the threshold voltage is 10 percent below a specified operating voltage. In certain embodiments, a trigger condition occurs when the voltage goes above a certain threshold value, such as, for example, 10 percent above a specified operating voltage. In some embodiments, a trigger condition occurs when the voltage goes below a threshold or above another threshold. In various embodiments, a backup and/or restore operation may occur in reboot and/or non-reboot trigger conditions.

As schematically illustrated by FIGS. **1** and **2**, in certain embodiments, the controller **62** may comprise a microcontroller unit (MCU) **60** and a logic element **70**. In certain embodiments, the MCU **60** provides memory management for the non-volatile memory subsystem **40** and controls data transfer between the volatile memory subsystem **30** and the non-volatile memory subsystem **40**. The MCU **60** of certain embodiments comprises a 16-bit microcontroller, although other types of microcontrollers are also compatible with certain embodiments described herein. As schematically illustrated by FIGS. **1** and **2**, the logic element **70** of certain embodiments is in electrical communication with the non-volatile memory subsystem **40** and the MCU **60**. The logic element **70** can provide signal level translation between the volatile memory elements **32** (e.g., 1.8V SSTL-2 for DDR2 SDRAM elements) and the non-volatile memory elements **42** (e.g., 3V TTL for NAND flash memory elements). In certain embodiments, the logic element **70** is also programmed to perform address/address translation between the volatile memory subsystem **30** and the non-volatile memory subsystem **40**. In certain preferred embodiments, 1-NAND type flash are used for the non-volatile memory elements **42** because of their superior read speed and compact structure.

The memory system **10** of certain embodiments is configured to be operated in at least two states. The at least two states can comprise a first state in which the controller **62** and the non-volatile memory subsystem **40** are operatively decoupled (e.g., isolated) from the volatile memory subsystem **30** by the at least one circuit **52** and a second state in which the volatile memory subsystem **30** is operatively coupled to the controller **62** to allow data to be communicated between the volatile memory subsystem **30** and the non-volatile memory subsystem **40** via the controller **62**. The memory system **10** may transition from the first state to the second state in response to a trigger condition, such as when the memory system **10** detects that there is a power interruption (e.g., power failure or reduction) or a system hang-up.

The memory system **10** may further comprise a voltage monitor **50**. The voltage monitor circuit **50** monitors the voltage supplied by the host system via the interface **22**. Upon detecting a low voltage condition (e.g., due to a power interruption to the host system), the voltage monitor circuit **50** may transmit a signal to the controller **62** indicative of the detected condition. The controller **62** of certain embodiments responds to the signal from the voltage monitor circuit **50** by transmitting a signal to the at least one circuit **52** to operatively couple the controller to the volatile memory system **30**, such that the memory system **10** enters the second state. For example, the voltage monitor **50** may send a signal to the MCU **60** which responds by accessing the data on the volatile memory system **30** and by executing a write cycle on the non-volatile memory subsystem **40**. During this write cycle, data is read from the volatile memory subsystem **30** and is transferred to the non-volatile memory subsystem **40** via the MCU **60**. In certain embodiments, the voltage monitor circuit **50** is part of the controller **62** (e.g., part of the MCU **60**) and the voltage monitor circuit **50** transmits a signal to the other portions of the controller **62** upon detecting a power threshold condition.

The isolation or operational decoupling of the volatile memory subsystem **30** from the non-volatile memory subsystem in the first state can preserve the integrity of the operation of the memory system **10** during periods of operation in which signals (e.g., data) are transmitted between the host system and the volatile memory subsystem **30**. For example, in one embodiment during such periods of operation, the controller **62** and the non-volatile memory subsystem **40** do not add a significant capacitive load to the volatile memory system **30** when the memory system **10** is in the first state. In certain such embodiments, the capacitive

load of the controller **62** and the non-volatile memory subsystem **40** do not significantly affect the signals propagating between the volatile memory subsystem **30** and the host system. This can be particularly advantageous in relatively high-speed memory systems where loading effects can be significant. In one preferred embodiment, the at least one circuit **52** comprises an FSA1208 Low-Power, Eight-Port, Hi-Speed Isolation Switch from Fairchild Semiconductor. In other embodiments, the at least one circuit **52** comprises other types of isolation devices.

Power may be supplied to the volatile memory subsystem **30** from a first power supply (e.g., a system power supply) when the memory system **10** is in the first state and from a second power supply **80** when the memory system **10** is in the second state. In certain embodiments, the memory system **10** is in the first state when no trigger condition (e.g., a power failure) is present and the memory system **10** enters the second state in response to a trigger condition. In certain embodiments, the memory system **10** has a third state in which the controller **62** is operatively decoupled from the volatile memory subsystem **30** and power is supplied to the volatile memory subsystem **30** from a third power supply (not shown). For example, in one embodiment the third power supply may provide power to the volatile memory subsystem **30** when the memory system **10** detects that a trigger condition is likely to occur but has not yet occurred.

In certain embodiments, the second power supply **80** does not comprise a battery. Because a battery is not used, the second power supply **80** of certain embodiments may be relatively easy to maintain, does not generally need to be replaced, and is relatively environmentally friendly. In certain embodiments, as schematically illustrated by FIGS. **1-3**, the second power supply **80** comprises a step-up transformer **82**, a step-down transformer **84**, and a capacitor bank **86** comprising one or more capacitors (e.g., double-layer capacitors). In one example embodiment, capacitors may take about three to four minutes to charge and about two minutes to discharge. In other embodiments, the one or more capacitors may take a longer time or a shorter time to charge and/or discharge. For example, in certain embodiments, the second power supply **80** is configured to power the volatile memory subsystem **30** for less than thirty minutes. In certain embodiments, the second power supply **80** may comprise a battery. For example, in certain embodiments, the second power supply **80** comprises a battery and one or more capacitors and is configured to power the volatile memory subsystem **30** for no more than thirty minutes.

In certain embodiments, the capacitor bank **86** of the second power supply **80** is charged by the first power supply while the memory system **10** is in the first state. As a result, the second power supply **80** is fully charged when the memory system **10** enters the second state. The memory system **10** and the second power supply **80** may be located on the same printed circuit board **20**. In other embodiments, the second power supply **80** may not be on the same printed circuit board **20** and may be tethered to the printed circuit board **20**, for example.

When operating in the first state, in certain embodiments, the step-up transformer **82** keeps the capacitor bank **86** charged at a peak value. In certain embodiments, the step-down transformer **84** acts as a voltage regulator to ensure that regulated voltages are supplied to the memory elements (e.g., 1.8V to the volatile DRAM elements **32** and 3.0V to the non-volatile flash memory elements **42**) when operating in the second state (e.g., during power down). In certain embodiments, as schematically illustrated by FIGS. **1-3**, the memory module **10** further comprises a switch **90** (e.g., FET switch) that switches power provided to the controller **62**, the volatile memory subsystem **30**, and the non-volatile memory subsystem **40**, between the power from the second power supply **80** and the power from the first power supply (e.g., system power) received via the interface **22**. For example, the switch **90** may switch from the first power supply to the second power supply **80** when the voltage monitor **50** detects a low voltage condition. The switch **90** of certain embodiments advantageously ensures that the volatile memory elements **32** and non-volatile memory elements **42** are powered long enough for the data to be transferred from the volatile memory elements **32** and stored in the non-volatile memory elements **42**. In certain embodiments, after the data transfer is complete, the switch **90** then switches back to the first power supply and the controller **62** transmits a signal to the at least one circuit **52** to operatively decouple the controller **62** from the volatile memory subsystem **30**, such that the memory system **10** re-enters the first state.

When the memory system **10** re-enters the first state, data may be transferred back from the non-volatile memory subsystem **40** to the volatile memory subsystem **30** via the controller **62**. The host system can then resume accessing the volatile memory subsystem **30** of the memory module **10**. In certain embodiments, after the memory system **10** enters or re-enters the first state (e.g., after power is restored), the host system accesses the volatile memory subsystem **30** rather than the non-volatile memory subsystem **40** because the volatile memory elements **32** have superior read/write characteristics. In certain embodiments, the transfer of data from the volatile memory bank **30** to the non-volatile memory bank **40**, or from the non-volatile memory bank **40** to the volatile memory bank **30**, takes less than one minute per GB.

In certain embodiments, the memory system **10** protects the operation of the volatile memory when communicating with the host-system and provides backup and restore capability in the event of a trigger condition such as a power failure. In certain embodiments, the memory system **10** copies the entire contents of the volatile memory subsystem **30** into the non-volatile memory subsystem **40** on each backup operation. Moreover, in certain embodiments, the entire contents of the non-volatile memory subsystem **40** are copied back into the volatile memory subsystem **30** on each restore operation. In certain embodiments, the entire contents of the non-volatile memory subsystem **40** are accessed for each backup and/or restore operation, such that the non-volatile memory subsystem **40** (e.g., flash memory subsystem) is used generally uniformly across its memory space and wear-leveling is not performed by the memory system **10**. In certain embodiments, avoiding wear-leveling can decrease cost and complexity of the memory system **10** and can improve the performance of the memory system **10**. In certain other embodiments, the entire contents of the volatile memory subsystem **30** are not copied into the non-volatile memory subsystem **40** on each backup operation, but only a partial copy is performed. In certain embodiments, other management capabilities such as bad-block management and error management for the flash memory elements of the non-volatile memory subsystem **40** are performed in the controller **62**.

The memory system **10** generally operates as a write-back cache in certain embodiments. For example, in one embodiment, the host system (e.g., a disk controller) writes data to the volatile memory subsystem **30** which then writes the data to non-volatile storage which is not part of the memory system **10**, such as, for example, a hard disk. The disk controller may wait for an acknowledgment signal from the memory system **10** indicating that the data has been written to the hard disk or is otherwise secure. The memory system **10** of certain

embodiments can decrease delays in the system operation by indicating that the data has been written to the hard disk before it has actually done so. In certain embodiments, the memory system **10** will still be able to recover the data efficiently in the event of a power outage because of the backup and restore capabilities described herein. In certain other embodiments, the memory system **10** may be operated as a write-through cache or as some other type of cache.

FIG. **5** schematically illustrates an example power module **100** of the memory system **10** in accordance with certain embodiments described herein. The power module **100** provides power to the various components of the memory system **10** using different elements based on a state of the memory system **10** in relation to a trigger condition. In certain embodiments, the power module **100** comprises one or more of the components described above with respect to FIG. **1**. For example, in certain embodiments, the power module **100** includes the second power supply **80** and the switch **90**.

The power module **100** provides a plurality of voltages to the memory system **10** comprising non-volatile and volatile memory subsystems **30**, **40**. The plurality of voltages comprises at least a first voltage **102** and a second voltage **104**. The power module **100** comprises an input **106** providing a third voltage **108** to the power module **100** and a voltage conversion element **120** configured to provide the second voltage **104** to the memory system **10**. The power module **100** further comprises a first power element **130** configured to selectively provide a fourth voltage **110** to the conversion element **120**. In certain embodiments, the first power element **130** comprises a pulse-width modulation power controller. For example, in one example embodiment, the first power element **130** is configured to receive a 1.8V input system voltage as the third voltage **108** and to output a modulated 5V output as the fourth voltage **110**.

The power module **100** further comprises a second power element **140** can be configured to selectively provide a fifth voltage **112** to the conversion element **120**. The power module **100** can be configured to selectively provide the first voltage **102** to the memory system **10** either from the conversion element **120** or from the input **106**.

The power module **100** can be configured to be operated in at least three states in certain embodiments. In a first state, the first voltage **102** is provided to the memory system **10** from the input **106** and the fourth voltage **110** is provided to the conversion element **120** from the first power element **130**. In a second state, the fourth voltage **110** is provided to the conversion element **120** from the first power element **130** and the first voltage **102** is provided to the memory system **10** from the conversion element **120**. In the third state, the fifth voltage **112** is provided to the conversion element **120** from the second power element **140** and the first voltage **104** is provided to the memory system **10** from the conversion element **120**.

In certain embodiments, the power module **100** transitions from the first state to the second state upon detecting that a trigger condition is likely to occur and transitions from the second state to the third state upon detecting that the trigger condition has occurred. For example, the power module **100** may transition to the second state when it detects that a power failure is about to occur and transitions to the third state when it detects that the power failure has occurred. In certain embodiments, providing the first voltage **102** in the second state from the first power element **130** rather than from the input **106** allows a smoother transition from the first state to the third state. For example, in certain embodiments, providing the first voltage **102** from the first power element **130** has capacitive and other smoothing effects. In addition, switching the point of power transition to be between the conversion element **120** and the first and second power elements **130**, **140** (e.g., the sources of the pre-regulated fourth voltage **110** in the second state and the pre-regulated fifth voltage **112** in the third state) can smooth out potential voltage spikes.

