NON-CONFIDENTIAL

No. 2010-1483

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

RAMBUS, INC.,

Appellant,

— v. —

INTERNATIONAL TRADE COMMISSION,

Appellee,

and

NVIDIA CORPORATION ET AL.,

Intervenors.

ON APPEAL FROM THE UNITED STATES INTERNATIONAL TRADE COMMISSION IN INVESTIGATION NO. 337-TA-661

CONFIDENTIAL ANSWERING BRIEF OF INTERVENORS NVIDIA CORPORATION ET AL.

RUFFIN B. CORDELL
ANDREW R. KOPSIDAS
FISH & RICHARDSON P.C.
1425 K Street, NW, 11th Floor
Washington, DC 20005
Tel. No. 202-626-6449
Attorneys for Customer Intervenors

NEEL CHATTERJEE
RICHARD SWOPE
ORRICK, HERRINGTON
& SUTCLIFFE LLP
1000 Marsh Road
Menlo Park, CA 94025
Attorneys for Intervenor NVIDIA Corporation

MARK S. DAVIES
ORRICK, HERRINGTON
& SUTCLIFFE LLP
1152 15th Street, NW
Washington, DC 20005
Tel. No. 202-339-8400
Attorneys for Intervenor NVIDIA Corporation

E. JOSHUA ROSENKRANZ
ORRICK, HERRINGTON
& SUTCLIFFE LLP
51 West 52nd Street
New York, NY 10019
Attorneys for Intervenor NVIDIA Corporation

KAREN JOHNSON-MCKEWAN
ORRICK, HERRINGTON
& SUTCLIFFE LLP
405 Howard Street
San Francisco, CA 94105
Attorney for Intervenor NVIDIA Corporation

***Caption continued on next page**

COMPANION CASE TO THE FOLLOWING CONSOLIDATED CASES:

2010-1556

ASUSTEK COMPUTER INC., ASUS COMPUTER INTERNATIONAL, INC., BFG TECHNOLOGIES, INC., BOISTAR MICROTECH (U.S.A.) CORP., BIOSTAR MICROTECH INTERNATIONAL CORP., DIABLOTEK INC., EVGA CORP., G.B.T. INC., GIGABYTE TECHNOLOGY CO., LTD., HEWLETT-PACKARD COMPANY, MSI COMPUTER CORP., MICRO-STAR INTERNATIONAL COMPANY, LTD., GRACOM TECHNOLOGIES LLC (FORMERLY KNOWN AS PALIT MULTIMEDIA INC.), PALIT MICROSYSTEMS LTD., PINE TECHNOLOGY (MACAO COMMERCIAL OFFSHORE) LTD., AND SPARKLE COMPUTER COMPANY, LTD.,
Appellants,
v.

INTERNATIONAL TRADE COMMISSION,
Appellee,

and

RAMBUS, INC.,
Intervenor,

and

NVIDIA CORPORATION,
Intervenor.

2010-1557

NVIDIA CORPORATION
Appellant,

v.

INTERNATIONAL TRADE COMMISSION,
Appellee,

and

RAMBUS, INC.,
Intervenor

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rule 47.4 counsel of record for Intervenor NVIDIA Corporation certifies the following:

1. The full name of the party represented by counsel of record is NVIDIA Corporation, a corporation formed under the laws of California.

2. The entity identified above is the real party in interest.

3. No parent corporation or any publicly held company owns 10 percent or more of the stock of NVIDIA.

4. The names of all law firms and the partners and associates that have appeared for NVIDIA Corporation in the proceeding before the United States International Trade Commission or who are expected to appear for the party in this Court are:

Orrick, Herrington & Sutcliffe LLP: I. Neel Chatterjee, Kenneth J. Halpern (no longer with Orrick), Mark S. Davies, E. Joshua Rosenkranz, Karen G. Johnson-McKewan, Fabio Marino, Deborah E. Fishman (no longer with Orrick), William H. Wright, Mark Wine, Lauren J. Parker, Rachel M. McKenzie and Katherine M. Kopp.

Fish & Richardson P.C.: Ruffin B. Cordell, Andrew R. Kopsidas, Kori Anne Bagrowski (no longer with F&R), Peter J. Sawert, Anita E. Kadala, John P. Brinkmann, John Lahad (no longer with F&R), Barbara A. Benoit (no longer with F&R), Jeffrey R. Whieldon, Wasif Qureshi, Michael Chibib (no longer with F&R), John F. Horvath (no longer with F&R), David M. Hoffman, Josh Tucker (no longer with F&R), Timothy J. Devlin, Joshua M. Masur (no longer with F&R), Leeron G. Kalay and Kimberley Kennedy (no longer with F&R).

Dated: May 9, 2011

Respectfully submitted,

By 

Mark S. Davies
Attorney For Appellant
NVIDIA Corporation

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rule 47.4 counsel of record for the customer Intervenors certifies the following:

1. The full names of the parties we represent in this case are:

ASUSTeK Computer, Inc. and ASUS Computer International

BFG Technologies, Inc.

Biostar Microtech (U.S.A.) Corporation

Biostar Microtech International Corporation

Diablotek Inc.

EVGA Corporation

G.B.T. Inc.

Giga-Byte Technology Co., Limited

Hewlett-Packard Company

MSI Computer Corporation

Micro-Star International Company, Limited

Gracom Technologies LLC

Palit Microsystems Limited

Pine Technology (Macao Commercial Offshore) Limited

Sparkle Computer Company, Ltd.

2. All the real parties in interest represented by us are named in the caption.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the parties or amicus curiae represented by us are:

ASUS Computer International, Inc. is a wholly owned subsidiary of ASUSTeK Computer, Inc.

Biostar Microtech (U.S.A.) Corporation is a subsidiary of Biostar Microtech International Corporation.

G.B.T. Inc. is a subsidiary of Giga-Byte Technology Co., Limited.

MSI Computer Corporation is a subsidiary of Micro-Star International Company, Limited.

Gracom Technologies LLC is a subsidiary of Palit Microsystems Limited.

Pine Technology (Macao Commercial Offshore) Limited is a subsidiary of Pan Eagle Limited, which is a subsidiary of Pine Technology (BVI) Limited, which is a subsidiary of PINE Technology Holdings Limited.

4. The names of all law firms and the partners or associates that appeared for the parties now represented by us in the agency or are expected to appear in this Court, are:

Fish & Richardson P.C.: Ruffin B. Cordell, Andrew R. Kopsidas, Kori Anne Bagrowski (no longer with F&R), Peter J. Sawert, Anita E. Kadala, John P. Brinkmann, John Lahad (no longer with F&R), Barbara A. Benoit (no longer with F&R), Jeffrey R. Whieldon, Wasif Qureshi, Michael Chibib (no longer with F&R), John F. Horvath (no longer with F&R), David M. Hoffman, Josh Tucker (no longer with F&R), Timothy J. Devlin, Joshua M. Masur (no longer with F&R), Leeron G. Kaly and Kimberley Kennedy (no longer with F&R).

Dated: May 9, 2011

Respectfully submitted,

By Ruffin B. Cordell /EG

Ruffin B. Cordell
Attorney For Customer Intervenors

TABLE OF CONTENTS

Page

TABLE OF CONTENTS

Page

TABLE OF CONTENTS

Material has been deleted from pages 12, 56 and 57 of the non-confidential brief of Intervenors NVIDIA et. al. This material is deemed confidential business information pursuant to 19 U.S.C. 1337(n) and 19 C.F.R. 210.5, and pursuant to the Protective Order entered December 10, 2008. The material omitted from these pages contains confidential licensing information.

TABLE OF AUTHORITIES

Page

FEDERAL CASES

TABLE OF AUTHORITIES
(continued)

Page

TABLE OF AUTHORITIES
(continued)

Page

TABLE OF AUTHORITIES
(continued)

Page

FEDERAL STATUTES

OTHER AUTHORITIES

PLEADINGS

STATEMENT OF RELATED CASES

1. By order of the Court, this case is coordinated with appeals No. 2010-1557 and No. 2010-1556. In those appeals, NVIDIA Corporation ("NVIDIA") and its customers appeal portions of a ruling of the International Trade Commission ("ITC"). Here, in appeal No. 2010-1483, Rambus Inc. ("Rambus") appeals other portions of the same ITC ruling.

2. On October 6, 2010, this Court heard re-argument in two coordinated appeals involving Rambus, No. 2009-1263 (reviewing *Micron Technology, Inc. v. Rambus, Inc.*, 255 F.R.D. 135 (D. Del. 2009)) and No. 2009-1299 (reviewing *Hynix Semiconductor Inc. v. Rambus Inc.*, 591 F. Supp. 2d 1038 (N.D. Cal. 2006)). The *Micron* and *Hynix* matters address Rambus's document destruction practices prior to commencing patent litigation against the leading members of the semiconductor industry. Resolution of those appeals may address the discovery-related issues raised here by Rambus.

3. The parties that are involved in this appeal are also involved in ongoing district court proceedings. *See Rambus Inc. v. NVIDIA Corp.*, No. 08-03343 (N.D. Cal. filed July 10, 2008). Rambus has also recently

filed other actions against the same parties in the district court and the ITC. *See Rambus Inc. v. NVIDIA Corp.*, No. 10-5448 (N.D. Cal. filed Dec. 1, 2010); *In the Matter of Certain Semiconductor Chips and Products Containing Same*, Inv. No. 337-TA-2771 (USITC filed Dec. 2, 2010). Rambus has also filed actions against other parties that raise identical or similar issues. *See Rambus Inc. v. Broadcom Corp.*, No. 10-5437 (N.D. Cal. filed Dec. 1, 2010); *Rambus Inc. v. Freescale Semiconductor, Inc.*, No. 10-5445 (N.D. Cal. filed Dec. 1, 2010); *Rambus Inc. v. LSI Corp.*, No. 10-5446 (N.D. Cal. filed Dec. 1, 2010); *Rambus Inc. v. Mediatek Inc.*, No. 10-5447 (N.D. Cal. filed Dec. 1, 2010); *Rambus Inc. v. STMicroelectronics N.V.*, No. 10-5449 (N.D. Cal. filed Dec. 1, 2010); *In the Matter of Certain Semiconductor Chips and Products Containing Same*, Inv. No. 337-TA-2771 (USITC filed Dec. 2, 2010).

INTRODUCTION

Soon after Rambus, Inc. ("Rambus") launched its first set of patent assaults against industry standards for computer memory, it filed a patent application directed to coordinating computer memory components. This application tried to create a new theatre of operations, Dual In-Line Memory Modules ("DIMMs"). Specifically, the Ware patents that resulted from that application address well-recognized problems caused by a simple physical reality: every electronic signal necessarily takes time to travel on a wire from point A to point B. In the tightly synchronized high-speed operations of a modern computer, slight differences in signal arrival time can lead to signal interference and thus degrade performance.