In certain embodiments, the second power element **140** does not comprise a battery and may comprise one or more capacitors. For example, as schematically illustrated in FIG. **4**, the second power element **140** comprises a capacitor array **142**, a buck-boost converter **144** which adjusts the voltage for charging the capacitor array and a voltage/current limiter **146** which limits the charge current to the capacitor array **142** and stops charging the capacitor array **142** when it has reached a certain charge voltage. In one example embodiment, the capacitor array **142** comprises two 50 farad capacitors capable of holding a total charge of 4.6V. For example, in one example embodiment, the buck-boost converter **144** receives a 1.8V system voltage (first voltage **108**) and boosts the voltage to 4.3V which is outputted to the voltage current limiter **146**. The voltage/current limiter **146** limits the current going to the capacitor array **142** to 1A and stops charging the array **142** when it is charged to 4.3V. Although described with respect to certain example embodiments, one of ordinary skill will recognize from the disclosure herein that the second power element **140** may include alternative embodiments. For example, different components and/or different value components may be used. For example, in other embodiments, a pure boost converter may be used instead of a buck-boost converter. In another embodiment, only one capacitor may be used instead of a capacitor array **142**.

The conversion element **120** can comprise one or more buck converters and/or one or more buck-boost converters. The conversion element **120** may comprise a plurality of sub-blocks **122**, **124**, **126** as schematically illustrated by FIG. **4**, which can provide more voltages in addition to the second voltage **104** to the memory system **10**. The sub-blocks may comprise various converter circuits such as buck-converters, boost converters, and buck-boost converter circuits for providing various voltage values to the memory system **10**. For example, in one embodiment, sub-block **122** comprises a buck converter, sub-block **124** comprises a dual buck converter, and sub-block **126** comprises a buck-boost converter as schematically illustrated by FIG. **4**. Various other components for the sub-blocks **122**, **124**, **126** of the conversion element **120** are also compatible with certain embodiments described herein. In certain embodiments, the conversion element **120** receives as input either the fourth voltage **110** from the first power element **130** or the fifth voltage **112** from the second power element **140**, depending on the state of the power module **100**, and reduces the input to an appropriate amount for powering various components of the memory system. For example, the buck-converter of sub-block **122** can provide 1.8V at 2 A for about 60 seconds to the volatile memory elements **32** (e.g., DRAM), the non-volatile memory elements **42** (e.g., flash), and the controller **62** (e.g., an FPGA) in one embodiment. The sub-block **124** can provide the second voltage **104** as well as another reduced voltage **105** to the memory system **10**. In one example embodiment, the second voltage **104** is 2.5V and is used to power the at least one circuit **52** (e.g., isolation device) and the other reduced voltage **105** is 1.2V and is used to power the controller **62** (e.g., FPGA). The sub-block **126** can provide yet another voltage **107** to the memory system **10**. For example, the voltage **107** may be 3.3V and may be used to power both the controller **62** and the at least one circuit **52**.

Although described with respect to certain example embodiments, one of ordinary skill will recognize from the

disclosure herein that the conversion element **120** may include alternative embodiments. For example, there may be more or less sub-blocks which may comprise other types of converters (e.g., pure boost converters) or which may produce different voltage values. In one embodiment, the volatile memory elements **32** and non-volatile memory elements **42** are powered using independent voltages and are not both powered using the first voltage **102**.

FIG. **6** is a flowchart of an example method **200** of providing a first voltage **102** and a second voltage **104** to a memory system **10** including volatile and nonvolatile memory subsystems **30**, **40**. While the method **200** is described herein by reference to the memory system **10** schematically illustrated by FIGS. **1-4**, other memory systems are also compatible with embodiments of the method **200**. During a first condition, the method **200** comprises providing the first voltage **102** to the memory system **10** from an input power supply **106** and providing the second voltage **104** to the memory system **10** from a first power subsystem in operational block **210**. For example, in one embodiment, the first power subsystem comprises the first power element **130** and the voltage conversion element **120** described above with respect to FIG. **4**. In other embodiments, other first power subsystems are used.

The method **200** further comprises detecting a second condition in operational block **220**. In certain embodiments, detecting the second condition comprises detecting that a trigger condition is likely to occur. During the second condition, the method **200** comprises providing the first voltage **102** and the second voltage **104** to the memory system **10** from the first power subsystem in an operational block **230**. For example, referring to FIG. **4**, a switch **148** can be toggled to provide the first voltage **102** from the conversion element **120** rather than from the input power supply.

The method **200** further comprises charging a second power subsystem in operational block **240**. In certain embodiments, the second power subsystem comprises the second power element **140** or another power supply that does not comprise a battery. For example, in one embodiment, the second power subsystem comprises the second power element **140** and the voltage conversion element **120** described above with respect to FIG. **4**. In other embodiments, some other second power subsystem is used.

The method **200** further comprises detecting a third condition in an operational block **250** and during the third condition, providing the first voltage **102** and the second voltage **104** to the memory system **10** from the second power subsystem **140** in an operational block **250**. In certain embodiments, detecting the third condition comprises detecting that the trigger condition has occurred. The trigger condition may comprise various conditions described herein. In various embodiments, for example, the trigger condition comprises a power reduction, power failure, or system hang-up. The operational blocks of the method **200** may be performed in different orders in various embodiments. For example, in certain embodiments, the second power subsystem **140** is charged before detecting the second condition.

In certain embodiments, the memory system **10** comprises a volatile memory subsystem **30** and a non-volatile memory subsystem **40** comprising at least 100 percent more storage capacity than does the volatile memory subsystem. The memory system **10** also comprises a controller **62** operatively coupled to the volatile memory subsystem **30** and operatively coupled to the non-volatile memory subsystem **40**. The controller **62** can be configured to allow data to be communicated between the volatile memory subsystem **30** and the host system when the memory system **10** is operating in a first state and to allow data to be communicated between the volatile memory subsystem **30** and the non-volatile memory subsystem **40** when the memory system **10** is operating in a second state.

Although the memory system **10** having extra storage capacity of the non-volatile memory subsystem **40** has been described with respect to certain embodiments, alternative configurations exist. For example, in certain embodiments, there may be more than 100 percent more storage capacity in the non-volatile memory subsystem **40** than in the volatile memory subsystem **30**. In various embodiments, there may be at least 200, 300, or 400 percent more storage capacity in the non-volatile memory subsystem **40** than in the volatile memory subsystem **30**. In other embodiments, the non-volatile memory subsystem **40** includes at least some other integer multiples of the storage capacity of the volatile memory subsystem **30**. In some embodiments, the non-volatile memory subsystem **40** includes a non-integer multiple of the storage capacity of the volatile memory subsystem **30**. In one embodiment, the non-volatile memory subsystem **40** includes less than 100 percent more storage capacity than does the volatile memory subsystem **30**.

The extra storage capacity of the non-volatile memory subsystem **40** can be used to improve the backup capability of the memory system **10**. In certain embodiments in which data can only be written to portions of the non-volatile memory subsystem **40** which do not contain data (e.g., portions which have been erased), the extra storage capacity of the non-volatile memory subsystem **40** allows the volatile memory subsystem **30** to be backed up in the event of a subsequent power failure or other trigger event. For example, the extra storage capacity of the non-volatile memory subsystem **40** may allow the memory system **10** to backup the volatile memory subsystem **30** efficiently in the event of multiple trigger conditions (e.g., power failures). In the event of a first power failure, for example, the data in the volatile memory system **30** is copied to a first, previously erased portion of the non-volatile memory subsystem **40** via the controller **62**. Since the non-volatile memory subsystem **40** has more storage capacity than does the volatile memory subsystem **30**, there is a second portion of the non-volatile memory subsystem **40** which does not have data from the volatile memory subsystem **30** copied to it and which remains free of data (e.g., erased). Once system power is restored, the controller **62** of the memory system **10** restores the data to the volatile memory subsystem **30** by copying the backed-up data from the non-volatile memory subsystem **40** back to the volatile memory subsystem **30**. After the data is restored, the memory system **10** erases the non-volatile memory subsystem **40**. While the first portion of the non-volatile memory subsystem **40** is being erased, it may be temporarily un-accessible.

If a subsequent power failure occurs before the first portion of the non-volatile memory subsystem **40** is completely erased, the volatile memory subsystem **30** can be backed-up or stored again in the second portion of the non-volatile memory subsystem **40** as described herein. In certain embodiments, the extra storage capacity of the non-volatile memory subsystem **40** may allow the memory system **10** to operate more efficiently. For example, because of the extra storage capacity of the non-volatile memory subsystem **40**, the memory system **10** can handle a higher frequency of trigger events that is not limited by the erase time of the non-volatile memory subsystem **40**.

FIG. **7** is a flowchart of an example method **300** of controlling a memory system **10** operatively coupled to a host system and which includes a volatile memory subsystem **30** and a non-volatile memory subsystem **40**. In certain embodiments, the non-volatile memory subsystem **40** comprises at least 100

percent more storage capacity than does the volatile memory subsystem **30** as described herein. While the method **300** is described herein by reference to the memory system **10** schematically illustrated by FIGS. **1-3**, the method **300** can be practiced using other memory systems in accordance with certain embodiments described herein. In an operational block **310**, the method **300** comprises communicating data between the volatile memory subsystem **30** and the host system when the memory system **10** is in a first mode of operation. The method **300** further comprises storing a first copy of data from the volatile memory subsystem **30** to the non-volatile memory subsystem **40** at a first time when the memory system **10** is in a second mode of operation in an operational block **320**.

In an operational block **330**, the method **300** comprises restoring the first copy of data from the non-volatile memory subsystem **40** to the volatile memory subsystem **30**. The method **300** further comprises erasing the first copy of data from the non-volatile memory subsystem **40** in an operational block **340**. The method further comprises storing a second copy of data from the volatile memory subsystem **30** to the non-volatile memory subsystem **40** at a second time when the memory system **10** is in the second mode of operation in an operational block **350**. Storing the second copy begins before the first copy is completely erased from the non-volatile memory subsystem **40**.

In some embodiments, the memory system **10** enters the second mode of operation in response to a trigger condition, such as a power failure. In certain embodiments, the first copy of data and the second copy of data are stored in separate portions of the nonvolatile memory subsystem **40**. The method **300** can also include restoring the second copy of data from the non-volatile memory subsystem **40** to the volatile memory subsystem **30** in an operational block **360**. The operational blocks of method **300** referred to herein may be performed in different orders in various embodiments. For example, in some embodiments, the second copy of data is restored to the volatile memory subsystem **30** at operational block **360** before the first copy of data is completely erased in the operational block **340**.

FIG. **8** schematically illustrates an example clock distribution topology **400** of a memory system **10** in accordance with certain embodiments described herein. The clock distribution topology **400** generally illustrates the creation and routing of the clock signals provided to the various components of the memory system **10**. A clock source **402** such as, for example, a 25 MHz oscillator, generates a clock signal. The clock source **402** may feed a clock generator **404** which provides a clock signal **406** to the controller **62**, which may be an FPGA. In one embodiment, the clock generator **404** generates a 125 MHz clock signal **406**. The controller **62** receives the clock signal **406** and uses it to clock the controller **62** master state control logic. For example, the master state control logic may control the general operation of an FPGA controller **62**.

The clock signal **406** can also be input into a clock divider **410** which produces a frequency-divided version of the clock signal **406**. In an example embodiment, the clock divider **410** is a divide by two clock divider and produces a 62.5 MHz clock signal in response to the 125 MHz clock signal **406**. A non-volatile memory phase-locked loop (PLL) block **412** can be included (e.g., in the controller **62**) which distributes a series of clock signals to the non-volatile memory subsystem **40** and to associated control logic. For example, a series of clock signals **414** can be sent from the controller **62** to the non-volatile memory subsystem **40**. Another clock signal **416** can be used by the controller logic which is dedicated to controlling the non-volatile memory subsystem **40**. For example, the clock signal **416** may clock the portion of the controller **62** which is dedicated to generating address and/or control lines for the non-volatile memory subsystem **40**. A feedback clock signal **418** is fed back into the non-volatile memory PLL block **412**. In one embodiment, the PLL block **412** compares the feedback clock **418** to the reference clock **411** and varies the phase and frequency of its output until the reference **411** and feedback **418** clocks are phase and frequency matched.

A version of the clock signal **406** such as the backup clock signal **408** may be sent from the controller to the volatile memory subsystem **30**. The clock signal **408** may be, for example, a differential version of the clock signal **406**. As described herein, the backup clock signal **408** may be used to clock the volatile memory subsystem **30** when the memory system **10** is backing up the data from the volatile memory subsystem **30** into the non-volatile memory subsystem **40**. In certain embodiments, the backup clock signal **408** may also be used to clock the volatile memory subsystem **30** when the memory system **10** is copying the backed-up data back into the volatile memory subsystem **30** from the non-volatile memory subsystem **40** (also referred to as restoring the volatile memory subsystem **30**). The volatile memory subsystem **30** may normally be run at a higher frequency (e.g., DRAM running at 400 MHz) than the non-volatile memory subsystem **40** (e.g., flash memory running at 62.5 MHz) when communicating with the host system (e.g., when no trigger condition is present). However, in certain embodiments the volatile memory subsystem **30** may be operated at a reduced frequency (e.g., at twice the frequency of the non-volatile memory subsystem **40**) without introducing significant delay into the system during backup operation and/or restore operations. Running the volatile memory subsystem **30** at the reduced frequency during a backup and/or restore operation may advantageously reduce overall power consumption of the memory system **10**.