In the type of computer memory at issue here, each memory device is directly connected to the memory controller by wires that carry *data* signals and each memory device shares wires with other devices that carry *control* signals. Because the individual memory devices are sequentially connected to the control path, the distance between each memory device and the controller varies depending on where in the sequence the particular memory device is connected. Likewise, even

though the individual memory devices are directly connected to the data path, the distance that data signals must travel between each memory device and the controller varies depending on where in the sequence the particular memory device is connected.

To address the timing challenges that result from the different path distances, the Ware patents claim a method and system that includes (1) "control transmit circuitry" with a "shared control signal path" that results in "different" "respective times required for the control signal to propagate from the memory controller to the memory devices" and (2) "timing circuitry" that "delay[s] reception of data signals" "based, at least in part, on the time required" for the control signals to travel "from the memory controller to a respective memory device."

The International Trade Commission ("ITC") properly concluded, however, that Rambus's Frederick Ware was not the first to invent the solution that the Ware patents described. IBM's Paul Coteus obtained a patent three years earlier that teaches the use of "timing circuitry" to delay reading data signals until control signals originating from a greater distance have time to arrive. As a result, the ITC properly

ruled that the Ware patents are invalid in light of the prior art. 35 U.S.C. § 102(e).

Moreover, as the ITC also recognized, even if Rambus's claimed differences between the Coteus and Ware patents are credited, the Ware claims are still invalid because the claimed invention is obvious. The need to compensate for differences in control signal circuitry lengths was obvious to those skilled in the art, as was compensating for those differences by delaying some data signals based in part on the additional transmission time required for other signals originating at a greater distance from the memory controller. 35 U.S.C. § 103.

The ITC's decision can also be affirmed on the alternative basis that Rambus never established the "domestic industry" prerequisite for an ITC action based on the Ware patents.

Separately, the Administrative Law Judge ("ALJ") had ample reason to pierce Rambus's assertions of attorney-client privilege over certain material related to its document destruction program. Indeed, three other courts and a federal agency had already concluded that Rambus could not claim privilege with respect to materials related to that improper document destruction program.

ISSUES PRESENTED

The first question presented is whether the ITC had substantial evidence to find that the claims of the Ware patents are invalid because they are anticipated (35 U.S.C. § 102(e)) by the earlier Coteus patent.

The second question presented is whether the ITC correctly found that even if the "differences between the Coteus patent and the asserted claims of the Ware patents urged by Rambus are assumed to exist," the Ware patents are invalid because obvious (35 U.S.C. § 103) where one of ordinary skill in the art would have realized that the methods in the Coteus patent could be used to address problems caused by signal delays.

As an alternative basis for affirming the ITC, the third question presented is whether Rambus failed to meet the "domestic industry" requirement (19 U.S.C. § 1337(a)(2)) for an ITC action because Rambus did not clearly link any licensing revenue to the Ware patents.

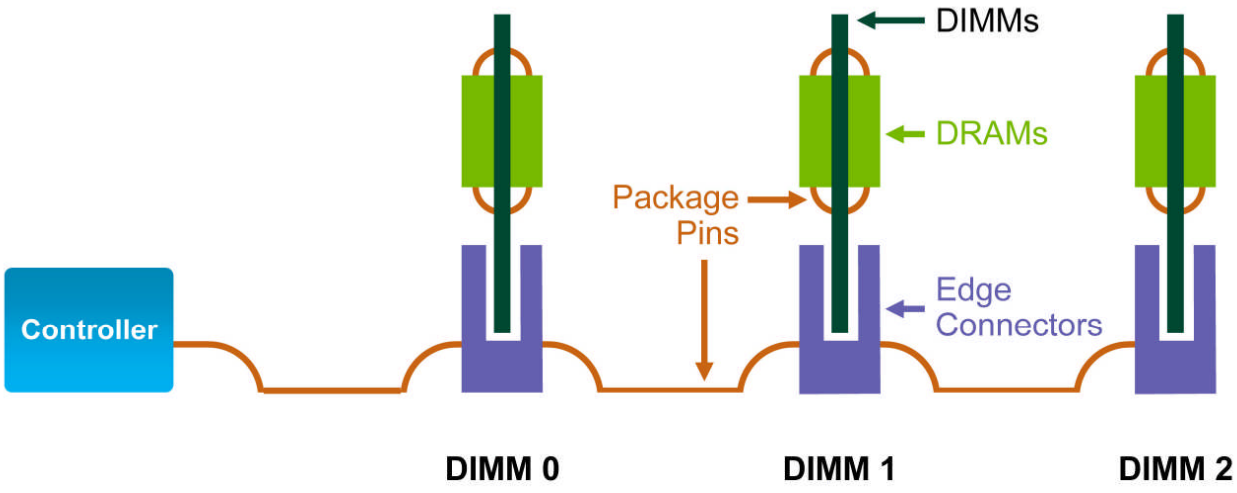
Rambus urges (at 2) that this appeal also presents the question whether the ALJ was correct in piercing Rambus's asserted attorney-client privilege—as three district courts and another federal agency

have done—because of (and with respect to) Rambus's improper destruction of documents.

STATEMENT OF THE FACTS

A. Dual In-Line Memory Modules

This case involves the high-speed transfer of data between a memory controller and a series of memory chips or Dynamic Random Access Memory ("DRAMs") that are arranged in a specific fashion. A typical memory module consists of eight DRAMs. In Dual In-Line Memory Modules ("DIMMs"), several DRAM memory modules are combined in a line:



Thus, a DRAM is one unit for storing memory; a DRAM module is a collection of DRAMs; and a DIMM is a collection of DRAM modules.

As the picture illustrates, a DIMM includes a number of interconnected DRAMs on a single physical structure (light green). The DRAMs are organized in a line, one after another, on a physical card (the DIMM) that connects to a communication path consisting of a set of wires. All DIMMs share a communication path (orange) to and from the memory controller (blue). A subset of those wires is referred to as a "bus." A data bus transmits signals representing information while a control bus transmits signals that control the timing and flow of the information carried on the data buses. The memory controller, in turn, communicates with the Central Processing Unit ("CPU") (not shown).

Because the DRAMs in a DIMM are arranged in a series, a signal from the memory controller reaches the first DRAM before it reaches the second DRAM, and reaches the second DRAM before the third, and so on. Similarly, when a signal leaves the first DRAM, the signal arrives at the memory controller faster than the signal from the second DRAM and so on. These subtle differences in arrival time caused by the physical locations of the DRAMs may impair the operation of a high-speed memory system. For example, if a program requires data that is stored in two separate DRAMs on a DIMM for a particular

operation and data from one of the DRAMs is delayed because the data path is longer, the system may not have the correct data to perform the operation correctly.

B. The Ware Patents

In April 2001 (A861), Frederick Ware, a Rambus employee, filed a patent application describing a "method and apparatus for coordinating memory operations among diversely-located memory components." A863. The background section notes "a continually increasing need to process more information in a given amount of time." A861. As computers "process each element of information in a shorter amount of time," the amount of time available "approaches the physical speed limits that govern the communication of electronic signals." *Id.* "While it would be ideal to be able to move electronic representations of information with no delay, such delay is unavoidable." *Id.* "In fact, not only is the delay unavoidable, but, since the amount of delay is a function of distance, the delay varies according to the relative locations of the devices in communication." *Id.* "[A]s performance demands have increased, traditional timing paradigms have imposed barriers to progress." *Id.*

At issue here are claims in two patents issued in 2007 descending from the Ware application, U.S. Patents No. 7,177,998 ('998) and 7,210,016 ('016).[1] Claim 7 of the '998 is representative:

> A memory controller comprising:
>
> **control transmit circuitry** to transmit a control signal to a plurality of memory devices via a shared control signal path, the shared control signal path being coupled to each of the memory devices at a different point along its length such that respective times required for the control signal to propagate from the memory controller to the memory devices are different;
>
> data receive circuitry to receive data signals from the memory devices via respective data signal paths;
>
> and
>
> **timing circuitry** to delay reception of data signals on each of the data signal paths by a respective time interval that is based, at least in part, on the time required for the control signal to propagate on the control signal path from the memory controller to a respective memory device of the memory devices.

[1] The '998 is titled Method, System and Memory Controller Utilizing Adjustable Read Data Delay Settings and claims "read" operations (i.e., checking the DRAM for data). A824. The '016 is titled Method, System and Memory Controller Utilizing Adjustable Write Data Delay Settings and refers to write operations (i.e., storing data in the DRAM). A892. The difference between read and write operations is not material here. *See* Ad98 (patents "substantially similar").

A888 (emphasis added).[2]

Under the Ware claims, a memory controller includes (a) control signal "transmit circuitry" that connects the controller serially to multiple memory devices and thus results in "different" times required for the control signal to travel from the controller to the device (and vice versa) and (b) "timing circuitry" that delays reception of the data based, at least in part, on the "time required for the control signal" to travel on the signal path.

C. The Coteus Patent

In June 1998, almost three years ***before*** the Ware patent application was filed, Paul William Coteus et al. filed for a patent, assigned to IBM, titled Smart Memory Interface (U.S. Patent No. 6,292,903). A40620. The Coteus patent discloses "a technique for optimizing the performance of a memory subsystem of a computer system." A40637.[3] The patent describes "memory subsystems" that include a "memory controller" coupled to "multiple memory modules

---

[2] For the convenience of the Court, the representative Ware patent ('998) is included in the addendum to this brief. The pagination retains the pagination used in the Joint Appendix.

[3] For the convenience of the Court, the Coteus patent is included in the addendum to this brief. The pagination retains the pagination used in the Joint Appendix.

(e.g., dual in-mode memory modules)." *Id*. The Coteus invention is a method and apparatus that "compensates for any differences in times at which portions of data being transferred from the memory controller to the memory device, and vice versa, arrive at the respective destination components." A40638. Coteus claims a method that includes "determin[ing] a range within which temporal relationships of electrical signals need to be set in order to operate the system without error" and "transferring information" "in accordance with temporal relationships between" the signals. A40650. In other words, as with the Ware claims, the Coteus patent describes a memory system that delays reception of data signals based, at least in part, on the time required for the control signal to travel on the signal path.

Rambus did not cite the Coteus patent in the Ware applications. A824. As a result, the Coteus patent was not before the PTO when it evaluated the Ware applications.

D. Prior Proceedings

In 2007, the Joint Electronic Device Engineering Council ("JEDEC") released a standard for Double Data Rate, Synchronous DRAM Dual In-Line Memory Modules ("DDR 3 SDRAM DIMMs").

A40010; A9506. The "traditional tree structure" for DIMMs involved arranging DRAMs in "equal length" distances from the memory controller "to eliminate any variation of the timing of the control signals." A861; *see also* A40033. "Rather than the traditional tree structure utilized on legacy DDR modules," the DDR3 standard implemented a "fundamental topology change" that was "instrumental in enabling the higher operating speeds of DDR3." A40033.

Soon thereafter, in November 2008, Rambus filed a complaint with the ITC alleging that NVIDIA and its customers were violating section 337 of the Tariff Act of 1930 (19 U.S.C. § 1337) by the sale and importation within the United States of "certain semiconductor chips having synchronous dynamic random access memory controllers" and products containing such chips. A1173. Rambus invoked the Ware patents, arguing that certain products that interface with the then-new DDR3 DRAM infringe the Ware patents. Ad79.

The Commission instituted an investigation (No. 337-TA-661). A28666-70.

1. The Administrative Law Judge rejects Rambus's allegations based on the Ware patents.

At the outset of the proceedings, NVIDIA argued that Rambus's complaint did not meet the "domestic industry" prerequisite for ITC action (19 U.S.C. § 1337(a)(2)). The ALJ refused to accept that argument [redacted] The ITC declined to review the domestic industry decision. A113.

In Order 15, the ALJ compelled Rambus to produce allegedly privileged documents about its document destruction practices which had already been produced in other cases. A67-79. The ALJ concluded that Rambus had not "'zealously' protected its privileged materials" (A75) and "that the crime-fraud exception" to attorney-client privilege "applies here" (A79).

On the merits, the ALJ correctly explained that "the parties' dispute focuses on whether Coteus discloses … 'control transmit circuitry" with the same topology (i.e., physical organization) as the Ware claims and whether Coteus discloses "a timing circuitry

limitation" that compensates for delays in the "topology as covered by the asserted claims of the Ware Patent." Ad124. In a detailed analysis (discussed further below at 19-46), the ALJ concluded that the asserted claims "are invalid under 35 U.S.C. § 102 for anticipation." Ad185.

The ALJ also found that the asserted claims "are invalid under 35 U.S.C. § 103 for obviousness." Ad185. The ALJ explained that as the operating speed of the DIMMs increased, one of ordinary skill in the relevant art would understand the need to compensate for the delays caused by varying lengths the signals must travel. Ad151. Although Rambus argued that "commercial success, long felt need, failure of others, copying and praise by others" suggested the patent was not obvious, the ALJ stated that "Rambus presents no specific instances of the factors listed above." Ad152. The ALJ found that "by simply making a conclusory argument that simply states that Rambus has met the general secondary consideration factors, Rambus has blatantly failed to meet its burden of establishing secondary considerations and failed to establish a nexus between the evidence and the merits of the claimed invention." *Id.*

Accordingly, the ALJ ruled that Rambus had not "established that a violation exists of section 337." Ad186.

2. The International Trade Commission affirms the ALJ.

With respect to the ALJ's anticipation ruling, the ITC affirmed without further discussion. Ad274.

With respect to the ALJ's obviousness findings, the ITC provided "further analysis" "in which it is assumed that certain limitations of asserted claims of the Ware patents are not disclosed by the Coteus patent." *Id.* In particular, the "differences between the Coteus patent and asserted claims of the Ware patents urged by Rambus are assumed to exist." *Id.* "Under an alternative scenario in which Coteus does not explicitly disclose the disputed limitations of the asserted Ware claims," the ITC "agree[d] with [NVIDIA] that it would have been obvious to one of ordinary skill in the art to implement the system and method of the asserted Ware claims." Ad275. The ITC agreed that "one of ordinary skill in the art would have understood the need to compensate for device-to-device propagation delays" and "agree[d] that one of ordinary skill in the art would have understood that Coteus discloses a method to compensate for such delays." *Id.* The ITC found that "even if Coteus

does not explicitly disclose the disputed limitations of the asserted Ware claims, its disclosure would render these limitations obvious to one of ordinary skill in the art." *Id.*

The ITC also agreed with the ALJ that "Rambus failed to meet its burden of establishing secondary considerations and failed to establish a nexus between the evidence and the merits of the claimed invention, and therefore find no secondary considerations of non-obviousness." *Id.*

SUMMARY OF ARGUMENT

1. Figure 2 and the associated description of the Coteus patent provides substantial evidence to support the ITC's invalidation of the Ware claims as anticipated. Figure 2 discloses "transmit circuitry" for control signals that results in "different" "respective times to propagate." For example, Coteus Figure 2 shows a control line that reaches a series of DRAMs, and the varying line lengths inherently result in varying times for the signals to arrive at each DRAM. Coteus Figure 2 also discloses "timing circuitry"—delay elements 15a-n and 16a-n—that is based in part "on the time for the control signal to propagate on the control signal path." Because the Ware patents claim nothing more than transmit and timing circuitry that are disclosed in

the Coteus patent, the Ware patents are invalid. The Ware patents teach the public nothing new.

2. The ITC found that "even if Coteus does not explicitly disclose the disputed limitations of the asserted Ware claims, its disclosure would render these limitations obvious to one of ordinary skill in the art." Rambus incorrectly claims that the ITC "agreed" with Rambus that the Coteus patent does not disclose "shared control signal path length differences between memory devices." To the contrary, the ITC discussed an "**alternative scenario** in which Coteus does not **explicitly** disclose the disputed limitations of the asserted Ware claims." There is nothing contradictory in the ITC holding that the Ware claims are anticipated or, in the alternative and accepting Rambus's reading of the Coteus patent, obvious.

The ALJ also rejected Rambus's argument (at 56-57) regarding objective indications that the Ware patents disclose a non-obvious invention. The ALJ declined to consider the secondary consideration arguments because "permitting Rambus to present their arguments on secondary considerations based on conclusory sentences without providing any analysis is tantamount to allowing the parties to

circumvent the page limitation for the post-hearing briefs set by the ALJ." The ITC agreed with the ALJ. On appeal, Rambus presents a more extended discussion of secondary considerations, but it is far too late for that. Moreover, none of this evidence is linked to the Ware patents and thus, on the merits, Rambus's highlighting of the market success of certain products cannot overcome the obviousness of the Ware claims.

3. As an alternative basis for affirmance, the Court should find that Rambus failed to establish the domestic industry requirement for invoking the ITC's authority. To assure that the ITC retains its proper and limited place in the U.S. patent system, Congress requires that an ITC complainant clearly link the asserted patent to exploitation of the patent. Rambus failed to show any licensing is clearly linked to the Ware patents.

4. The ALJ properly found, consistent with many other decision-makers, that Rambus has waived any relevant assertion of the attorney-client privilege and that in any event the privilege is properly pierced here.

STANDARD OF REVIEW

"Anticipation is a question of fact" that the Court reviews for "substantial evidence." *In re Giacomini*, 612 F.3d 1380, 1382-83 (Fed. Cir. 2010).

"'Obviousness is a question of law based on underlying factual inquiries,'" and thus the Court "'review[s] the Commission's ultimate determination de novo and factual determinations for substantial evidence.'" *Lucky Litter LLC v. Int'l Trade Comm'n*, 403 F. App'x 490, 494-95 (Fed. Cir. 2010) (quoting *Vizio, Inc. v. Int'l Trade Comm'n*, 605 F.3d 1330, 1342 (Fed. Cir. 2010)).

The Court reviews legal determinations in Section 337 investigations without deference. *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1367 (Fed. Cir. 2003).

Discovery rulings "will not be overturned absent a clear and harmful abuse of discretion." *Ayres v. Dep't of Homeland Security*, 280 F. App'x 991, 995 (Fed. Cir. 2008).

ARGUMENT

Rambus's scattershot approach to appellate briefing reflects the lack of any serious complaint with the ITC's well-reasoned decision declaring the Ware patents invalid in light of the prior art Coteus

patent. At bottom, Rambus puts the most weight on a suggestion that the Coteus patent does not expressly describe control lines of different lengths. This is a striking claim considering the Coteus patent expressly discusses, not once but several times, "variations between the lengths of the buses." Even more striking, the Coteus patent includes a diagram that shows control lines of different lengths. The fact that these are the best arguments Rambus can advance here speaks volumes about the soundness of the invalidity decision below. The ITC should be affirmed.

I. THE WARE PATENTS ARE INVALID BECAUSE THE INVENTION CLAIMED WAS PREVIOUSLY DISCLOSED BY THE COTEUS PATENT

The ITC had substantial evidence to declare the Ware patents invalid because they are anticipated by the Coteus patent. Under 35 U.S.C. § 102(e)(2), a patent is invalid if "the invention was described in ... a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent." "A claim is anticipated" under Section 102(e) "if each and every element as set forth in the claim is found, either expressly or inherently described,

in a single prior art reference." *Verdegaal Bros., Inc. v. Union Oil Co. of Cal.*, 814 F.2d 628, 631 (Fed Cir. 1987).

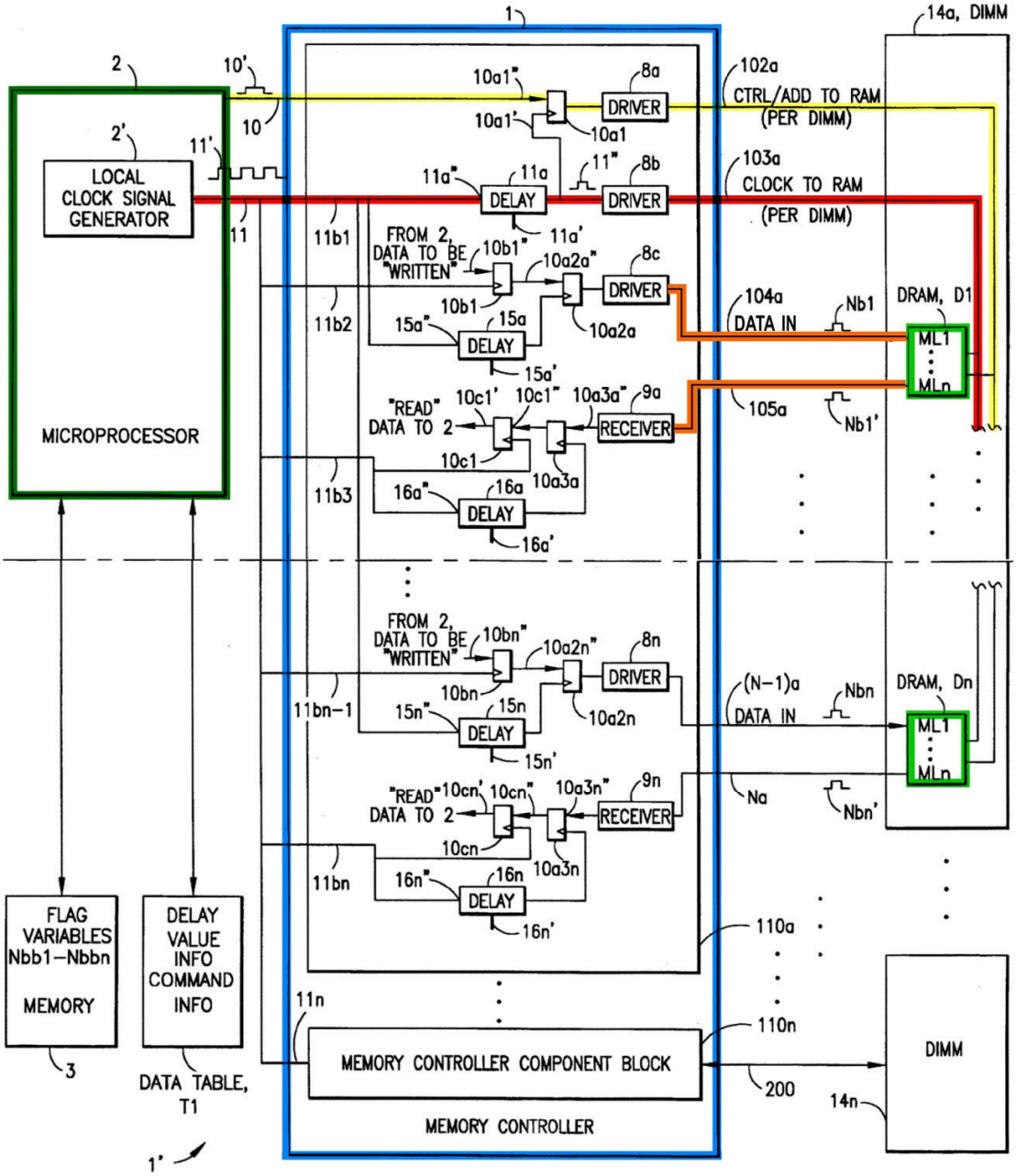
Rambus concedes that the Coteus patent "is prior art to the Ware patents" for purposes of Section 102(e). Rambus Opening Brief ("R.O.B.") 14. Rambus only points to two terms—"control transmit circuitry" and "timing circuitry"—that could possibly distinguish the Ware patents from the Coteus patent. Ad124. As shown next, however, the Coteus patent discloses both control transmit circuitry and timing circuitry. The Coteus patent anticipates the Ware patents.