In one embodiment, the backup clock **408** and the volatile memory system clock signal **420** are received by a multiplexer **422**, as schematically illustrated by FIG. **8**. The multiplexer **422** can output either the volatile memory system clock signal **420** or the backup clock signal **408** depending on the backup state of the memory system **10**. For example, when the memory system **10** is not performing a backup or restore operation and is communicating with the host system (e.g., normal operation), the volatile memory system clock signal **420** may be provided by the multiplexer **422** to the volatile memory PLL block **424**. When the memory system **10** is performing a backup (or restore) operation, the backup clock signal **408** may be provided.

The volatile memory PLL block **424** receives the volatile memory reference clock signal **423** from the multiplexer **422** and can generate a series of clock signals which are distributed to the volatile memory subsystem **30** and associated control logic. For example, in one embodiment, the PLL block **424** generates a series of clock signals **426** which clock the volatile memory elements **32**. A clock signal **428** may be used to clock control logic associated with the volatile memory elements, such as one or more registers (e.g., the one or more registers of a registered DIMM). Another clock signal **430** may be sent to the controller **62**. A feedback clock signal **432** is fed back into the volatile memory PLL block **424**. In one embodiment, the PLL block **424** compares the feedback clock signal **432** to the reference clock signal **423** and varies the phase and frequency of its output until the reference clock signal **423** and the feedback clock signal **432** clocks are phase and frequency matched.

The clock signal **430** may be used by the controller **62** to generate and distribute clock signals which will be used by controller logic which is configured to control the volatile memory subsystem **30**. For example, control logic in the controller **62** may be used to control the volatile memory subsystem **30** during a backup or restore operation. The clock signal **430** may be used as a reference clock signal for the PLL block **434** which can generate one or more clocks **438** used by logic in the controller **62**. For example, the PLL block **434** may generate one or more clock signals **438** used to drive logic circuitry associated with controlling the volatile memory subsystem **30**. In certain embodiments, the PLL block **434** includes a feedback clock signal **436** and operates in a similar manner to other PLL blocks described herein.

The clock signal **430** may be used as a reference clock signal for the PLL block **440** which may generate one or more clock signals used by a sub-block **442** to generate one or more other clock signals **444**. In one embodiment, for example, the volatile memory subsystem **30** comprises DDR2 SDRAM elements and the sub-block **442** generates one or more DDR2 compatible clock signals **444**. A feedback clock signal **446** is fed back into the PLL block **440**. In certain embodiments, the PLL block **440** operates in a similar manner to other PLL blocks described herein.

While described with respect to the example embodiment of FIG. **8**, various alternative clock distribution topologies are possible. For example, one or more of the clock signals have a different frequency in various other embodiments. In some embodiments, one or more of the clocks shown as differential signals are single ended signals. In one embodiment, the volatile memory subsystem **30** operates on the volatile memory clock signal **420** and there is no backup clock signal **408**. In some embodiments, the volatile memory subsystem **30** is operated at a reduced frequency during a backup operation and not during a restore operation. In other embodiments, the volatile memory subsystem **30** is operated at a reduced frequency during a restore operation and not during a backup operation.

FIG. **9** is a flowchart of an example method **500** of controlling a memory system **10** operatively coupled to a host system. Although described with respect to the memory system **10** described herein, the method **500** is compatible with other memory systems. The memory system **10** may include a clock distribution topology **400** similar to the one described above with respect to FIG. **8** or another clock distribution topology. The memory system **10** can include a volatile memory subsystem **30** and a non-volatile memory subsystem **40**.

In an operational block **510**, the method **500** comprises operating the volatile memory subsystem **30** at a first frequency when the memory system **10** is in a first mode of operation in which data is communicated between the volatile memory subsystem **30** and the host system. In an operational block **520**, the method **500** comprises operating the non-volatile memory subsystem **40** at a second frequency when the memory system **10** is in a second mode of operation in which data is communicated between the volatile memory subsystem **30** and the non-volatile memory subsystem **40**. The method **500** further comprises operating the volatile memory subsystem **30** at a third frequency in an operational block **530** when the memory system **10** is in the second mode of operation. In certain embodiments, the memory system **10** is not powered by a battery when it is in the second mode of operation. The memory system **10** may switch from the first mode of operation to the second mode of operation in response to a trigger condition. The trigger condition may be any trigger condition described herein such as, for example, a power failure condition. In certain embodiments, the second mode of operation includes both backup and restore operations as described herein. In other embodiments, the second mode of operation includes backup operations but not restore operations. In yet other embodiments, the second mode of operation includes restore operations but not backup operations.

The third frequency can be less than the first frequency. For example, the third frequency can be approximately equal to the second frequency. In certain embodiments, the reduced frequency operation is an optional mode. In yet other embodiments, the first, second and/or third frequencies are configurable by a user or by the memory system **10**.

FIG. **10** schematically illustrates an example topology of a connection to transfer data slices from two DRAM segments **630**, **640** of a volatile memory subsystem **30** of a memory system **10** to a controller **62** of the memory system **10**. While the example of FIG. **10** shows a topology including two DRAM segments **630**, **640** for the purposes of illustration, each address location of the volatile memory subsystem **30** comprises more than the two segments in certain embodiments. The data lines **632**, **642** from the first DRAM segment **630** and the second DRAM segment **640** of the volatile memory subsystem **30** are coupled to switches **650**, **652** which are coupled to the controller **62** (e.g., logic element **70**) of the memory system **10**. The chip select lines **634**, **644** and the self-refresh lines **636**, **646** (e.g., CKe signals) of the first and second DRAM segments **630**, **640**, respectively, are coupled to the controller **62**. In certain embodiments, the controller **62** comprises a buffer (not shown) which is configured to store data from the volatile memory subsystem **30**. In certain embodiments, the buffer is a first-in, first out buffer (FIFO). In certain embodiments, data slices from each DRAM segment **630**, **640** comprise a portion of the volatile memory subsystem data bus. In one embodiment, for example, the volatile memory subsystem **30** comprises a 72-bit data bus (e.g., each data word at each addressable location is 72 bits wide and includes, for example, 64 bits of accessible SDRAM and 8 bits of ECC), the first data slice from the first DRAM segment **630** may comprise 40 bits of the data word, and the second data slice from the second DRAM segment **640** may comprise the remaining 32 bits of the data word. Certain other embodiments comprise data buses and/or data slices of different sizes.

In certain embodiments, the switches **650**, **652** can each be selectively switched to selectively operatively couple the data lines **632**, **642**, respectively from the first and second DRAM segments **630**, **640** to the controller **62**. The chip select lines **634**, **644** enable the first and second DRAM segments **630**, **640**, respectively, of the volatile memory subsystem **30**, and the self-refresh lines **636**, **646** toggle the first and second DRAM segments **630**, **640**, respectively, from self-refresh mode to active mode. In certain embodiments, the first and second DRAM segments **630**, **640** maintain stored information but are not accessible when they are in self-refresh mode, and maintain stored information and are accessible when they are in active mode.

In certain embodiments, when the memory system **10** is backing up the volatile memory system **30**, data slices from only one of the two DRAM segments **630**, **640** at a time are sent to the controller **62**. For example, when the first slice is being written to the controller **62** during a back-up, the controller **62** sends a signal via the CKe line **636** to the first DRAM segment **630** to put the first DRAM segment **630** in active mode. In certain embodiments, the data slice from the first DRAM segment **630** for multiple words (e.g., a block of words) is written to the controller **62** before writing the sec-

ond data slice from the second DRAM segment **640** to the controller **62**. While the first data slice is being written to the controller **62**, the controller **62** also sends a signal via the CKe line **646** to put the second DRAM segment **640** in self-refresh mode. Once the first data slice for one word or for a block of words is written to the controller **62**, the controller **62** puts the first DRAM segment **630** into self-refresh mode by sending a signal via the CKe line **636** to the first DRAM segment **640**. The controller **62** also puts the second DRAM segment **640** into active mode by sending a signal via the CKe line **646** to the DRAM segment **640**. The second slice for a word or for a block of words is written to the controller **62**. In certain embodiments, when the first and second data slices are written to the buffer in the controller **62**, the controller **62** combines the first and second data slices **630**, **640** into complete words or blocks of words and then writes each complete word or block of words to the non-volatile memory subsystem **40**. In certain embodiments, this process is called "slicing" the volatile memory subsystem **30**.

In certain embodiments, the data may be sliced in a restore operation as well as, or instead of, during a backup operation. For example, in one embodiment, the non-volatile memory elements **42** write each backed-up data word to the controller **62** which writes a first slice of the data word to the volatile memory subsystem **30** and then a second slice of the data word to the volatile memory subsystem **30**. In certain embodiments, slicing the volatile memory subsystem **30** during a restore operation may be performed in a manner generally inverse to slicing the volatile memory subsystem **30** during a backup operation.

FIG. **11** is a flowchart of an example method **600** of controlling a memory system **10** operatively coupled to a host system and which includes a volatile memory subsystem **30** and a non-volatile memory subsystem **40**. Although described with respect to the memory system **10** described herein with respect to FIGS. **1-3** and **10**, the method **600** is compatible with other memory systems. The method **600** comprises communicating data words between the volatile memory subsystem **40** and the host system when the memory system **10** is in a first mode of operation in an operational block **610**. For example, the memory system **10** may be in the first mode of operation when no trigger condition has occurred and the memory system is not performing a backup and/or restore operation or is not being powered by a secondary power supply.

In an operational block **620**, the method further comprises transferring data words from the volatile memory subsystem **30** to the non-volatile memory subsystem **40** when the memory system **10** is in a second mode of operation. In certain embodiments, each data word comprises the data stored in a particular address of the memory system **10**. The memory system **10** may enter the second mode of operation, for example, when a trigger condition (e.g., a power failure) occurs. In certain embodiments, transferring each data word comprises storing a first portion (also referred to as a slice) of the data word in a buffer in an operational block **622**, storing a second portion of the data word in the buffer in an operational block **624**, and writing the entire data word from the buffer to the non-volatile memory subsystem **40** in an operational block **626**.

In one example embodiment, the data word may be a 72 bit data word (e.g., 64 bits of accessible SDRAM and 8 bits of ECC), the first portion (or "slice") may comprise 40 bits of the data word, and the second portion (or "slice") may comprise the remaining 32 bits of the data word. In certain embodiments, the buffer is included in the controller **62**. For example, in one embodiment, the buffer is a first-in, first-out buffer implemented in the controller **62** which comprises an FPGA. The method **600** may generally be referred to as "slicing" the volatile memory during a backup operation. In the example embodiment, the process of "slicing" the volatile memory during a backup includes bringing the 32-bit slice out of self refresh, reading a 32-bit block from the slice into the buffer, and putting the 32-bit slice back into self-refresh. The 40-bit slice is then brought out of self-refresh and a 40-bit block from the slice is read into a buffer. Each block may comprise a portion of multiple words. For example, each 32-bit block may comprise 32-bit portions of multiple 72-bit words. In other embodiments, each block comprises a portion of a single word. The 40-bit slice is then put back into self-refresh in the example embodiment. The 32-bit and 40-bit slices are then combined into a 72-bit block by the controller **62** and ECC detection/correction is performed on each 72-bit word as it is read from the buffer and written into the non-volatile memory subsystem (e.g., flash).

In some embodiments, the entire data word may comprise more than two portions. For example, the entire data word may comprise three portions instead of two and transferring each data word further comprises storing a third portion of each data word in the buffer. In certain other embodiments, the data word may comprise more than three portions.

In certain embodiments, the data may be sliced in a restore operation as well as, or instead of, during a backup operation. For example, in one embodiment, the non-volatile memory elements **40** write each backed-up data word to the controller **62** which writes a first portion of the data word to the volatile memory subsystem **30** and then a second portion of the data word to the volatile memory **30**. In certain embodiments, slicing the volatile memory subsystem **30** during a restore operation may be performed in a manner generally inverse to slicing the volatile memory subsystem **30** during a backup operation.

The method **600** can advantageously provide significant power savings and can lead to other advantages. For example, in one embodiment where the volatile memory subsystem **30** comprises DRAM elements, only the slice of the DRAM which is currently being accessed (e.g., written to the buffer) during a backup is configured in full-operational mode. The slice or slices that are not being accessed may be put in self-refresh mode. Because DRAM in self-refresh mode uses significantly less power than DRAM in full-operational mode, the method **600** can allow significant power savings. In certain embodiments, each slice of the DRAM includes a separate self-refresh enable (e.g., CKe) signal which allows each slice to be accessed independently.

In addition, the connection between the DRAM elements and the controller **62** may be as large as the largest slice instead of as large as the data bus. In the example embodiment, the connection between the controller **62** and the DRAM may be 40 bits instead of 72 bits. As a result, pins on the controller **62** may be used for other purposes or a smaller controller may be used due to the relatively low number of pin-outs used to connect to the volatile memory subsystem **30**. In certain other embodiments, the full width of the data bus is connected between the volatile memory subsystem **30** and the controller **62** but only a portion of it is used during slicing operations. For example, in some embodiments, memory slicing is an optional mode.

Various embodiments of the present invention have been described above. Although this invention has been described with reference to these specific embodiments, the descriptions are intended to be illustrative of the invention and are not intended to be limiting. Various modifications and applica-

tions may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined in the appended claims.