A. The ITC Correctly Found That The Coteus Patent Discloses "Control Transmit Circuitry" With Different Signal Transmission Times

The ALJ found that Figure 2 of Coteus discloses the same circuit layout called for in the "control transmit circuitry" limitation of claim 7 of the Ware patent. Ad130. In particular, the ALJ found that paths 102a and 103a disclose a control transmit circuit with different transmission times. The ALJ's reading of Figure 2 is correct, and thus the ITC had substantial evidence to affirm the ALJ's finding. Ad274. Rambus's appeal arguments rest on misreadings of the Coteus patent.

1. Paths 102a and 103a of the Coteus patent disclose "control transmit circuitry" with different transmission times.

A memory system "suitable for practicing" the Coteus invention is set out in Figure 2:



A40622-23. Coteus Figure 2 shows a microprocessor, or CPU (2 in dark green), and a memory controller (1 in blue). The figure also shows two control lines—the address line (102a in yellow) and a clock line (103a in red)—that reach a series of DRAMs in a DIMM (D1 to Dn in light green). The figure also shows data lines (104a and 105a, both in orange) that reach the same series of DRAMs in a DIMM (D1 to Dn).

The signals on the data circuitry are the information being written to, or read from, the DRAM—i.e., the "ones and zeros" of digital data—while the control circuitry signals instruct the DRAM whether and when to read or write information. DRAMs on a DIMM are sequentially connected to a shared control path (yellow), but each has a unique data path (orange). Because the DRAMs are sequentially located on the DIMM, the length of the path of both the data line and the control line between the memory control and an individual DRAM varies. *See, e.g.*, A40639 (5:18-25); A40641 (9:48-10:5); A40622-23 (Fig. 2).

Regardless of the type of signal transmitted on a wire, the laws of physics remain the same. The signals from the memory controller travelling on the control lines will reach closer DRAMs before reaching

DRAMs that are sequentially further along the control line. Also, data signals travelling on the data lines necessarily will reach closer DRAMs before reaching DRAMs that are sequentially further along the DIMM.

Discussing this figure, the Coteus patent describes how variations in line lengths cause errors due to variances in when the signals arrive. The patent states:

> As was previously described, in at least some memory subsystems, **there may be variations between the lengths of the buses** 102a [control], 103a [clock], 104a-(n-1)a [data write] and 105a-((N)a) [data read] employed for coupling a memory controller to memory devices. … The **variations in the bus lengths can cause data** that is transferred through different ones of the buses [*i.e.*, the data buses] **to arrive at destination components at different times**, and may ultimately result in portions of the data not being simultaneously loaded into the destination components.

A40641 (9:51-9:62) (emphasis added). The control and other buses thus are of different lengths, and these variations cause data transferred along the data buses to be written or read at different times due to those differences in bus lengths.

NVIDIA's expert confirmed that the Coteus control signal paths are of varying lengths and thus of varying transmission times. Regarding Coteus Figure 2, Dr. Subramanian testified:

> There are multiple memory devices labeled DRAM D1–DRAM Dn in Figures 2a and 2b, and we can see that the control signal such as the clock signal path 103a and control signal path 102a are coupled to these at different points along their respective lengths. That is very clear from the figure.

A23434. He explained: "Given this configuration, signals transmitted down either of the paths will necessarily arrive at the DRAM D1 before they arrive at the DRAM Dn. There is simply a greater distance for the signals to travel to reach DRAM Dn than DRAM D1, and this takes extra time." *Id.*

Stripped of technical terminology, the point here is an obvious and simple one: it takes time to get from Point A to Point B. In the demanding environment of a computer memory system, the timing differences caused by the serial arrangement of DRAMs connected to the controller set out in Figure 2 necessarily have operational consequences. Indeed, this is the basic background point that the Ware patent emphasizes. A861. As the Ware patent explains, "[w]hile it

would be ideal to be able to move electronic representations of information with no delay, such delay is unavoidable." *Id.*

In view of this necessary delay, the ALJ found that Figure 2 of Coteus discloses the same topology called for in the "control transmit circuitry" limitation of claim 7 of the Ware patent. Ad128-129.

2. Rambus misreads the Coteus specification.

Rambus argues (at 38-44, 53-54) that Coteus is "silent" about any timing differences in signals due to varying distances the signals travel. But Rambus ignores the plain language of the Coteus specification and the basic disclosure taught in Figure 2.

Rambus's principal argument (at 39) is that the Coteus patent does not expressly assert that the control signal paths have different lengths. Rambus states that the Coteus patent "explains in detail that its data paths have different lengths and provides a mechanism for delaying data transmission to compensate for delays." R.O.B. 39 (citing A40641 at 9:58-62). From this, Rambus draws a negative inference, suggesting that the lack of similar language about control signal paths means that the control signal paths have identical lengths. *Id.*

But the Coteus patent expressly provides that the control signal paths are of different lengths. As explained above (at 21-25), in the Coteus patent, buses 102a and 103a are the control signal paths. The Coteus patent states: "[T]here may be variations between the lengths of the buses 102a, 103a, 104a-(n-1)a, and 105a-((N)a) employed for coupling a memory controller to memory devices." A40641 (9:52-9:55). Not only does the patent expressly make clear that the control paths are of different lengths, but the patent notes the problem caused by the differences: "The variations in the bus lengths can cause data that is transferred through different ones of the buses to arrive at destination components at different times, and may ultimately result in portions of the data not being simultaneously loaded into the destination components." A40641 (9:58-9:62).

Moreover, where the Coteus patent describes the invention apart from any particular embodiment, the patent is clear that the invention applies to differences in the length of any type of bus and is not limited to data or any other specific type of bus. *See* A40649-50 (26:66-27:4) ("technique of the invention may also be performed to overcome latency

resulting" "from data loading variations on the buses."); A40637 (1:67) ("variations in the lengths of buses").

Not only does the text directly refute Rambus's suggestion, Figure 2 does as well. As the ALJ explained, "the topology shown in Coteus's Figure 2 inherently results in 'different' propagation times for signals propagated between the memory controller 1 and the memory devices D1-Dn on paths 102a and 103a." Ad137 n.14. Thus, even if "one of ordinary skill in the art would have mistakenly believed that the signals reach the devices at 'essentially the same time,'" the ALJ held that under "established inherency case law," the timing circuitry limitation "is clearly met" by Figure 2. *Id.*

Under this Court's inherency law, a "necessary consequence" of a prior disclosure can "suppl[y] the missing aspect" of a disclosure for purpose of invalidating a patent due to prior art. As an example of the Court's inherency law, the ALJ cited *Schering Corp. v. Geneva Pharmaceuticals, Inc.,* 339 F.3d 1373 (Fed. Cir. 2003). Ad137 n.14. There, the record showed that a "patient ingesting loratadine would necessarily metabolize that compound to DCL [descarboethoxy-loratadine]." 339 F.3d at 1380. The prior art did not "disclose any

compound that is identifiable as DCL," *id.* at 1378, but did disclose administering loratadine to a patient, *id.* at 1381. Although the prior art lacked an express disclosure of DCL, the Court found that claims to DCL were invalidated by the prior art reference showing administration of loratadine to a patient: "[i]nherency supplied the missing aspect of the description." *Id.* at 1378-79.

The ALJ properly relied on *Schering Corp.* and similar cases here. The Coteus patent discloses differences in control line lengths, and because those differences necessarily lead to variations in the timing of the arrival of the signals, the patent necessarily discloses different "propagation times for signals propagated between the memory controller 1 and the memory devices D1-Dn on paths 102a and 103a." Ad137 n.14. *See also CVI/Beta Ventures, Inc. v. Tura LP*, 112 F.3d 1146, 1153 (Fed. Cir. 1997) ("the patent drawings are highly relevant in construing the … limitations of the claims.").

Rambus argues that relying on the drawing of the control lines "requires ignoring how all other lines in Coteus Figure 2 appear." R.O.B. 40. Rambus states that "Coteus draws the write data buses 104a to (N-1)a as appearing to have the same length but describes them

as having different lengths." *Id.* But the only reason the data buses appear to have the same lengths is that this is "not a scale drawing"; it "is an architectural drawing to show the general floor plan." A11382. As explained above (at 22), the differences in data path lengths vary depending on the particular location of a DRAM. Figure 2 does not attempt to draw the many unique data paths—paths that would necessarily overlap with each other in a schematic of this scale.[4]

Turning away from Figure 2, Rambus next argues that Coteus Figure 8 "is clear that the clock signals reach destination components at the same time." R.O.B. 41 (citing, e.g., A40637 at 2:5-11). But, again, Rambus misreads the Coteus patent. With regard to Figure 8, the very language that Rambus cites includes this: "there may be variations in the lengths of the buses" and the "variations in the lengths of the buses can cause data that is simultaneously transmitted" to "arrive ... at different times." A40637.

---

[4] Contrary to Rambus's statement (at 41), the cited testimony does **not** state that the clock lines are the same length. Rambus asked NVIDIA's expert: "Now, the length of the clock line to register 10c1 is shown to be shorter than the length of the clock line to register 10cn, correct?" A11381. The expert agreed, noting that "we are not talking about distances of centimeters … [i]t's distances of microns." A11382. Thus, the expert explained that "the line is shorter" even though the differences are "tiny." *Id.*

Rambus also attempts to turn a semantic difference into a substantive issue. Rambus states there is "evidence showing fly-by is different from multidrop," apparently suggesting that the Coteus patent cannot disclose the same circuitry as the Ware patents because Coteus refers to a "multi-drop" line. R.O.B. 41. "[R]egardless of the actual 'name' of the topology," the ALJ correctly noted that "the evidence shows that both Coteus and the Ware Patents disclose similar topologies." Ad129.

Contrary to Rambus's suggestion (at 43) that NVIDIA's expert improperly gave "gap-filling" testimony, the expert properly only explained the relevant technology. As the Court has stated, where a reference is silent regarding an inherent characteristic, extrinsic evidence, such as expert testimony, can be used to show that the reference discloses "technological facts … known to those in the field of the invention, albeit not known to judges." *Cont'l Can Co. USA v. Monsanto Co.*, 948 F.2d 1264, 1269-70 (Fed. Cir. 1991). In explaining that the Coteus control signal line is of varying length, and, thus, the signal will reach the DRAMs at different times, the expert merely explained known technological facts to the ALJ and thus did not expand

the Coteus disclosure. *In re Baxter Travenol Labs*, 952 F.2d 388, 390 (Fed. Cir. 1991) ("[E]xtrinsic evidence may be considered when it is used to explain, but not expand, the meaning of a reference.").

For this reason, Rambus's reliance on *Motorola Inc. v. InterDigital Technology Corp*., 121 F.3d 1461 (Fed. Cir. 1997) (cited at R.O.B. 38, 42), is unavailing. In *Motorola*, the prior art "[i]ndisputedly" did not disclose the "particular synchronization functions" claimed. *Id.* at 1472-73. Nevertheless, after an expert testified that the prior art "'does discuss the need for synchronization,'" the jury concluded that the prior art anticipated the patent. *Id*. at 1473. This Court reversed: "An expert's conclusory testimony, unsupported by the documentary evidence, cannot supplant the requirement of anticipatory disclosure in the prior art reference itself." *Id*. Here, however, the prior art Coteus patent itself, in the specification and Figure 2, "indisputedly" does disclose the control signals with varying bus lengths. The expert testimony here explains the technology, as is appropriate, and does not impermissibly expand the disclosures.

3. Rambus misreads the Ware patents.

Rambus incorrectly argues that the "Ware patent contradicts the ALJ's inherency finding because it makes clear that, when Ware was filed, clock lines to individual memory devices were equalized." R.O.B. 41 (citing the prior art discussed in Ware at A861 (1:61-2:7)). Contrary to Rambus's representation, the Coteus patent expressly addresses devices like DIMMs (A40637 (2:16-26)) and shows memory devices arranged in sequence along a control line that is thus not equalized (A40622-23). Indeed, the very point of the Coteus invention is to minimize errors in a system with varying bus lengths. A40641.

Moreover, as the ALJ emphasized, there is no material difference between the Ware patent's depiction of the circuitry, including the controller bus, and the depiction in the Coteus patent. Ware Figure 2 provides further detail for an "embodiment of the invention" (A862):

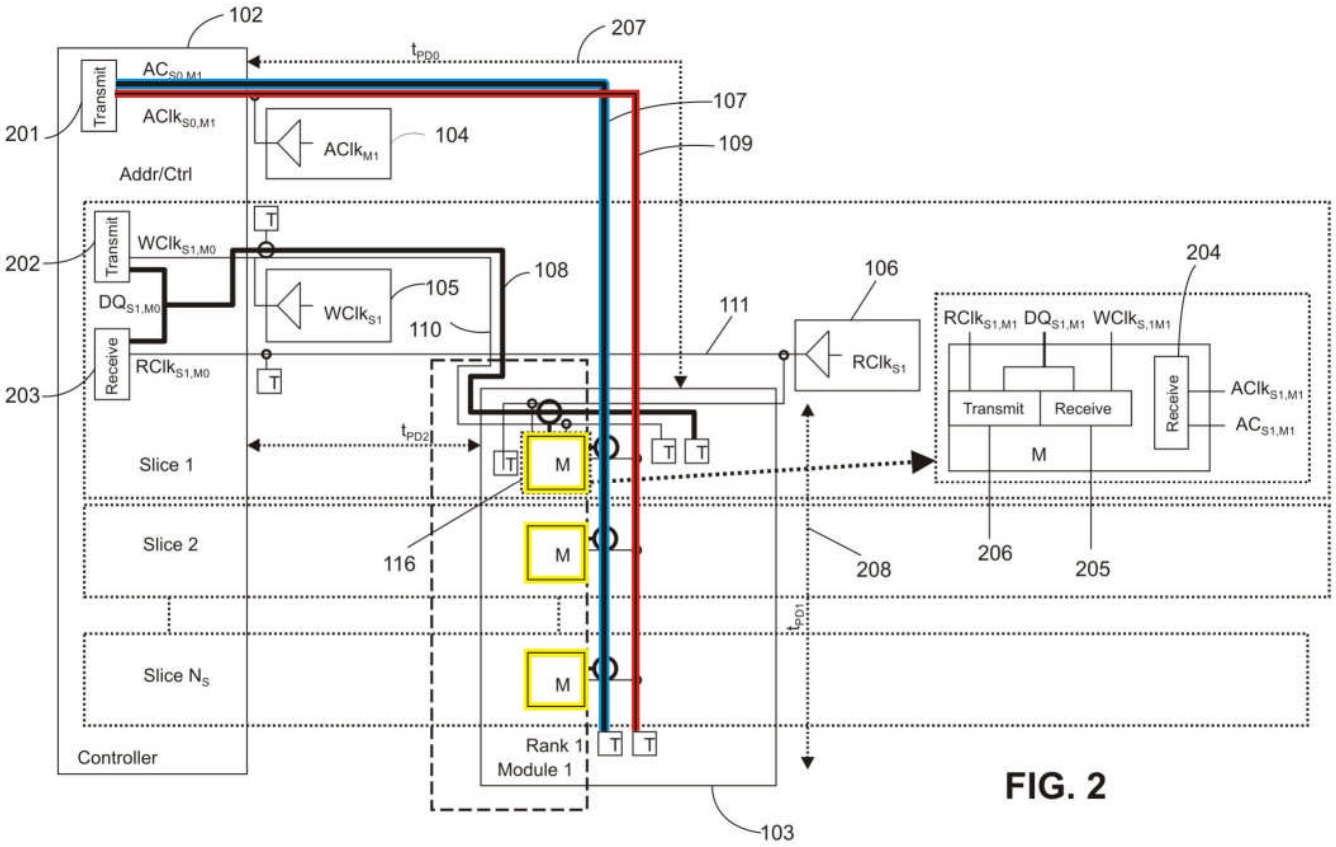


FIG. 2

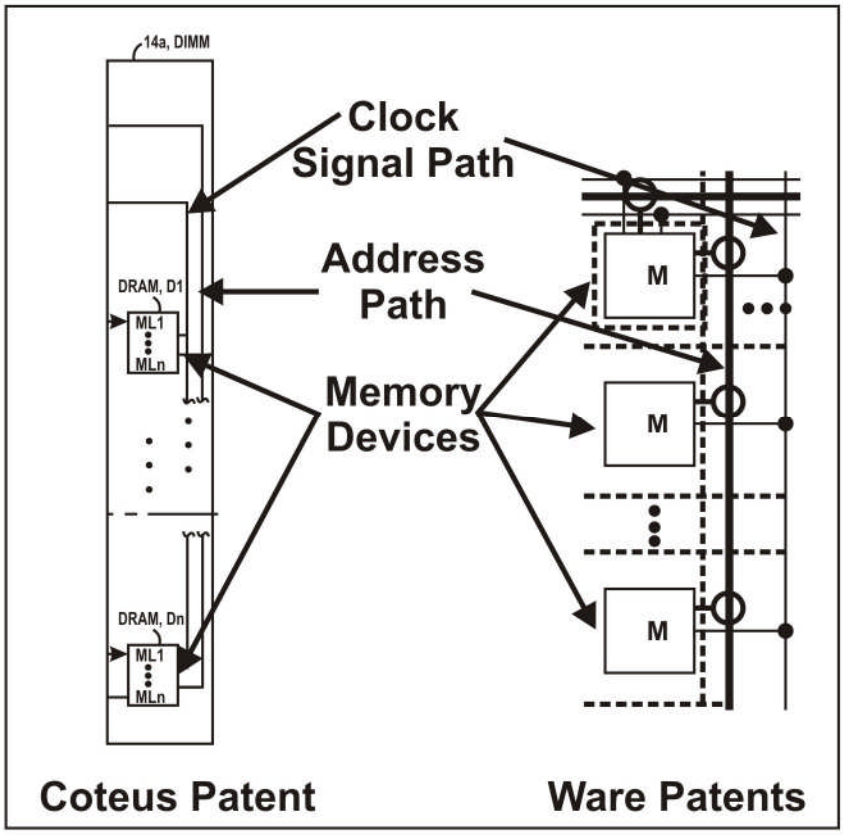
A828

See A828.[5] "As illustrated in Figure 2 of the Ware Patents, shared paths 107 [address in blue] and 109 [clock in red] electrically connect the memory controller to the plurality of memory devices 116 such that signals propagated thereon take progressively longer to reach each successive memory device [in yellow]." Ad128.

[5] For clarity, the image in the text is a reproduction of the image at A828.

Coteus Figure 2 also shows shared control paths, the address line (102a in yellow) and a clock line (103a in red), that electrically connect the memory controller (in blue) to the plurality of memory devices such that signals propagated thereon take progressively longer to reach each successive memory device. In other words, the **Coteus control lines are drawn in the same fashion as the Ware control lines and, therefore, in both systems control signals propagated thereon take progressively longer to reach successive memory devices.**

Here is an illustration of the similarity in control signal paths, using excerpts from Coteus Figure 2 (A40622-23) and Ware Figure 2 (A828):



The two drawings describe functionally identical circuits arranged in the same way. Both drawings show multiple memory devices connected serially along clock lines and address lines. If, as Rambus urges (at 41), the Coteus control (clock and address) lines are equalized, then so too are the same lines in the Ware patents. Instead, the ALJ properly judged that the specification and the drawings of both patents disclose lines of varying lengths.

Rambus also argues (at 53-54) that the ALJ erred by comparing the Coteus figure to the Ware figure. Rambus cites cases for the proposition that "'it is error for a court to compare in its infringement

analysis the accused product or process with the patentee's commercial embodiment.'" R.O.B. 54 (quoting *Zenith Labs., Inc. v. Bristol-Meyers Squibb Co.*, 19 F.3d 1418, 1423 (Fed. Cir. 1994)). But Figure 2 of Ware is not a commercial embodiment; it is part of the patent disclosure. Rambus states that "the only permissible side-by-side comparison is of the prior art's disclosure to the asserted patents claims." R.O.B. 54. And that is what the ALJ did here. The Ware figure was used to understand the meaning of the relevant Ware claims, such as "timing circuitry," and then was compared to the prior art Coteus disclosure, a disclosure that includes Figure 2.