What is claimed is:

1. A method for controlling a memory system operatively coupled to a host system, the memory system including a volatile memory subsystem and a non-volatile memory subsystem, the method comprising:

operating the volatile memory subsystem at a first clock frequency when the memory system is in a first mode of operation in which data is communicated between the volatile memory subsystem and the host system;

operating the non-volatile memory subsystem at a second clock frequency when the memory system is in a second mode of operation in which data is communicated between the volatile memory subsystem and the non-volatile memory subsystem; and

operating the volatile memory subsystem at a third clock frequency when the memory system is in the second mode of operation, the third clock frequency being less than the first clock frequency.

2. The method of claim 1, wherein the third clock frequency is substantially equal to the second clock frequency.

3. The method of claim 1, wherein the memory system is not powered by a battery when it is in the second mode of operation.

4. The method of claim 1, wherein the memory system switches from the first mode of operation to the second mode of operation in response to a trigger condition.

5. The method of claim 4, wherein the trigger condition comprises a power failure condition.

6. The method of claim 1, wherein the memory system further comprises a printed circuit board and the volatile memory subsystem and the non-volatile memory subsystem are located on the printed circuit board.

7. The method of claim 1, wherein data communicated between the volatile memory subsystem and the non-volatile memory subsystem is intermediate data from a host computer system computation.

8. The method of claim 1, wherein data communicated between the volatile memory subsystem and the non-volatile memory subsystem is backup data from a backup operation.

9. The method of claim 8, wherein the backup operation is conducted at repeating time intervals.

10. The method of claim 9, wherein the backup operation is initiated in response to a trigger event.

11. The method of claim 1, wherein the second mode of operation comprises a backup operation in which data is communicated from the volatile memory subsystem to the non-volatile memory subsystem.

12. The method of claim 1, wherein the second mode of operation comprises a restore operation in which data is communicated from the non-volatile memory subsystem to the volatile memory subsystem.

13. The method of claim 1, wherein one or more of the first, second or third clock frequencies is configurable by the memory system.

14. The method of claim 1, wherein one or more of the first, second or third clock frequencies is configurable by a user.

15. A memory system operatively coupled to a host system, the memory system comprising:

a volatile memory subsystem operable at a first clock frequency when the memory system is in a first mode of operation in which data is communicated between the volatile memory subsystem and the host system; and

a non-volatile memory subsystem operable at a second clock frequency when the memory system is in a second mode of operation in which data is communicated between the volatile memory subsystem and the non-volatile memory subsystem,

the volatile memory subsystem further being operable at a third clock frequency when the memory system is in the second mode of operation, the third clock frequency being less than the clock first frequency.

16. The memory system of claim 15, further comprising:

a controller configured to decouple the non-volatile memory subsystem from the volatile memory subsystem in the first mode of operation and to couple the non-volatile memory subsystem to the volatile memory subsystem in the second mode of operation.

17. The memory system of claim 15, further comprising a plurality of power supplies and a switch configured to selectively deliver power from the plurality of power supplies to the volatile memory subsystem and the non-volatile memory subsystem as a function of the mode of operation.

18. The memory system of claim 15, wherein the third clock frequency is substantially equal to the second clock frequency.

19. The memory system of claim 15, wherein the memory system is not powered by a battery when it is in the second mode of operation.

20. The memory system of claim 15, wherein the memory system switches from the first mode of operation to the second mode of operation in response to a trigger condition.

21. The memory system of claim 20, wherein the trigger condition comprises a power failure condition.

22. The memory system of claim 15, wherein the memory system further comprises a printed circuit board and the volatile memory subsystem and the non-volatile memory subsystem are located on the printed circuit board.

23. The memory system of claim 15, wherein data communicated between the volatile memory subsystem and the non-volatile memory subsystem is intermediate data from a host computer system computation.

24. The memory system of claim 15, wherein data communicated between the volatile memory subsystem and the non-volatile memory subsystem is backup data from a backup operation.

25. The memory system of claim 24, wherein the backup operation is conducted at repeating time intervals.

26. The memory system of claim 25, wherein the backup operation is initiated in response to a trigger event.

27. The memory system of claim 15, wherein the second mode of operation comprises a backup operation in which data is communicated from the volatile memory subsystem to the non-volatile memory subsystem.

28. The memory system of claim 15, wherein the second mode of operation comprises a restore operation in which data is communicated from the non-volatile memory subsystem to the volatile memory subsystem.

29. The memory system of claim 15, wherein one or more of the first, second or third clock frequencies is configurable by the memory system.

30. The memory system of claim 15, wherein one or more of the first, second or third clock frequencies is configurable by a user.

* * * * *

US008359501B1

(12) **United States Patent**
Lee et al.

(10) **Patent No.: US 8,359,501 B1**
(45) **Date of Patent: *Jan. 22, 2013**

(54) **MEMORY BOARD WITH SELF-TESTING CAPABILITY**

(75) Inventors: **Hyun Lee**, Ladera Ranch, CA (US); **Jayesh R. Bhakta**, Cerritos, CA (US); **Soonju Choi**, Irvine, CA (US)

(73) Assignee: **Netlist, Inc.**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/183,253**

(22) Filed: **Jul. 14, 2011**

Related U.S. Application Data

(63) Continuation of application No. 12/422,925, filed on Apr. 13, 2009, now Pat. No. 8,001,434.

(60) Provisional application No. 61/044,801, filed on Apr. 14, 2008, provisional application No. 61/044,825, filed on Apr. 14, 2008, provisional application No. 61/044,839, filed on Apr. 14, 2008.

(51) **Int. Cl.**
G11C 29/00 (2006.01)

(52) **U.S. Cl.** **714/719**; 714/718

(58) **Field of Classification Search** 714/718, 714/719, 736, 733, 734, 738, 819; 365/201
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,660,675	A	5/1972	Andrews, Jr.	
3,757,235	A	9/1973	McCormick et al.	
4,305,091	A	12/1981	Cooper	
4,586,168	A	4/1986	Adlhoch et al.	
4,701,845	A	10/1987	Andreasen et al.	
4,752,741	A	6/1988	Kim et al.	
4,782,487	A	11/1988	Smelser	
4,837,743	A	6/1989	Chiu et al.	
4,885,799	A	12/1989	Van Horn	
4,903,266	A	2/1990	Hack	
4,910,597	A	3/1990	Harada et al.	
4,942,556	A	7/1990	Sasaki et al.	
4,958,346	A *	9/1990	Fujisaki	714/720
4,987,321	A	1/1991	Toohey	
5,033,048	A	7/1991	Pierce et al.	
5,051,997	A	9/1991	Sakashita et al.	
5,138,619	A	8/1992	Fasang et al.	
5,173,906	A	12/1992	Dreibelbis et al.	

(Continued)

OTHER PUBLICATIONS

Der-Chang et al. "*A parallel built-in self-diagnostic method for embedded memoryarrays*", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Apr. 2002, vol. 21, Issue 4, pp. 449-465.

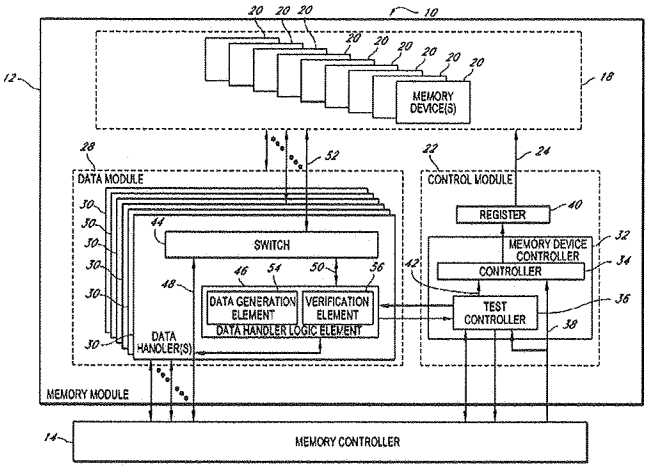
(Continued)

Primary Examiner — Phung M Chung

(57) **ABSTRACT**

A self-testing memory module includes a printed circuit board configured to be operatively coupled to a memory controller of a computer system and includes a plurality of memory devices on the printed circuit board, each memory device of the plurality of memory devices comprising data, address, and control ports. The memory module also includes a control module configured to generate address and control signals for testing the memory devices. The memory module includes a data module comprising a plurality of data handlers. Each data handler is operable independently from each of the other data handlers of the plurality of data handlers. Each data handler is operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and is configured to generate data for writing to the corresponding plurality of data ports.

20 Claims, 5 Drawing Sheets



Exh C

U.S. PATENT DOCUMENTS

5,222,066	A	6/1993	Grula et al.	
5,241,503	A	8/1993	Cheng	
5,304,856	A	4/1994	Rainal	
5,337,254	A	8/1994	Knee et al.	
5,359,235	A	10/1994	Coyle et al.	
5,394,037	A	2/1995	Josephson et al.	
5,430,335	A	7/1995	Tanoi	
5,525,917	A	6/1996	Wong et al.	
5,841,296	A	11/1998	Churcher et al.	
5,914,543	A	6/1999	Scherpenberg et al.	
6,000,048	A	12/1999	Krishna et al.	
6,044,481	A	3/2000	Kornachuk et al.	
6,070,217	A	5/2000	Connolly et al.	
6,169,696	B1	1/2001	Bissey	
6,194,959	B1	2/2001	Kamoshida et al.	
6,216,240	B1	4/2001	Won et al.	
6,467,056	B1	10/2002	Satou et al.	
6,560,740	B1	5/2003	Zuraski, Jr. et al.	
6,614,702	B2	9/2003	Lee	
6,681,358	B1 *	1/2004	Karimi et al.	714/733
6,721,150	B1	4/2004	Guerrero, Jr.	
6,812,869	B1	11/2004	Rahman et al.	
6,829,728	B2	12/2004	Cheng et al.	
6,910,162	B2 *	6/2005	Co et al.	714/718
6,918,072	B2	7/2005	Cowles et al.	
6,928,024	B2	8/2005	Pfeiffer et al.	
6,928,593	B1	8/2005	Halbert et al.	
6,930,509	B2	8/2005	Banik	
6,934,900	B1	8/2005	Cheng et al.	
7,036,064	B1	4/2006	Kebichi et al.	
7,053,470	B1	5/2006	Sellers et al.	
7,062,696	B2	6/2006	Barry et al.	
7,184,915	B2	2/2007	Hansquine et al.	
7,190,210	B2	3/2007	Azrai et al.	
7,203,873	B1	4/2007	Adams	
7,253,652	B2	8/2007	Azimi et al.	
7,284,166	B2	10/2007	Zappa et al.	
7,774,667	B2 *	8/2010	Saito et al.	714/733
2001/0048342	A1	12/2001	Yoshida et al.	
2002/0000847	A1	1/2002	Taguchi	
2002/0131535	A1	9/2002	Huber	
2002/0140523	A1	10/2002	Park et al.	
2003/0098742	A1	5/2003	Nakagawa et al.	
2003/0126346	A1	7/2003	Kuo	
2003/0197797	A1	10/2003	Segura	
2003/0218491	A1	11/2003	Nagasue	
2004/0155702	A1	8/2004	Danielsson	
2004/0199843	A1	10/2004	Hansquine et al.	
2005/0093620	A1	5/2005	Ho et al.	
2005/0127989	A1	6/2005	Miyagi	
2005/0289423	A1	12/2005	Yabuta	
2006/0082383	A1	4/2006	Choi	
2006/0107156	A1	5/2006	Lee et al.	
2006/0140015	A1	6/2006	Kasamsetty	
2006/0144015	A1	7/2006	Cash et al.	
2006/0147217	A1	7/2006	Hahin et al.	
2006/0192653	A1	8/2006	Atkinson et al.	
2006/0242458	A1	10/2006	Feldman et al.	
2006/0262586	A1	11/2006	Solomon et al.	
2006/0271748	A1	11/2006	Jain et al.	
2007/0030814	A1	2/2007	Shin et al.	
2007/0058471	A1	3/2007	Rajan et al.	
2007/0079199	A1	4/2007	Chorn et al.	
2007/0109707	A1	5/2007	Honda	
2007/0152743	A1	7/2007	Keeth et al.	
2007/0204075	A1	8/2007	Rajan et al.	
2007/0223296	A1	9/2007	Miller et al.	

OTHER PUBLICATIONS

Bayard "On the LTI properties of adaptive feed forward systems with tapdelay-line regressors", IEEE Transactions on Signal Processing, May 1999, vol. 47, Issue 5, pp. 1288-1296.

Mutoh et al. "EMI Noise controlling methods suitable for electric vehicle drive systems", Industrial Electronics Society, 2004, IECON 2004, Nov. 2-6, 2004, vol. 1, pp. 963-968.

Sekiguchi et al "Low-noise, high-speed datat transmission using a ringing-canceling output buffer", IEEE Journal of Solid-State Circuits, Dec. 1995, vol. 30, Issue 12, pp. 1569-1574.