B. The ITC Correctly Found That The Coteus Patent Discloses "Timing Circuitry" That Delays Data Signals Based In Part On The Time The Control Signals Must Travel

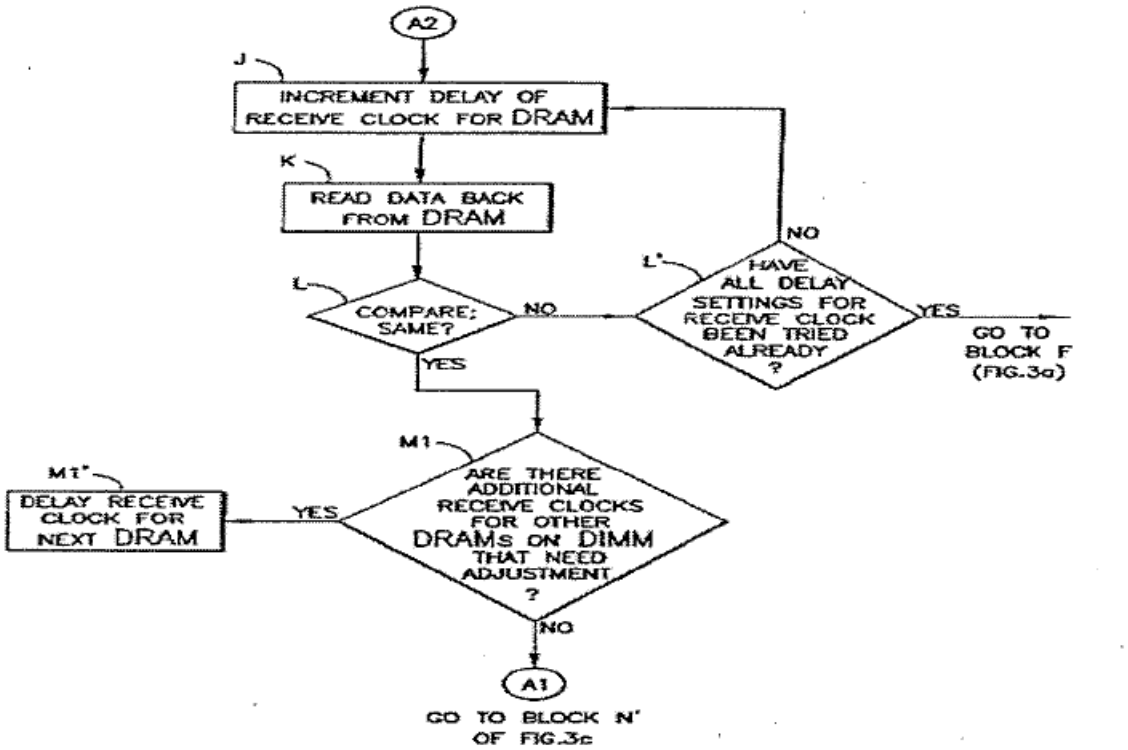
The ALJ found that the Coteus patent includes signals that delay the transmission of data and that the length of delay is based, in part, on the time it takes for the signal to travel to or from the memory controller. Ad138. In particular, the ALJ found that the delay elements (15a-15n and 16a-16n) are "programmed based, at least in part, on the propagation time of the clock signal 11' on clock signal path 103a." Ad131-32. In other words, the amount of time a signal is

delayed depends, at least in part, on the amount of time the control signal will take to travel ("propagate") along the wire. The ITC affirmed the finding of the ALJ that the Coteus patent disclosed the "timing circuitry" claimed in the Ware patents. Ad274. Rambus's appeal arguments rest on misreadings of the relevant figures, disclosure and testimony.

1. Elements 15a-15n and 16a-16n disclose timing circuitry that delays reading and writing data based in part on the different timing of the control signals.

In finding that the Coteus patent delays reading and writing data based at least in part on control signal delays due to different path lengths, the ALJ discussed two figures. The two figures describe distinct parts of the process for determining how long to delay a data signal. The first figure sets out the steps for determining if a delay is necessary. If a delay is needed, the second figure explains how the length of the delay is calculated.

The ALJ reproduced Figure 3b of the Coteus patent, a flowchart showing how the length of delay is determined:



Ad132. *See* A40642-43 (12:61-14:24). Under this algorithm, the amount of signal delay is changed if the data received from the DRAM at device start-up does not match the test data stored in the controller. *See* Ad132.

The ALJ then used another figure to explain how the "amount of delay is gradually adjusted until data is properly read by the memory controller 1 from the DRAMs D1-Dn." Ad132. Figure 4c of the Coteus patent shows how the delay is changed until data is properly read by the memory controller:

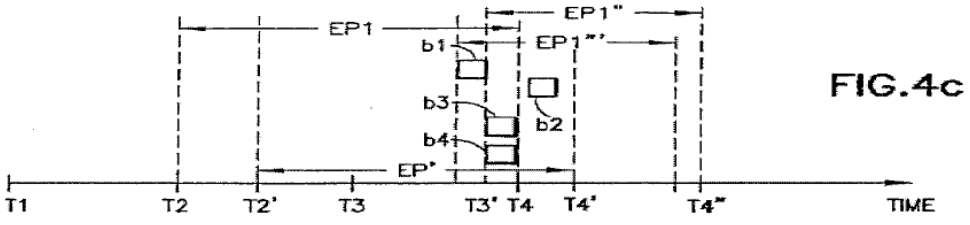


FIG.4c

Ad133. In Figure 4c (and elsewhere in the Coteus patent), "EP," the "enablement period," refers to a predetermined length of time in which a register on a memory controller is ready to receive or transfer data. *See*, e.g., A40639 (6:33-42) ("each register" "is 'enabled' for a predetermined time period" "for accepting (i.e., loading) data received at a respective input" "of the register."). In Figure 4c, the system adjusts the timing of the enablement period so that each of the four bits of data (b1-b4) arrive at the register within the enablement period of the memory controller. Ad133.

Explaining how Figures 3b and 4c work with Figure 2, the ALJ summarized that Coteus teaches both when and how to adjust the delay value: "Coteus adjusts the enablement period of the memory controller 1 by manipulating delay elements 16a through 16n of Figure 2 using the method of Figure 3b to thereby compensate for varying data arrival times from the DRAMs until all of bits b1-b4 appear within the

enablement period of the memory controller 1 as shown in Figure 4c." Ad133.

After using figures to explain how the Coteus patent teaches determining when to delay and, if a delay is necessary, how to adjust the delay, the ALJ linked the delay period to delays caused by the various lengths of the clock signal bus. The ALJ quoted NVIDIA's expert: "'Since the memory device sends the data when it receives a clock signal edge along the shared clock bus 103a, the delay on the receive side is calibrated based on the propagation time of the clock signal to the memory device.'" Ad133.

Accordingly, the ALJ properly found that there was a "direct relationship" between the time the DRAM transmits the data in response to the clock signal and the time the memory controller receives the data. Ad134. "Thus," the ALJ concluded, "any variation of when the clock signal is received by the memory devices D1-Dn necessarily results in a similar variation of when data is received from the memory devices D1-Dn by the memory controller 1." *Id.* "It follows that Coteus's read-leveling function for adjusting the enablement period based on the actual arrival time of bits b1-b4 must necessarily be based,

at least in part, on when the clock signal is received by the memory devices D1-Dn and they begin the process of transmitting data back to the memory controller 1." *Id.*

In a similar fashion, the ALJ correctly explained that, in figures 3c and 4d, Coteus teaches adjusting delays prior to data writing. Ad134-136. "The delay elements 15a-15n are indeed calibrated based, at least in part, by an algorithm that is based, at least in part, on the propagation time of a clock signal on shared clock path 103a to each individual memory device." Ad134. So too, "[b]ecause the enablement period EP1 is defined by when the clock signal is received by memory devices D1-Dn via shared path 103a and because the data transmission delay is based, at least in part, on that enablement period EP1, Coteus's write-leveling function is based at least in part on clock propagation time via shared path 103a." Ad136.

In sum, as the ALJ found, the adjustments of the enablement period are based, at least in part, on the time required for a control signal to propagate on a shared control signal path from the memory controller to the memory devices.

2. Rambus misreads the Coteus enablement period figures.

On appeal, Rambus principally argues (at 45-49, 49-52) that the Coteus patent shows that "its memory devices have the same enablement period." R.O.B. 45. Rambus asserts that "[t]he Coteus figures and disclosure are clear that, in successful write and read operations, there is a single collective enablement period, EP1, triggered at time T2." R.O.B. 45 (citing Coteus Fig. 4e (A40633), two portions of the disclosure (A40644 (16:18-22 and 21:13-24)), and expert testimony (A21952-55)). In other words, Rambus argues that Coteus assumes that all of the devices will read or write data at the same time (i.e., at time T2). But Rambus's argument misreads the disclosure, the figures and the expert testimony.

In rejecting Rambus's single enablement period theory, the ALJ explained that "Figure 2 details signals that serially hit DRAMs D1-Dn." Ad137. The ALJ noted that "Coteus describes a calibration procedure to calculate an individual delay for each DRAM so that data sent to that DRAM arrives in the center of that DRAM's enablement period." Ad137. Contrary to Rambus's view, "the memory devices D1-

Dn as illustrated in Figure 2a, 2b will correspondingly have different enablement periods." Ad138.

Put simply, Rambus's argument assumes away the very problem that the Coteus patent addresses. The only way that T2 is the same for all devices in a DIMM is if it were possible "to be able to move electronic representations of information with no delay." A861 (1:35-36). But as the Ware patent explains, "such delay is unavoidable." A861 (1:36-37). "In fact, not only is the delay unavoidable, but, since the amount of delay is a function of distance, the delay varies according to the relative locations of the devices in communication." A861 (1:37-40). It is not possible for a DIMM to have the same enablement period (i.e., T2) throughout the various DRAMs arranged at different distances along the various buses from the controller. As explained above (at 24, 33), the Ware patent and expert testimony establish that signals take time to travel along a line and thus signals will arrive at different times in a DIMM.

Rambus highlights (at 46-48) Coteus figures 4e and 4a and suggests they show that there is a single enablement period for each DRAM D1 to Dn. But the disclosure teaches the opposite. For example,

the specification states that Figure 4e depicts a relationship "between times at which enablement **periods** for memory devices of the memory subsystem of FIG. 1 occur." A40638 (4:4-6). So too, Figure 4a depicts a relationship "between times at which enablement **periods** for registers of the memory subsystem of FIG. 1 occur." A40638 (3:62-65) (emphasis added). Indeed, the descriptions of all but one of the Figure 4 figures use the plural "periods," and the only one that does not (Figure 4(f)) is an explanatory figure that applies to the other seven. A40638 (3:62-4:22). The very figures that Rambus highlights confirm that the patent discloses a method that uses multiple enablement periods. *See also* A40647 (21:21-23) ("an enablement period"); A40642 (11:25-34) (same); A11560 (Rambus's expert agreeing that "[w]hen a clock pulse reaches D1 on whatever conductor it's traveling on, that starts **the enablement period of DRAM D1**") (emphasis added). The "evidence shows," the ALJ properly found, "that Coteus actually discloses distinct enablement periods for each of the memory devices D1-Dn as embodied in Figures 1, 2a, 2b" and elsewhere. Ad137. *Accord* Ad138 ("the memory devices D1-Dn as illustrated in Fugure 2a, 2b will correspondingly have different enablement periods").

The reason that Coteus discloses a method with multiple enablement periods is that addressing the problem of variance in signal delays requires different enablement periods. The Coteus invention is a method and apparatus that "compensates for any differences in times at which portions of data being transferred from the memory controller to the memory device, and vice versa, arrive at the respective destination components." A40638 (3:29-32). If the enablement period is the same, then the method cannot compensate for any differences in transfer times. NVIDIA's expert explained that in the "hypothetical where the devices have an identical enablement period" and thus the "propagation delay within each device is the same," "I would expect that the clocks would have reached them at the same time." A11021. But the point of the Coteus invention is to address problems caused by the reality that the signal delay to each device is **not** the same. A40637-38.

Rambus argues that if Figure 4e shows multiple embodiments, it would show a "second, slightly shifted enablement period for DRAM D2, and additional distinct and shifted enablement periods for DRAMs D3 and Dn." R.O.B. 47. While the Coteus patent could have included such a figure, there is no requirement for such a figure where the rest of the

patent makes the claimed method clear. *See, e.g., Arlington Indus., Inc. v. Bridgeport Fittings, Inc.*, 632 F.3d 1246, 1254 (Fed. Cir. 2011) ("[D]rawings in a patent need not illustrate the full scope of the invention.").

Rambus argues (at 52) that "[d]elivery of the local clock signal is unrelated to the time required for any control signal to travel from the memory controller to the memory devices," (citing A40643 (13:63-14:16)). This is inaccurate. Rambus cites the Coteus patent's discussion of "delay element 16a," an element, that, as described above, is programmed in part based on when the clock signal on path 103a reaches the memory device. Because the timing of the local clock signals are based, in part, on the timing of the signals on path 103a, local clock signals are related to the time required for signals to travel from the controller to the memory device.

II. THE SUBJECT MATTER OF THE WARE PATENTS WAS OBVIOUS

The ITC's obviousness finding is supported by substantial evidence. Under 35 U.S.C. § 103(a), a patent may not issue if "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been

obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." Here, the ITC found that "one of ordinary skill in the art would have understood the need to compensate for device-to-device propagation delays on fly-by paths as depicted in Coteus's Figure 2," and "agree[d] that one of ordinary skill in the art would have understood that Coteus discloses a method to compensate for such delays." Ad275. Indeed, as the Ware patents emphasize, a delay is necessary any time two signals that must arrive at the same time each have different distances to travel. Rambus's arguments distort both the ITC's holding and the cited expert testimony.

A. The Ware Patents' Claimed Invention Was Obvious Because There Is Nothing Innovative In Delaying A Signal Until Other Signals Arrive

The ITC's obviousness finding is an alternative rationale that supports the ALJ's obviousness finding—a rationale that gives Rambus every benefit of the doubt. The ALJ explained that "at the operating speeds of memory devices" when the Ware patent was filed, "one of ordinary skill in the relevant art would have understood the need to compensate for device-to-device propagation delays on fly-by paths as

depicted in Coteus's Figure 2a, 2b." Ad151. Further, one of ordinary skill "would have understood that the methods disclosed in Coteus would compensate for device-to-device propagation delays that are present on fly-by paths." Ad151. Thus, the ALJ found "by clear and convincing evidence that the asserted claims of the Ware Patents are obvious in light of Coteus." Ad151.

The ITC provided "further analysis." The ITC "assumed that certain limitations of asserted claims of the Ware patents are not disclosed by the Coteus patent." Ad274. In particular, the "differences between the Coteus patent and asserted claims of the Ware patents urged by Rambus are assumed to exist." Ad274. "Under an alternative scenario in which Coteus does not **explicitly** disclose the disputed limitations of the asserted Ware claims," the ITC "agree[d] with [NVIDIA] that it would have been obvious to one of ordinary skill in the art to implement the system and method of the asserted Ware claims." Ad275 (emphasis added). The ITC agreed that "one of ordinary skill in the art would have understood the need to compensate for device-to-device propagation delays on fly-by paths as depicted in Coteus's Figure 2," and "agree[d] that one of ordinary skill in the art would have

understood that Coteus discloses a method to compensate for such delays." Ad275. The ITC found that "even if Coteus does not explicitly disclose the disputed limitations of the asserted Ware claims, its disclosure would render those limitations obvious to one of ordinary skill in the art." Ad275.

Rambus argues that the ITC "agreed" with Rambus that the Coteus patent does not disclose "shared control signal path length differences between memory devices." R.O.B. 55. This argument is not accurate. As noted above, the ITC discussed an "**alternative scenario** in which Coteus does not **explicitly** disclose the disputed limitations of the asserted Ware claims." Ad275. *See id*. ("**even if** Coteus does not **explicitly** disclose the disputed limitations of the asserted Ware claims" (emphasis added)); Ad274 ("**assumed** that certain limitations of asserted claims of the Ware patents are not disclosed by the Coteus patent." (emphasis added)).

Contrary to Rambus's suggestion (at 55), the ITC's obviousness ruling is entirely consistent with the ITC's statement that it would "assume" that the Coteus patent did not expressly disclose certain limitations as Rambus urged. Even if the Coteus patent does not

expressly disclose the signal path variations, the obviousness doctrine permits the ITC to conclude that one skilled in the art would have understood the "need to compensate for device-to-device propagation delays." Ad275. There is nothing contradictory in the Commission's holding. *See*, *e.g.*, *Süd-Chemie, Inc. v. Multisorb Techs., Inc.*, 554 F.3d 1001, 1005-06 (Fed. Cir. 2009) ("[B]y disclosing air-permeable films, Komatsu necessarily discloses films that are water-vapor-permeable…. Even if Komatsu did not disclose water-vapor-permeable films, it would have been obvious to a person of skill in the art to create a desiccant container with packaging materials that are permeable to water vapor.").

Rambus is wrong to suggest (at 56) that NVIDIA's expert testimony undermines the ALJ's holding that one of skill in the area would have realized that higher clock speeds require compensating for propagation delays. The ALJ found that "at the operating speeds of memory devices" when the Ware patent application was filed, one of ordinary skill in the relevant art would have "understood the need to compensate for device-to-device propagation delays." Ad151. To be sure, as Rambus notes, the expert testified that clock speed had

"nothing to do with implementing a 'fly-by' topology." A11000. But in so saying, the expert explained that at any clock speed there are "advantage[s] of arranging things in a fly-by topology" because "you can use a single clock line and distribute it across multiple devices." *Id.* This topology "allows you to maintain signal integrity." *Id.* Here, "[a]s you go to higher clock speeds" in DDR3, "signal quality was more important than ever." *Id.*

B. Rambus Waived Its Meritless "Secondary Considerations" Argument

The ALJ also rejected Rambus's argument (at 56-57) regarding objective indications that the Ware patents disclose a non-obvious invention. As the Supreme Court explained in *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17-18 (1966), "secondary considerations," such "as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented." But, as this Court has made clear, "[w]here the inventions represent[] no more than 'the predictable use of prior art elements according to their established functions,' the secondary considerations … are inadequate to establish nonobviousness as a matter of law." *W. Union*

Co. v. MoneyGram Payment Sys., 626 F.3d 1361, 1373 (Fed. Cir. 2010) (quoting *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007)). "[W]eak secondary considerations generally do not overcome a strong prima facie case of obviousness." 626 F.3d at 1373.

Rambus's cursory briefing before the ALJ on secondary considerations of obviousness relevant to the Coteus patent is at A12982. After one paragraph of boilerplate legal statements, the entirety of Rambus's secondary considerations argument is the following two sentences:

> Here, secondary considerations of nonobviousness include commercial success, long-felt need, failure of others, copying, and praise by others. Each of these secondary considerations, undisputed by Respondents, supports the non-obviousness of the asserted claims of the Barth I patents. CFF-VI.B.410-22.

A12982. As the quoted sentences show, Rambus recited general secondary considerations and then summarily asserted the considerations supported its nonobviousness argument. The only citation given was to various proposed findings of fact—findings not discussed in Rambus's briefing. This cursory treatment was not for lack of space, as the relevant brief was 120 pages long. A12917-13054.

The ALJ found this briefing inadequate. The ALJ stated that "Rambus presents no specific instances of the factors listed above." Ad152. The ALJ found that "by simply making a conclusory argument that simply states that Rambus has met the general secondary consideration factors, Rambus has blatantly failed to meet its burden of establishing secondary considerations and failed to establish a nexus between the evidence and the merits of the claimed invention." Ad152. The ALJ noted that "permitting Rambus to present their arguments on secondary considerations based on conclusory sentences without providing any analysis is tantamount to allowing the parties to circumvent the page limitation for the post-hearing briefs set by the ALJ." Ad152 n.17.

Because Rambus did not adequately brief the secondary considerations of obviousness, the ALJ ruled that Rambus "failed to establish a nexus between the evidence and the merits of the claimed invention." Ad152. The ITC agreed with the ALJ that Rambus "failed to establish a nexus between the evidence and the merits of the claimed invention." Ad275.

On appeal, Rambus argues (at 57) that the ALJ and ITC failed to address the evidence of secondary considerations relevant to obviousness. Rambus points to evidence such as "the Sony PlayStation 3 and Respondents' substantial sales of infringing products, A22058-61 at Q/A 639-653; A152-61; A40000-01 and A40002." R.O.B. 57. Rambus omits to say **why** the ALJ and ITC did not address this evidence, and with good reason: Rambus did not bother to present the evidence to the ALJ. Ad152 ("presents no specific instances of the factors"; "simply making a conclusory argument that simply states that Rambus has met the general secondary consideration factors"). Rambus's argument below did not discuss the Sony PlayStation, for example. Two summary sentences and a citation to twelve findings of fact do not present an argument that requires an ALJ's attention or resolution. *See*, *e.g.*, *Ajinomoto Co. v. Int'l Trade Comm'n*, 597 F.3d 1267, 1278 (Fed. Cir. 2010) ("[A] conclusory assertion unaccompanied by developed argumentation does not preserve the issue for appeal.").

Moreover, even the secondary considerations that Rambus now invokes cannot begin to show that the Ware claims satisfy the nonobviousness requirements. Rambus still has not provided any link

between increased sales of particular products and the innovation claimed in the Ware patents. Rambus emphasizes (at 57) that the ALJ found a "nexus" between the Ware patents and the licensing revenue for purposes of determining that Rambus met the domestic industry prerequisite for an ITC action. R.O.B. 57. As discussed next, however, the ALJ and ITC erred in their domestic industry ruling. Rambus did not establish a connection between the Ware patents and any licensing activity.