* cited by examiner

FIG. 1

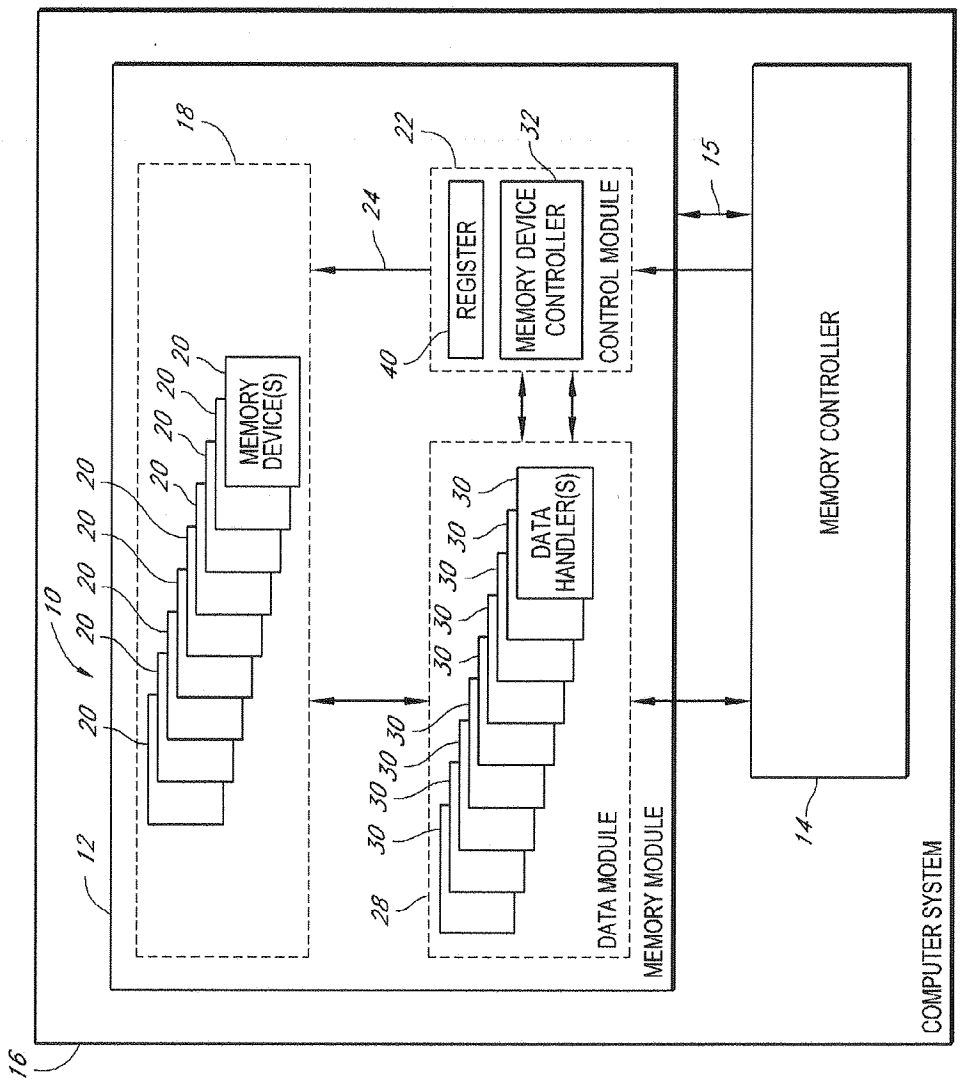
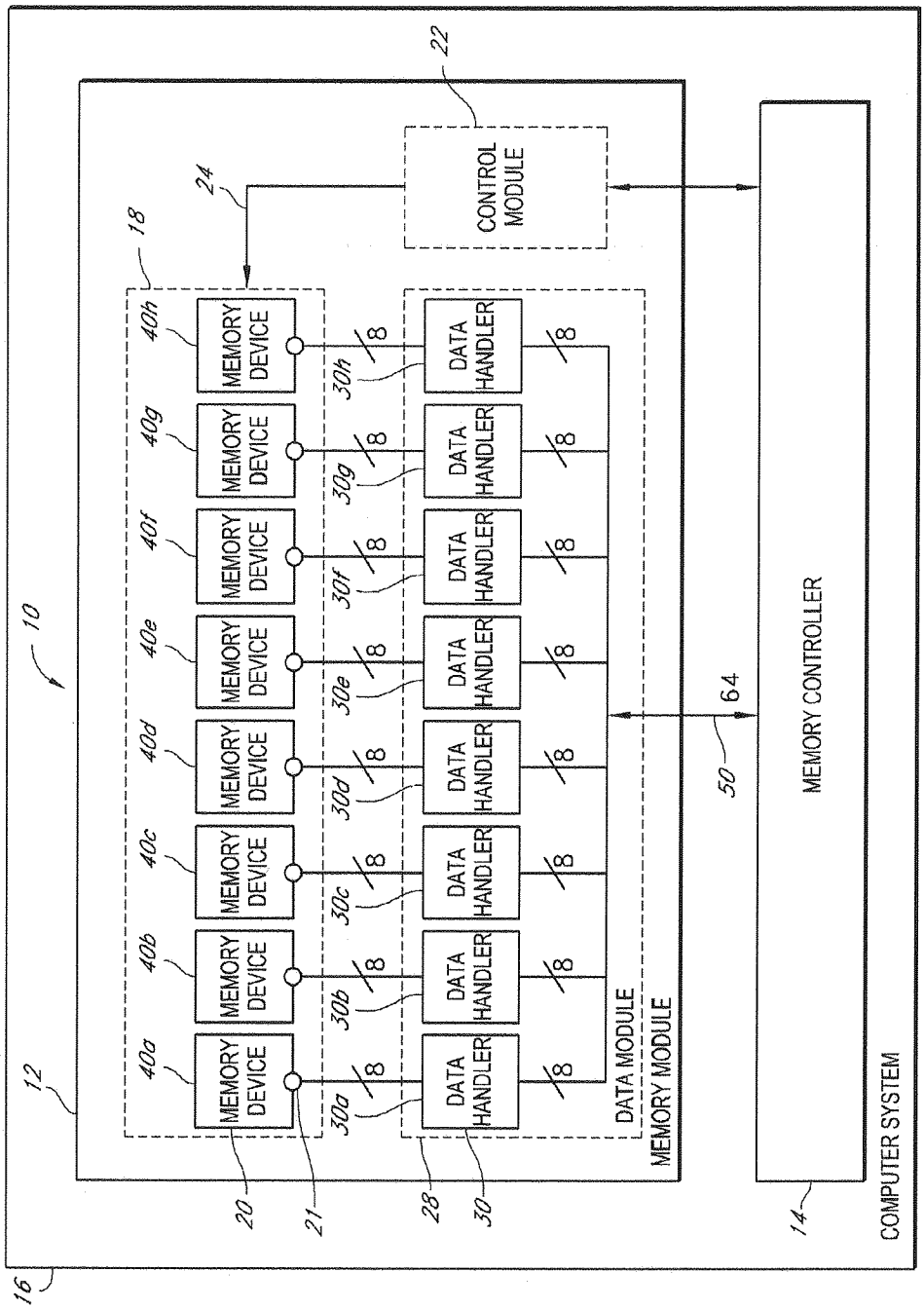

COMPUTER SYSTEM
MEMORY CONTROLLER
MEMORY MODULE
DATA MODULE
CONTROL MODULE
REGISTER
MEMORY DEVICE CONTROLLER
DATA HANDLER(S)
MEMORY DEVICE(S)
10
12
14
15
16
18
20
22
24
28
30
32
40

FIG. 2


10
12
16
18
20
21
22
24
28
30
50
64
14
CONTROL MODULE
MEMORY CONTROLLER
COMPUTER SYSTEM
DATA MODULE
MEMORY MODULE
MEMORY DEVICE
DATA HANDLER
40a
40b
40c
40d
40e
40f
40g
40h
30a
30b
30c
30d
30e
30f
30g
30h
8

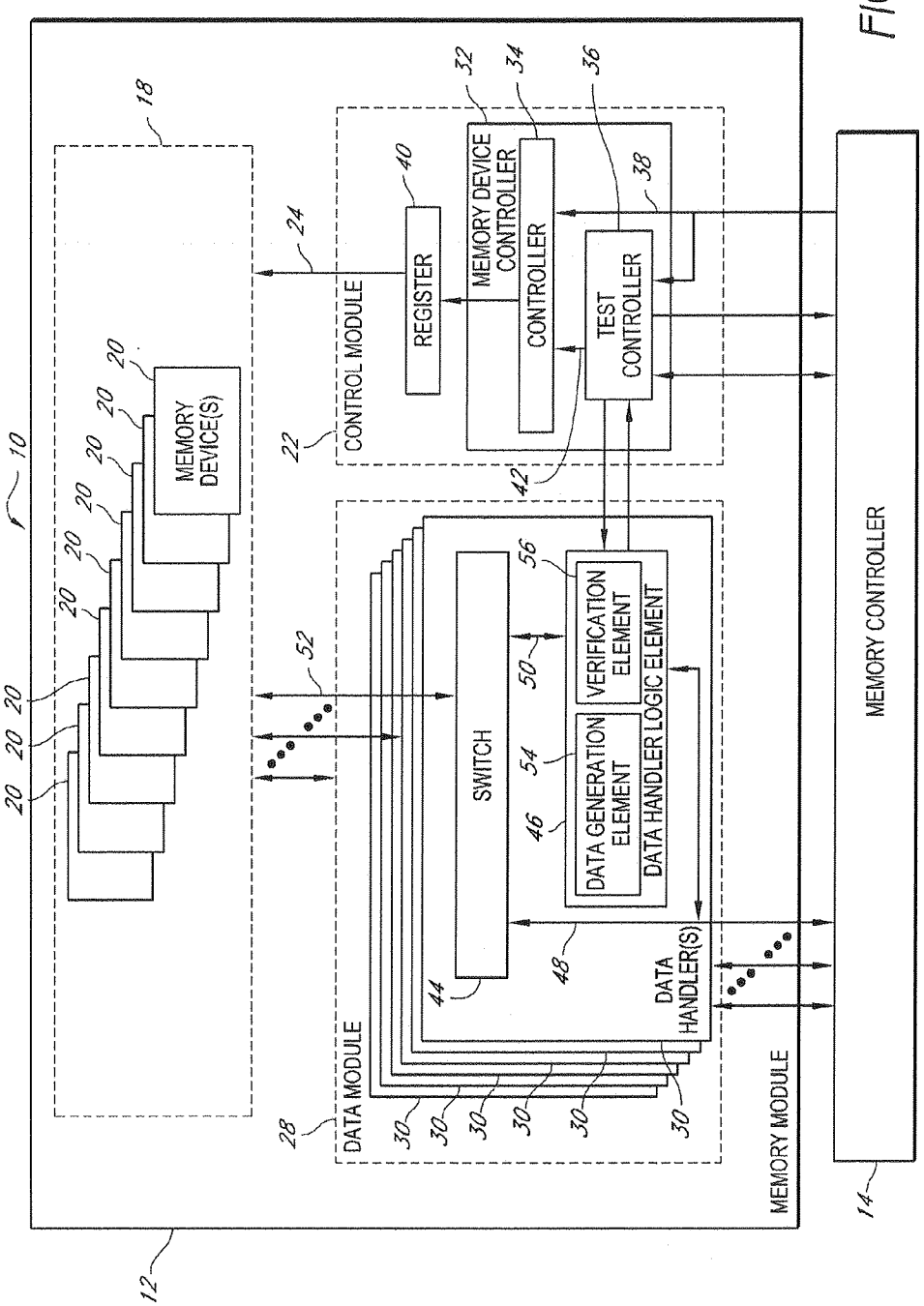

10
12
14
18
20
MEMORY DEVICE(S)
22
CONTROL MODULE
24
REGISTER
32
34
36
38
40
MEMORY DEVICE CONTROLLER
CONTROLLER
TEST CONTROLLER
42
MEMORY CONTROLLER
MEMORY MODULE
28
DATA MODULE
30
44
SWITCH
46
48
DATA HANDLER(S)
50
52
54
56
DATA GENERATION ELEMENT
VERIFICATION ELEMENT
DATA HANDLER LOGIC ELEMENT

FIG. 3

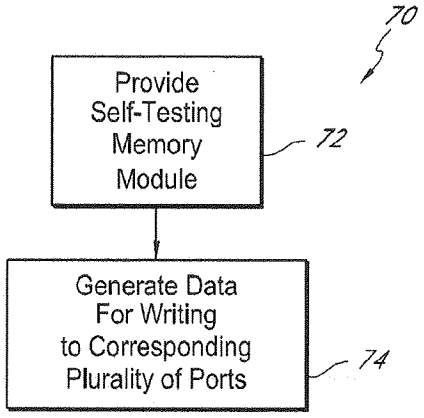

70
Provide Self-Testing Memory Module
72
Generate Data For Writing to Corresponding Plurality of Ports
74

FIG. 4

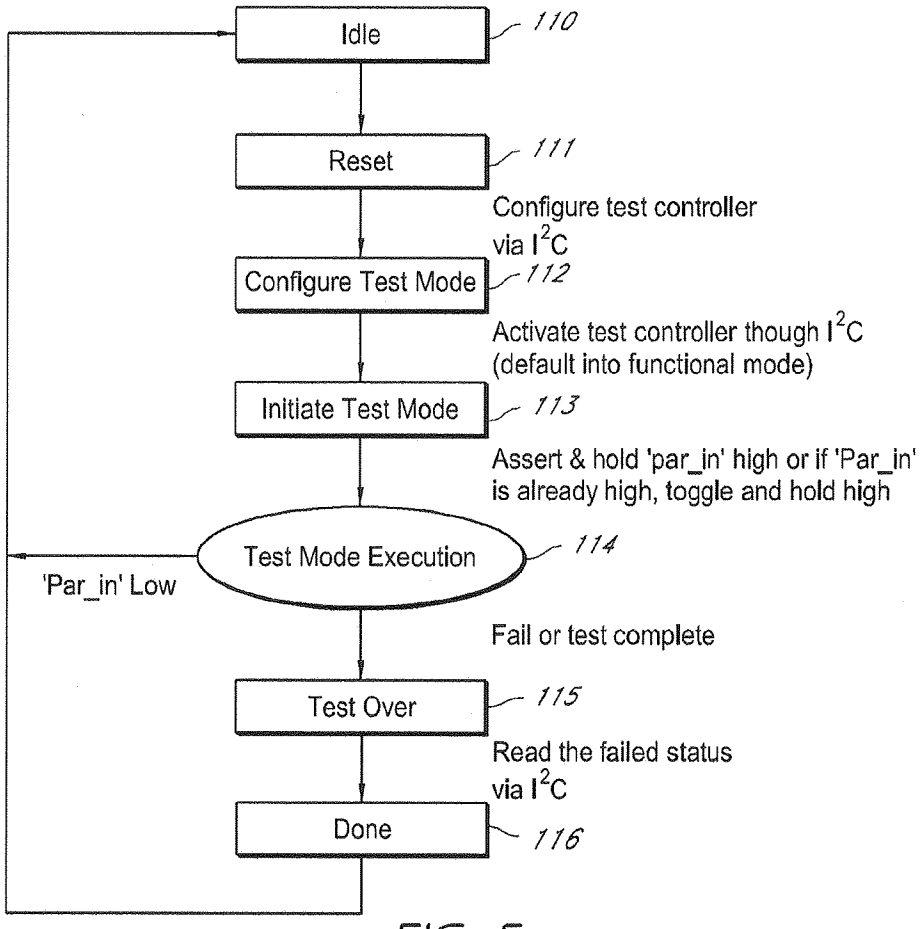

Idle
110
Reset
111
Configure test controller via I2C
Configure Test Mode
112
Activate test controller though I2C (default into functional mode)
Initiate Test Mode
113
Assert & hold 'par_in' high or if 'Par_in' is already high, toggle and hold high
Test Mode Execution
114
'Par_in' Low
Fail or test complete
Test Over
115
Read the failed status via I2C
Done
116

FIG. 5

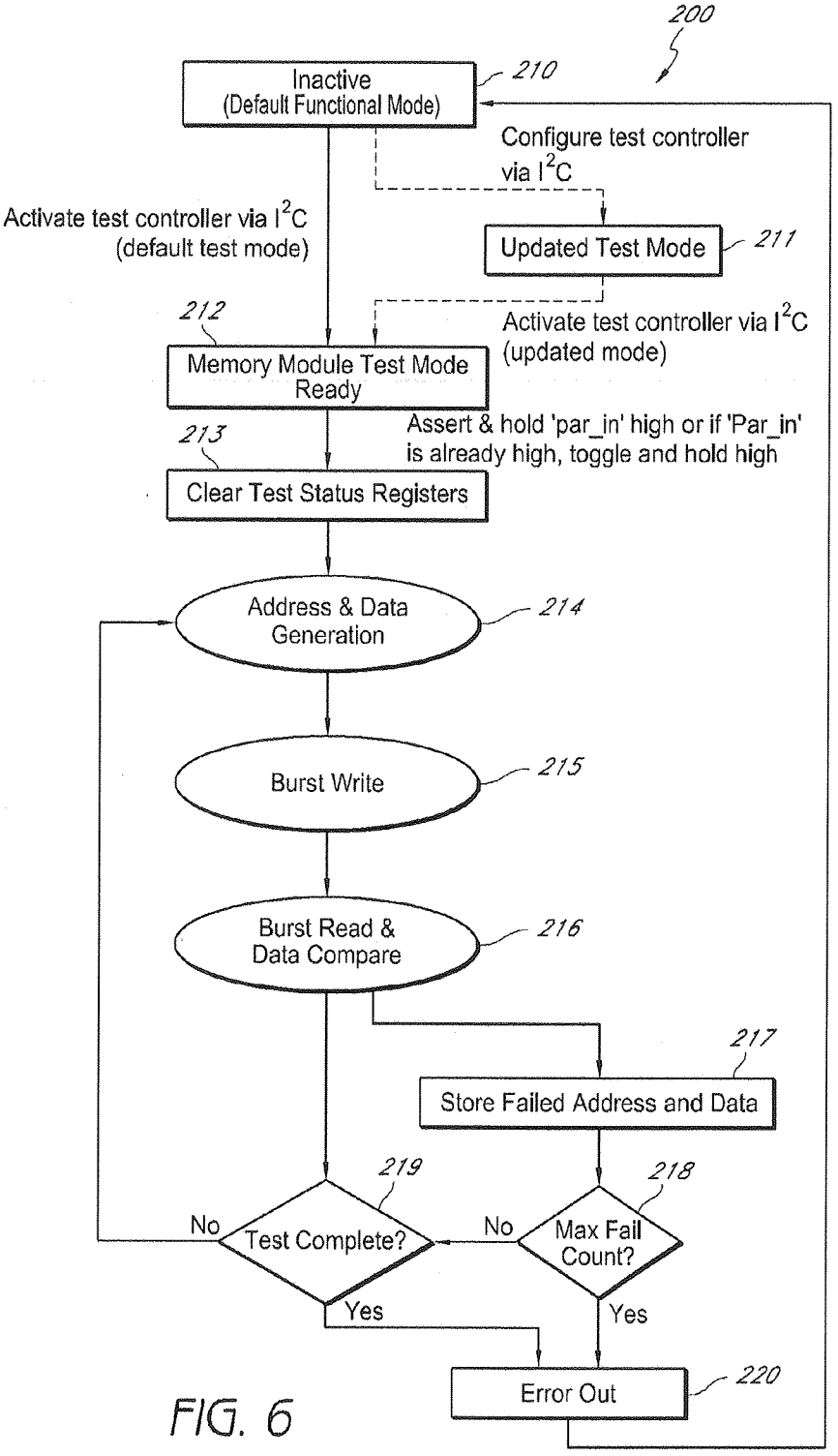

200
Inactive
(Default Functional Mode)
210
Configure test controller via I2C
Activate test controller via I2C
(default test mode)
Updated Test Mode
211
212
Activate test controller via I2C
(updated mode)
Memory Module Test Mode Ready
Assert & hold 'par_in' high or if 'Par_in' is already high, toggle and hold high
213
Clear Test Status Registers
Address & Data Generation
214
Burst Write
215
Burst Read & Data Compare
216
217
Store Failed Address and Data
219
218
No
Test Complete?
No
Max Fail Count?
Yes
Yes
Error Out
220

FIG. 6

MEMORY BOARD WITH SELF-TESTING CAPABILITY

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is a continuation of U.S. patent application Ser. No. 12/422,925, filed Apr. 13, 2009, now U.S. Pat. No. 8,001,434, which claims the benefit of priority from US. Provisional Application No. 61/044,801, filed Apr. 14, 2008, US. Provisional Application No. 61/044,825, filed Apr. 14, 2008, and US. Provisional Application No. 61/044,839, filed Apr. 14, 2008. Each of the foregoing applications is incorporated in their entirety by reference herein. This application is related to U.S. patent application Ser. No. 12/422,912, filed on Apr. 13, 2009 and entitled "Self-Adjusting Damper", which is still pending at the United States Patent and Trademark Office, and to U.S. application Ser. No. 12/422,853, filed on Apr. 13, 2009, and entitled "Circuit Providing Load Isolation and Noise Reduction", now U.S. Pat. No. 8,154,901, both of which are incorporated in their entirety by reference herein.

BACKGROUND

1. Field

The present invention relates to self-testing electronic modules and, more particularly, to self-testing electronic memory modules.

2. Description of the Related Art

The failure of memory components in an electronic system may result in the loss of valid data. Therefore, it is important to ensure proper memory operation in an electronic system. Memory integrated circuits ("memory chips") often go through a series of tests at various stages of system manufacture. Once memory chips are deployed in a system, they also generally go through a system level memory test each time the system is booted. In addition, memory chips may undergo a parity checking process during normal system operation.

There are typically at least three test phases which memories undergo during system manufacture. Each phase generally tests for memory defects and for the correct operation of the input/output interface. The first test phase is typically conducted by the memory chip manufacturer and generally involves checking for bit failures, correct memory access speed, etc. The second test phase is typically done by memory module manufacturers and generally involves testing the signal quality, the noise susceptibility, and the operational speed of the memory module as a single unit. The second test phase may also include checking for bit failures in individual memory chips. The third phase is usually carried out by the system manufacturer. During the third phase, the interaction of the memory subsystem with other components in the system is tested. During the third phase, the individual memory module operation is also tested again and the memory array is checked for defects. Because of the significant amount of testing that memories undergo during the manufacturing process, there is generally substantial test cost and test time associated with ensuring the proper memory operation. This test cost and test time translate into an increase in system cost and a decrease in system performance.

There are a number of memory test methodologies that employ either external test hardware, embedded self-test logic ("MBIST"), or both. However, the usefulness of these test methodologies is limited due to the high cost and other limitations associated with them. For instance, external test hardware such as automatic test equipment ("ATE") is very expensive. Moreover, the development time and cost associated with implementing MBIST is relatively high. These costs and limitations are especially significant when testing dynamic random access memory ("DRAM"). For example, technological developments, such as increases in DRAM speed, may require manufacturers to upgrade ATE machines relatively frequently. In addition, MBIST in DRAM chips generally cannot be fully utilized for system level testing of memory boards.

Because of the increasing cost, complexity, and time involved with fully testing DRAM chips, DRAM manufacturers often provide "effectively tested" ("ETT") DRAM chips to memory module manufacturers at a lower price rather than providing fully tested DRAM chips. Memory module manufacturers often prefer the ETT DRAM chips mainly due to their greater availability. Memory module manufacturers who receive ETT DRAM chips then have to assume a part of the responsibility of validating the DRAM chips, adding to the complexity of the memory module test process.

SUMMARY

In certain embodiments, a self-testing memory module is provided comprising a printed circuit board configured to be operatively coupled to a memory controller of a computer system. The memory module may further comprise a plurality of memory devices on the printed circuit board. Each memory device of the plurality of memory devices comprises data, address, and control ports. In certain embodiments, the memory module includes a control module configured to generate address and control signals for testing the memory devices and a data module comprising a plurality of data handlers. Each data handler in certain embodiments is operable independently from each of the other data handlers of the plurality of data handlers. Each data handler may be operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and configured to generate data for writing to the corresponding plurality of data ports.

A self-testing memory module is provided in certain embodiments comprising a printed circuit board configured to be operatively coupled to a memory controller of a computer system. The memory module may further comprise a plurality of memory devices on the printed circuit board. Each memory device of the plurality of memory devices comprises data, address, and control ports in some embodiments. The memory module of certain embodiments includes a control module configured to generate address and control signals for testing the memory devices. In certain embodiments, the memory module comprises a data module comprising at least one data handler operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices and configured to generate cyclic data for writing to the corresponding plurality of data ports.

In certain embodiments, a method of self-testing a memory module includes providing a self-testing memory module. The memory module comprises a printed circuit board configured to be operatively coupled to a memory controller of a computer system. The memory module further comprises a plurality of memory devices on the printed circuit board. Each memory device of the plurality of memory devices comprising data, address, and control ports. In certain embodiments, the memory module comprises a control module configured to generate address and control signals for testing the memory devices. The memory module may further comprise a data module comprising a plurality of data han-

dlers. Each data handler may be operable independently from each of the other data handlers of the plurality of data handlers and operatively coupled to a corresponding plurality of the data ports. The method of certain embodiments further includes generating, by each of the data handlers, data for writing to the corresponding plurality of data ports.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of an example self-testing memory module in accordance with certain embodiments described herein.

FIG. **2** is a block diagram of an example self-testing memory module including eight memory devices and eight data handlers in accordance with certain embodiments described herein.

FIG. **3** is a block diagram of an example data handler module and an example control module in accordance with certain embodiments described herein.

FIG. **4** is a flowchart of an example method of self-testing a memory module in accordance with certain embodiments described herein.

FIG. **5** is a flow diagram illustrating the self-testing operation of an example memory module in accordance with certain embodiments described herein.

FIG. **6** is a flow diagram illustrating the self-testing operation of an example memory module in accordance with certain embodiments described herein.

DETAILED DESCRIPTION

Certain embodiments described herein present a self-testing electronic system, such as for example, a self-testing electronic memory module. Some embodiments described herein present a self-testing memory module that is populated with ETT DRAM chips. Certain embodiments described herein present a self-testing registered dual in-line memory module ("RDIMM"). Some embodiments described herein present a self-testing RDIMM that does not require any additional pins other than the pins on the standard JEDEC RDIMM connector. For example, the memory module may utilize the address and control signals (e.g., address and control signals generated by the memory module to test the memory module) along with a test signal to enable and execute a self testing function.

Certain embodiments described herein present a self-testing memory module that includes a control module and a data module which can generate memory addresses and data according to the JEDEC standard memory protocol.

Some embodiments described herein presents a self-testing memory module that can be configured through an I^2C interface and that allows test results to be read out through the I^2C interface. Certain embodiments described herein present a self-testing memory module that allows a test function to be configured, controlled, and/or executed without substantial system memory controller involvement. Some embodiments present a self-testing memory module that can be tested without any external test equipment. For example, certain embodiments present a self-testing memory module that can be tested without any system driven test procedure. Various embodiments described herein present a self-testing memory module that can be self-tested at a target system speed. Certain embodiments described herein present a self-testing memory module that can generate data (DQ) and data-strobe (DQS) signals with wave characteristics that resemble the wave characteristics of DQ and DQS signals from a system memory controller.

MBIST is commonly used to test memories embedded in application specific integrated circuits ("ASICs") or system on chip integrated circuits ("SoCs") such as, for example, advanced memory buffers ("AMBs"). The MBIST implementation usually includes three distinct functional blocks: the address/control generator, the data generator/checker (sometimes referred to as a signature analyzer), and the test interface controller/register ("JTAG"). The test instructions and test patterns are generally loaded to the address/control generator and the data generator/checker through the JTAG interface. The test results are generally read out through the JTAG interface.

In memory module applications, there are obstacles associated with using MBIST. One of these obstacles is the large number of signals used to interface the three MBIST functional blocks. The inter-block timing constraints can present another obstacle. In addition, the ability to control the test is limited. For example, synchronizing the address/or control signals with the data signal can be difficult. The relatively large amount of information to be gathered and stored while testing the memory presents yet another obstacle.

Unlike on an ASIC, in which the three MBIST functional blocks are in a single chip, the MBIST functional blocks on a memory module generally would be segregated into multiple chips on the memory module due to physical and electrical limitations and requirements. This makes implementing MBIST on a memory module difficult because, while there is virtually no limit on the number of available interface signals among the three MBIST functional blocks in an ASIC, the memory module can support only a limited amount of interface signals between memory chips. In addition, because the MBIST functional blocks are spread out to multiple chips on a memory module, the inter-block signal delay is generally much longer on a memory module than on an ASIC. It is also generally not feasible to implement cross-checking logic that operates the three MBIST functional blocks in lock-operational block. This is due to the limitation on the number of interface signals and to the relatively long inter-block signal delay time on a memory module versus on an ASIC.

In some cases such as where a memory module includes an AMB, the self-test logic (MBIST) implemented in the AMB includes command and address generation logic in addition to a data generator and checker. Each of these functional blocks may be implemented on a single physical AMB device (e.g., a single integrated circuit package). However, because all of the self-test command, address and data signals are combined in one physical area of the memory module, a memory module (e.g., DIMM) level routing problem can occur, making it difficult to route the self-test signals on the memory module and resulting in performance degradation and/or implementation difficulty. In addition, the data width of self-test logic of the memory module will be limited to the data width of the AMB (e.g., to the data width of the self-test logic implemented on the AMB, to the number of available ports on the AMB, etc.). As such, implementing memory module test logic on an AMB is not flexible (e.g., to changes in the data width of the memory module) and generally supports only memory modules having certain predetermined, fixed data widths.

Finally, in most cases, since an ASIC cannot be repaired, the ASIC MBIST is generally only capable of detecting and reporting the pass/fail status of memory tests. For memory module test results, on the other hand, it is generally advantageous to include both addresses of the memory locations where failures occur and the data patterns that were read back from the failed memory locations. This type of reporting can

help to facilitate the repair of the memory module by, for example, allowing for the identification and replacement of failed components.

FIG. **1** is a block diagram of an example self-testing memory module **10** in accordance with certain embodiments described herein. The memory module **10** includes a printed circuit board **12** configured to be operatively coupled to a memory controller **14** of a computer system **16**. The memory module **10** further includes a plurality of memory devices **18** on the printed circuit board (PCB) **12**, each memory device **20** of the plurality of memory devices **18** comprising data, address, and control ports. The memory module **10** comprises a control module **22** configured to generate address and control signals **24** for testing the memory devices **18**. The memory module **10** also includes a data module **28** comprising a plurality of data handlers **30**. Each data handler **30** is operable independently from each of the other data handlers **30** of the plurality of data handlers **28** and is operatively coupled to a corresponding plurality of the data ports of one or more of the plurality of memory devices **18**. For example, each of the data handlers **30** may be operatively coupled to (e.g., logically and/or electrically coupled to) the corresponding plurality of data ports. Each data handler **30** is further configured to generate data for writing to the corresponding plurality of data ports. The memory module **10** may further include an I^2C interface **15** in certain embodiments.

As described more fully below, in certain embodiments the data module **28** generates test data patterns to write to the plurality of memory devices **18** of the memory module **12** and checks the data patterns read or received back from the plurality of memory devices **18** for agreement with corresponding data patterns that are expected to be read back from the plurality of memory devices **18**. For example, in one embodiment, the data module **28** generates cyclic patterns to write to the plurality of memory devices **18**. In some embodiments, the data module **18** also isolates the data path from the system board of the computer system **16** to the plurality of memory devices **18** while the memory module **10** is not accessed by the computer system **16**. For example, the data module **28** may isolate the data path from the system board to the plurality of memory devices **18** when the memory module **10** is in a self-testing mode. The control module **22** may include, for example, a dual input register (e.g., the memory device controller **32** described more fully below) for registering address and control signals coming from either self-testing logic (e.g., from the test controller **36** described more fully below) or from the memory controller **14** on the system board. In some embodiments, during testing, the control module **22** generates address and control signals **24** associated with memory locations to be tested and the data module **28** generates corresponding test data patterns and provides them to the appropriate memory devices **20**. For example, the data module **28** may receive a write command from the control module **22** and provide data to be written to certain locations in the memory devices **20** during a write operation. The data module **28** may then receive a read command to read back the data from those locations and check the read data for agreement with the expected data. If there is a mismatch between the read data and the expected data, the data module **28** may, for example, store the failure information (e.g., the failed data word) and inform the control module **22** about the failure. The control module may save the address of the memory location where the failure occurred.

In certain embodiments, the memory module **10** is configured to be operated in a test mode in which the control module **22** selectively inputs the address and control signals to the address and control ports of the plurality of memory devices **18**. Moreover, in the test mode, each of the data handlers **30** write the data generated by the data handler **30** to the corresponding plurality of data ports by selectively inputting data signals to the data ports of the plurality of memory devices **18**. The data module **28** and/or the control module **22** of certain embodiments are configured to test the plurality of memory devices **18** at the normal operating speed of the memory devices **20**. For example, the data module **28** and/or the control module **22** are configured to provide memory signals (e.g., data, address and control signals) according the operating specification of the memory devices **20**. In some embodiments, the control module **22** and the data module **28** produce memory addresses, control and/or data signals according to the JEDEC standard memory protocol. In some embodiments, for example, the control module **22** and the data module **28** generate the memory interface signals with proper edge relationships based on the JEDEC standard. In certain embodiments, the test speed, for example, may be defined by the speed of the clock (e.g., the system clock). The address sequences and/or the data patterns of certain embodiments may be programmable either through the I^2C interface or they may be defaulted to pre-defined values.

In certain embodiments, data module **28** is configured to generate data signals with programmable slew rates and/or with variable peak values. In one embodiment, for example, the data module **28** is also able to generate data (DQ) and data-strobe (DQS) signals with programmable slew rates and programmable peak values so that the characteristics of the signals generated by the data module **28** generally correspond to the characteristics DQ and DQS signals generated by the system memory controller **14**.

In some embodiments, the data module **28** and/or the control module **22** are configured to test the plurality of memory devices **18** under non-normal conditions. For example, the data module **28** and/or control module **22** may be configured to provide signals having frequencies which are higher or lower than the normal operating frequencies of the memory devices **20**.

In certain embodiments, the memory module **10** has a memory capacity of 512-MB, 1-GB, 2-GB, 4-GB, or 8-GB. Other memory capacities are also compatible with certain embodiments described herein. In addition, memory modules **100** having widths of 4 bytes, 8 bytes, 16 bytes, 32 bytes, or 32 bits, 64 bits, 128 bits, 256 bits, as well as other widths (in bytes or in bits), are compatible with embodiments described herein. In certain embodiments, the PCB **12** has an industry-standard form factor. For example, the PCB **12** can have a low profile (LP) form factor with a height of 30 millimeters and a width of 133.35 millimeters. In certain other embodiments, the PCB **12** has a very high profile (VHP) form factor with a height of 50 millimeters or more. In certain other embodiments, the PCB **12** has a very low profile (VLP) form factor with a height of 18.3 millimeters. Other form factors including, but not limited to, small-outline (SO-DIMM), unbuffered (UDIMM), registered (RDIMM), fully-buffered (FBDIMM), mini-DIMM, mini-RDIMM, VLP mini-DIMM, micro-DIMM, and SRAM DIMM are also compatible with certain embodiments described herein. For example, in other embodiments, certain non-DIMM form factors are possible such as, for example, single in-line memory module (SIMM), multi-media card (MMC), and small computer system interface (SCSI).

In certain embodiments, the plurality of memory devices **18** of the memory module **10** may be arranged as ranks, each rank of memory generally having a bit width. In certain embodiments, each rank may comprise an independent set of memory devices **20** of the plurality of memory devices **18** that

can be accessed by the memory controller **14** to access the full bit-width of the memory bus of the memory module **10**. For example, a memory module **10** in which each rank of the memory module is 64 bits wide is described as having an "x 64" organization. Similarly, a memory module **10** having 72-bit-wide ranks is described as having an "x 72" organization. The number of memory devices **20** and corresponding memory capacity of a memory module **10** can be increased by increasing the number of memory devices **20** per rank or by increasing the number of ranks. For example, a memory module with four ranks with each rank having N 512-MB memory devices **20** has double the memory capacity of a memory module with two ranks with each rank having N 512-MB memory devices **20** and four times the memory capacity of a memory module with one rank with each rank having N 512-MB memory devices **20**. During operation, the ranks of a memory module **10** may be selected or activated by control signals that are received from a component of the system (e.g., a system memory controller **14** or a local memory controller of the memory module **10**). Examples of such control signals include, but are not limited to, rank-select signals, also called chip-select signals. In certain other embodiments, the memory module **10** comprises only one rank of memory devices **20**.

As discussed, the PCB **12** may include at least one connector (not shown) configured to operatively couple the memory module **10** to the memory controller **14** of the computer system **16**. The computer system **16** may include a host computer system. For example, the memory module is electrically coupled, logically coupled, or both, with the memory controller **14**. Examples of host computer systems **108** include, but are not limited to, blade servers, 1U servers, personal computers (PCs), data storage systems and other applications in which space is constrained or limited. The memory controller **14** may comprise a disk controller of the computer system **16**, for example. The memory controller **14** may be mounted on a system board of the host computer **16**. The connector can comprise a plurality of edge connections which fit into a corresponding slot connector of the host system **16**. The connector of certain embodiments provides a conduit for power voltage as well as data, address, and control signals between the memory module **10** and the host system **16**. For example, the connector can comprise a standard DDR2, DDR3, and other future generation edge connectors. Additionally, in certain embodiments, more than one memory module **10** is coupled to the host system **16**.

The plurality of memory devices **18** on the PCB **12** may include one or more volatile memory components. For example, the plurality of memory devices **18** of certain embodiments comprises two or more dynamic random-access memory (DRAM) elements **20**. Types of DRAM devices **20** compatible with certain embodiments described herein include, but are not limited to, DDR, DDR2, DDR3, and synchronous DRAM (SDRAM). The memory devices **18** may comprise other types of memory elements such as static random-access memory (SRAM). In addition, volatile memory devices **20** having bit widths of 4, 8, 16, 32, as well as other bit widths, are compatible with certain embodiments described herein. Memory devices **20** compatible with certain embodiments described herein have packaging which include, but are not limited to, thin small-outline package (TSOP), ball-grid-array (BGA), fine-pitch BGA (FBGA), micro-BGA (μBGA), mini-BGA (mBGA), and chip-scale packaging (CSP). The plurality of memory devices **18** may further include one or more non-volatile memory devices **20**, such as, for example, flash memories. The plurality of memory devices **18** of certain embodiments may include both volatile and non-volatile memory devices **20**. For example, the plurality of memory devices **18** may include one or more of DRAM, SRAM, and/or flash memory devices in some embodiments.

Each data handler **30** is operable independently from each of the other data handlers **30** of the plurality of data handlers **28**. For example, each data handler **30** is configured to write to and/or read from the corresponding plurality of data ports of one or more of the memory devices **20** without being in communication any of the other data handlers **30** or other data ports of the memory devices **20**. As such, each data handler **30** can be used to generally independently test a portion of the memory space of the memory module **10**. For example, each data handler **30** may be used to independently test one memory device **20** of the memory module **10**. In such a configuration, the corresponding plurality of data ports of each data handler **30** may comprise each data port of the corresponding memory device **20**. In other embodiments, each data handler **30** may be used to test a segment of one memory device **20**, more than one memory device **20**, segments more than one memory device **20**, or any combination or sub-combination thereof. Because each of the data handlers **30** is operable independently of each of the other data handlers **30**, the data handlers **30** are generally modular. As such, modifications in the configuration of the memory module **10** (e.g., changes in the bit-width of the memory bus, changes in the number of memory devices **20**, etc.) may be less complicated to accommodate than in other types of self-testing memory modules **10**. For example, where a new memory device **20** or set of memory devices **20** is added to the memory module **10**, the change may be generally accommodated by adding a corresponding data handler **30**. The change may be accommodated without having to implement a major reorganization of the memory module **10** or the self-testing logic of the memory module **10**, for example.

Each data handler **30** is further configured to generate data for writing to the corresponding plurality of data ports. FIG. **2** is a block diagram of an example self-testing memory module **10** including eight memory devices **20** (e.g., memory devices **40***a*-**40***h*) and a data module **28** comprising eight data handlers **30** (e.g., data handlers **30***a*-**30***h*) in accordance with certain embodiments described herein. Each of the memory devices **20** includes an eight bit output data word and eight corresponding data ports. In addition, the system memory bus **50** between the memory controller **14** and the example memory module **10** is 64 bits wide and each of the data handlers **30** receives an eight bit segment of the system memory bus **50**. Each of the data handlers **30** is operatively coupled to a corresponding plurality of data ports **21** of a corresponding one of the memory devices **20**. As such, the data handlers **30** may be operatively coupled (e.g., electrically and/or logically coupled or connected) to the eight data ports **21** of one of the corresponding memory devices **20**. For example, the data handler **30***a* may be operatively coupled to the eight data ports **21** of the memory device **40***a* of FIG. **2**.

The configuration shown in FIG. **2** is for the purposes of illustration and is not intended to be limiting. For example, while the example memory module **10** of FIG. **2** includes an equal number of memory devices **20** and data handlers **30**, other configurations are possible. In some configurations there are more memory devices **20** than data handlers **30** or vice versa. Moreover, the one or more data handlers **30** may be operatively coupled to a subset of the data ports **21** of one the memory devices **20** instead of all of the data ports **21** of one of the memory devices **20**. In other embodiments, one or more data handlers **30** may be operatively coupled to a subset or all of the data ports **21** of more than one of the memory

devices **20**. For example, in one embodiment, each of the data handlers **30** are operatively coupled to all of the data ports of two memory devices **20**.

In certain embodiments, the plurality of data handlers **28** comprises at least two physically separate components mounted on the PCB **12**. For example, the plurality of data handlers **28** may include at least two physically separate integrated circuit packages. The physically separate integrated circuit packages are mounted on different portions of the PCB **12** in some embodiments. For example, each of the eight data handlers **30***a*-**30***h* shown in FIG. **2** may include physically separate integrated circuit packages mounted on different portions of the PCB **12**. While eight data handlers **30** are shown in FIG. **2**, other numbers of data handlers **30** are possible including fewer or more than eight.

In certain embodiments, each of the plurality of data handlers **30** is positioned on the PCB **12** proximate to the corresponding plurality of data ports. For example, each data handler **30** of certain embodiments is positioned closer to the corresponding plurality of data ports **21** than the data handler **30** is to the other data ports **21** of the plurality of memory devices **18**. For example, the data handler **30***a* is positioned closer to the corresponding plurality of data ports **21** of the memory device **40***a* than to the other data ports **21** of the other memory devices **40***b*-**40***h*.

FIG. **3** is a block diagram an example data module **28** and an example control module **22** in accordance with certain embodiments described herein. The control module **22** can be configured to generate address and control signals **24** for testing the plurality of memory devices **18**. In some embodiments, the control module **22** includes a control mixer element **32**. The control mixer element **32** may include a memory device controller **34** (e.g., a DRAM controller) and a test controller **36**. In certain embodiments, the control mixer element **32** generally controls the address and the control signals for the self-testing function.

In certain embodiments, the memory device controller **34** generally pre-processes address and control information before it sends the information to a register **40**. In one embodiment, the memory device controller **34** receives signals **38** (e.g., address and control signals) from the system memory controller **14** and signals **42** (e.g., address and controls signals) from the test controller **36**. The control module **22** of certain embodiments is configured to selectively input to the address and control ports of the plurality memory devices **18** either the address and control signals **38** from the system memory controller **14** or the address and control signals **42** from the control module **22** (e.g., from the test controller **36**). For example, the memory device controller **34** may send either the signals **38** from the system memory controller **14** or, alternatively, the signals **42** from the test controller **36**, to the register **40** depending on whether the memory module **10** is in normal (non-test) mode or in a test mode, respectively. In one embodiment, the memory device controller **34** generates the address and control signals for memory device (e.g., DRAM device) operations. The test controller **36** controls the generation of the address and control signal sequences to be used during the self-testing operation of the memory module **10** and also communicates with the data module **28**. The control module **22** may be implemented in the control register of the memory module **10** in certain embodiments. In various embodiments, the control module **22** includes discrete logic, one or more application-specific integrated circuit (ASICs), one or more microprocessors, one or more field-programmable gate arrays (FPGAs), or one or more computer-programmable logic device (CPLDs).

The data module **28** and the subcomponents thereof (e.g., the data handlers **30**) may be in communication with one or more of the memory devices **20**, the control module **22**, and the memory controller **14**. In certain embodiments, the data module **28** comprises a plurality of data handlers **30**. In other embodiments the data module **28** includes at least one data handler **30**. Each of the data handlers **30** of certain embodiments comprises a switch **44**. For example, the switch **44** may include a data multiplexer/demultiplexer ("data mux/demux"). The switch **44** may provide a bi-directional data multiplexer function. In certain embodiments, the switch **44** is configured to selectively input to the corresponding plurality of data ports either data signals **48** from the system memory controller **14** or data signals **50** from the data handler logic element **46**. The switch **44** of certain embodiments may further be configured to receive data signals **52** (e.g., during a read operation) from the plurality of memory devices **18** and to propagate the data signals **52** to the data handler logic element **46** and/or the memory controller **14**. In some embodiments, for example, the switch **44** selectively inputs the data signals **48** to be written to the plurality of memory devices **18** from the system memory controller **14** when the memory module **10** is a normal (non-test mode) mode and, alternatively, inputs the data signals **50** from the data handler logic element **46** during a test mode. While the switch **44** is shown as being included in the data handler **30** in the example of FIG. **3**, other configurations are possible. For example, in other embodiments the switch **44** may be logically and/or physically separated from the data handler module **28** and/or the data handlers **30**.

Each of the data handlers **30** of certain embodiments further includes a data handler logic element **46**. The data handler logic element **46** of certain embodiments comprises a data generation element **54** and a verification element **56**. The data generation element **54** may be configured to generate data signals (e.g., patterns of data signals) for writing to the corresponding plurality of data ports, for example. The data signals and/or patterns of data signals may be based on information (e.g., programming or configuration information) the data handler logic element **46** receives from the control module **22**, for example. The data may be cyclic data in some embodiments or non-cyclic data in other embodiments. For example, the cyclic data may comprise at least one predetermined pattern of data which repeats or is cycled two or more times. In various embodiments, the data comprises one or more incrementing patterns or decrementing patterns, for example. In other embodiments, the data comprises a pattern which alternates each bit on successive memory writes. For example, a memory write comprising one or more hexadecimal "A" characters (each corresponding to a four-bit binary word of "1010") may be followed by a memory write comprising one or more hexadecimal "5" characters (each corresponding to a four-bit binary word of "0101"). The data may be generated in a variety of ways. In one embodiment, the data is generated based on a current write address value. For example, in one example configuration, on a first write cycle, hexadecimal "A's" are generated and written to even address locations and hexadecimal "5's" are generated and written to odd address locations, and on a second write cycle, "5's" are written to even addresses and "A's" are written to odd addresses, and this pattern repeats in subsequent cycles. The data may be generated based previously written data (e.g., inverting each of the bits of a previously written data word) in some embodiments. In general, any manner of generating a cyclic or otherwise deterministic data pattern may be compatible with embodiments described herein. In other embodiments, random or pseudorandom data may be generated and

written to the corresponding plurality of data ports. For example, a linear feedback shift register (LFSR) may be used in some embodiments. In addition, the data patterns may be programmable. For example, the data patterns may be programmable based on information received by the data generation element **54** from the memory controller **14** (e.g., through the control module **22**), or from the control module **22**.

The plurality of data handlers **30** are further configured to read data from the corresponding plurality of data ports. For example, the verification element **56** may be configured to receive data from the corresponding plurality of data ports (e.g., through the switch **44** during a test mode). The verification element **56** may further be configured to check for failures in the operation of the plurality memory devices **18** by verifying that data read from the corresponding plurality of data ports corresponds to the data generated by the data handler **30** and written to the corresponding plurality of data ports.

In certain embodiments, the verification element **56** is configured to perform the verification without storing a copy of the data written to the corresponding plurality of data ports or accessing a stored copy beyond the data read from the plurality of memory devices **18**. For example, the verification element **56** does not store or access a copy of the data that is written to the corresponding plurality of data ports except for the data stored and read back from the plurality of memory devices **18**. As such, the memory module **10** of certain embodiments advantageously does not require separate memory for storing duplicate copies of test data that is written to the plurality of memory devices **18** for later comparison. For example, the verification element **56** may calculate comparison data and may compares the comparison data to the data read from the corresponding plurality of data ports. In certain embodiments, the comparison data comprises data which expresses the data or values expected to be received from the plurality of memory devices **18** if the write, store, and read processes of the data using the data module **28** and the plurality of memory devices **18** are performed correctly or as expected. The calculation may be performed simultaneously or substantially simultaneously with receiving the data read from the corresponding plurality of data ports in certain embodiments. In other embodiments, the calculation is performed either before or after receiving the data. Other configurations are possible. For example, in one embodiment, the verification element **56** does store a separate copy of the data written to the corresponding plurality of data ports upon writing the data and compares the separate copy to the read data received from the plurality of memory devices **18**.

The verification element **56** of certain embodiments calculates the comparison data based on the cyclic data. For example, in one embodiment, the verification element **56** calculates the comparison data in substantially the same manner that the data generation element **54** generates the data as described above (e.g., based on a current write address, using an LFSR, etc.). As such, the comparison data of certain embodiments is substantially a repeat of the data written. In one example embodiment, the data handler **30** is configured (e.g., is programmed by the control module **22**) to write an alternating series of "A's" and "5's" to the data ports of the corresponding plurality of data ports as described herein. For example, the data handler **30** may be configured to write one or more "A's" to the first address location of an N-word memory device **20** including the corresponding plurality of data ports. The data handler **30** may then write one or more "5's" to the second address location, one or more "A's" to the third address location and so on until the data handler **30** has written to all N memory locations. The verification element **56** of the example embodiment then calculates the comparison data based on the cyclic data written to the corresponding plurality of data ports. For example, the verification element **56** calculates a comparison word including one or more "A's", "5's," and "A's," respectively, to be compared to the data read from the first, second, and third address locations based on the cyclic data (e.g., based on the known cycle of the data). In some embodiments, the comparison data is calculated based on a current read address.

In certain embodiments, data associated with failures in the operation of the plurality of memory devices **18** are stored in the data module **28**. For example, data read from the corresponding plurality of data ports which do not correspond to (e.g., match) the comparison data calculated by the verification element **56** may be stored in the data module. Moreover, in some embodiments, memory addresses associated with the failures in the operation of the plurality of memory devices **18** are stored in the control module **22**. For example, the data handler **30** may communicate data failures (e.g., when data read from the corresponding plurality of data ports does not correspond to calculated comparison data) to the control module **22** which may then store the addresses corresponding to the data failure. In certain embodiments, the memory module **12** is configured to report failures (e.g., the failed data, the address corresponding to the memory location of the failed data, and/or expected data) via the I^2C interface **15** to the memory controller **14**. In addition, in certain embodiments the test controller **36** and/or the data handlers **30** may be updated through the I^2C interface **15** with new data patterns and/or with alternative memory access sequences to conduct AC tests (e.g., tests of the power, current, I/O speed, etc.).

In various embodiments, the components of the data module **28** (e.g., the switch **44**, the data handlers **30**, the data handler logic element **46**, the data generation element **54**, and/or verification element **56**) may include discrete logic, one or more application-specific integrated circuits (ASICs) one or more microprocessors, one or more field-programmable gate arrays (FPGAs), or one or more computer-programmable logic devices (CPLDs). Additionally, one or more of the various functional blocks (e.g., the switch **44**) of the data module **28** of FIG. **3** may not be included. In some embodiments, additional functional blocks may be included. Moreover, some of the functional blocks are described as separate functional blocks for illustration purposes and may comprise one physical component. For example, in one embodiment, each of the data handlers **30** and the corresponding switch **44**, data generation element **54**, and verification element **56** comprise one physical component (e.g., are included in one integrated circuit package). In another embodiment, the data module **28** comprises one physical component.

Referring again to FIG. **1**, a self-testing memory module **10** of certain embodiments comprises a printed circuit board (PCB) **12** and is configured to be operatively coupled to a memory controller **14** of a computer system **16**. The memory module **10** further includes a plurality of memory devices **18** on the printed circuit board **12** where each memory device **20** of the plurality of memory devices **18** comprising data, address, and control ports. The memory module **10** further comprises a control module **22** configured to generate address and control signals for testing the plurality of memory devices **18**. In certain embodiments, the memory module **10** further comprises a data module **28** comprising at least one data handler **30** and operatively coupled to a corresponding plurality of the data ports of one or more of the memory devices **20**. The data handler **30** is configured to generate cyclic data