III. RAMBUS DID NOT DEMONSTRATE A SUBSTANTIAL INVESTMENT IN EXPLOITATION OF THE WARE PATENTS

As an alternative rationale for affirming the ITC's ruling of no Section 337 violation based on the Ware patents, the Court should find that Rambus failed to meet the domestic industry requirement for an ITC action. The analysis offered here tracks the analysis discussing the Barth patents in NVIDIA's opening brief ("N.O.B.") (at 29-35) in Nos. 2010-1557 and 2010-1556.

To invoke the ITC's jurisdiction and establish the necessary "substantial investment" in "exploitation" of a "patent" by reference to "licensing," 19 U.S.C. § 1337(a)(2)-(3), the ITC complainant must show

that there is a nexus between the licensing and the **particular patent** asserted in the ITC action.

The ALJ erred by finding a domestic industry related to the Ware patents because Rambus never clearly linked licensing to the Ware patents. [redacted] In those instances, the Ware patents are **not applicable** and thus outside the scope of the licenses.

[redacted]

[redacted]

[redacted] These facts further emphasize that the mere existence of these licenses is not enough to prove domestic industry with respect to the Ware patents.

Without evidence of a substantial investment in licensing related to the Ware patents, the ITC action should never have started.

IV. THE ALJ CORRECTLY ORDERED RAMBUS TO PRODUCE MATERIALS RELATED TO ITS DOCUMENT DESTRUCTION

In Order 15, the ALJ compelled Rambus to produce documents about its document destruction practices. A67-79. NVIDIA sought "[a]ll materials that Rambus has been ordered to produce pursuant to privilege piercing orders in *Hynix Semiconductor, Inc. v. Rambus, Inc.,* No. 00-20905 (N.D. Cal.), *Micron Technology, Inc. v. Rambus Inc.*, No. 00-792-SLR (D. Del.), *Rambus, Inc. v. Infineon Tech., AG*, No. 3:00cv524 (E.D. VA.), *Samsung Electronics Co., Ltd., v. Rambus Inc.*, No. 3:05cv406 (E.D. VA.), and *In the Matter of Rambus, Inc*., Docket No. 9302, (FTC)." A69; *see also* A70 (citing NVIDIA Reply Ex. 56 (listing requested documents)). The ALJ granted the motion. A79. Rambus asks the Court to vacate the ALJ's order. R.O.B. 57-66.

As demonstrated below, the ALJ's decision is an unremarkable application of waiver and privilege piercing doctrines that is squarely in line with four well-reasoned decisions. It should be upheld.

A. The ALJ Properly Required Rambus To Produce Allegedly Privileged Documents That Are Publicly Available

1. Rambus has waived any assertion of privilege over materials related to its document destruction.

The ALJ found that Rambus waived its privilege claims. Pointing to Rambus's voluntary disclosures in the *Hynix, Micron* and *Infineon* cases, as well as in proceedings before the Federal Trade Commission, the ALJ concluded that Rambus had not "'zealously' protected its privileged materials" (A75) and therefore ordered the documents produced (A79). The public record confirms that Rambus has failed to take steps to assure that the documents produced in other litigation, including those at issue here, were placed, and remained, under seal.

Of particular relevance here, each of the eleven allegedly privileged documents the ALJ cited in Order 15 (*see* R.O.B. at 59) was marked as an exhibit in the *Micron* litigation. A77-78. Eight of them bear an identification sticker indicating that they were also marked as exhibits in the *Infineon* litigation. *See* A25904; A25906-08; A25910-12;

A26055-64; A26066-67; A26157-60; A26162; and A26201-17 (Exs. 12-14, 23-24, 32-33 and 38 to Mem. in Support of NVIDIA's Mot. To Compel Production of Materials Subject to Privilege Piercing Orders). In *Hynix Semiconductor, Inc. v. Rambus Inc.*, No. C-00-20905, 2009 WL 2246204, at *1-2 (N.D. Cal. July 27, 2009), Judge Whyte noted that "[t]he allegedly privileged documents have apparently been available to the public for several years in the *Infineon* case file in Eastern District of Virginia," and that "[m]ost, if not all, of Rambus' allegedly privileged documents were filed in the Delaware action [*Micron*] and are publicly available."

A simple search of PACER confirms Judge Whyte's observations. At least five of the eleven documents Rambus claims are privileged are publicly available on the electronic docket in the *Micron* litigation, attached as exhibits to Micron's memorandum of law in support of its motion to compel certain materials. *See* Mem. in Support of Mot. To Compel Def. Rambus To Produce Documents, Testimony, and

Pleadings, Exs. 9, 10, 15, 32, 46, *Micron Tech., Inc. v. Rambus Inc.*, No. 00-792 (D. Del. March 28, 2005), Dkt. No. 638.[6]

Rambus asserts that it "has never voluntarily disclosed privileged documents relating to its document retention policy." R.O.B. 61 n.4. Even were that assertion accurate, Rambus's failure to take the steps necessary to assure that its (involuntarily) produced materials were placed, and remained, under seal waives any privilege that might otherwise exist. Under circumstances similar to those at issue here, the court in *The Navajo Nation v. Peabody Holding Co.*, 209 F. Supp. 2d 269, 284 (D.D.C. 2002), noted that it "appear[ed] that many of the documents" one party had produced to another pursuant to court order and stipulation "may have been included in public pleadings and referred to in oral argument." The court concluded that "[t]o the extent that these documents have been made public, any claim of confidentiality and privilege clearly must fail." *Id.* Because Rambus's

[6] Exhibit 9 to Micron's memorandum is the same as Exhibit 12 to NVIDIA's memorandum. Exhibit 10 to Micron's memorandum is the same as Exhibit 14 to NVIDIA's memorandum. Exhibit 15 to Micron's memorandum is the same as Exhibit 38 to NVIDIA's memorandum. Exhibit 32 to Micron's memorandum is the same as Exhibit 23 to NVIDIA's memorandum. And Exhibit 46 to Micron's memorandum is an earlier draft of Exhibit 24 to NVIDIA's memorandum.

allegedly privileged documents have been publicly available for some time, so too must its claim of privilege fail. *See In re Grand Jury (Impounded)*, 138 F.3d 978, 981 (3d Cir. 1998) ("[I]n the case of … involuntary disclosures, the party asserting the work product doctrine must pursue all reasonable means to restore the confidentiality of the materials and to prevent further disclosures within a reasonable period to continue to receive the protection of the privilege"); *Sec. Exch. Comm'n v. Lavin*, 111 F.3d 921, 929 (D.C. Cir. 1997) (privilege waived unless party "zealously protect[ed] the privileged materials, taking all reasonable steps to prevent their disclosure"); *The Navajo Nation v. Peabody Holding Co.*, 255 F.R.D. 37, 45 (D.D.C. 2009) ("[T]he party claiming privilege must prevent the introduction of privileged material into the public record.").

Moreover, Rambus's assertion that it never voluntarily disclosed privileged documents about its document destruction program is inaccurate. For example, in the *Infineon* and *Micron* cases, Rambus voluntarily produced Joel Karp's document retention policy memorandum, the slide presentation that Karp and other Rambus managers showed to Rambus employees regarding the document

retention program and the notes that Rambus's Consumer Communications Products Division Head, Kevin Donnelly, took during the July 1998 company-wide meetings regarding the document retention policy. *Rambus Inc. v. Infineon Techs. AG*, 220 F.R.D. 264, 288 (E.D. Va. 2004). Donnelly's notes described the "substance, content and protocols of the document retention policy." *Id.*[7]

Rambus cannot selectively disclose certain privileged documents and then refrain from producing other allegedly privileged documents related to the same subject matter. Its attempt to use some privileged documents about its document retention program in its defense, while refusing to disclose others, is an improper use of the privilege as both sword and shield that results in a waiver of the privilege for all documents relating to the subject matter of the documents disclosed. *In re Seagate Tech., LLC*, 497 F.3d 1360, 1372 (Fed. Cir. 2007) ("waiver applies to all other communications relating to the same subject

[7] In the *Infineon* litigation, Rambus also allowed witnesses to testify about the purposes of the document retention program, to promote the notion that the Court misunderstood the aims of the policy and the reason for destroying documents because of their "discoverability." *See* 220 F.R.D. at 285-86; *Rambus Inc. v. Infineon Techs. AG*, 222 F.R.D. 280, 284-85 (E.D. Va. 2004).

matter") (internal quotes omitted); *In re EchoStar Comm'ns Corp*., 448 F.3d 1294, 1301 (Fed. Cir. 2006).[8]

In a footnote, Rambus suggests that the ALJ's statements regarding waiver are "dicta" and that there was "not a finding of waiver." R.O.B. 61 n.4. But the ALJ's opinion expressly notes NVIDIA's argument that "Rambus has waived any privilege" (A74), addresses the "argument" and concludes Rambus did not "'zealously'" protect the privilege (A74-A75). Moreover, the opinion's "[t]herefore" clause is free-standing and thus refers to both the waiver and crime-fraud rationales. A79. "[W]here there are two grounds, upon either of which … [a] court may rest its decision, and it adopts both, 'the ruling on neither is obiter, but each is the judgment of the court, and of equal validity with the other.'" *United States v. Title Ins. & Trust Co*., 265 U.S. 472, 486 (1924) (quoting *Union Pacific R.R. Co. v. Mason City & Ft. Dodge R.R. Co*., 199 U. S. 160, 166 (1905)). *See Choctaw Nation v. United States*, 135 F. Supp. 536, 538 (Ct. Cl. Nov. 8, 1955) ("Although the decision would probably have been the same if either one of the

[8] Rambus also waived any privilege by disclosing "admitted trial exhibits from the 'unclean hands' trials in the *Infineon*, *Hynix* and *Micron* cases" to its testifying expert, John Montaña. A5069-5070. *See In re Pioneer Hi-Bred Int'l, Inc.*, 238 F.3d 1370, 1375 (Fed. Cir. 2001).

grounds had been lacking, yet that does not make both or either of the grounds obiter dictum." (citing *Title Ins. & Trust Co.*, 265 U.S. at 486)).

2. The crime-fraud exception to attorney-client privilege applies here.

In addition to finding waiver, the ALJ also found that the evidence regarding Rambus's document destruction was sufficient to satisfy the crime-fraud exception to the attorney-client privilege. A79. The ALJ correctly ordered the production of the documents sought by NVIDIA on that ground.

As the ALJ properly stated, to satisfy the crime-fraud exception, "the client must have made or received the otherwise privileged communication with the intent to further an unlawful or fraudulent act" and "the client must have carried out the crime or fraud." A75 (citing *In re Sealed Case*, 223 F.3d 775, 778-79 (D.C. Cir. 2000); *In re Spalding Sports Worldwide, Inc.*, 203 F.3d 800, 807 (Fed. Cir. 2000)). The ALJ explained that the party challenging the privilege "must first make a prima facie showing of a violation sufficiently serious to defeat the privilege, and second, establish some relationship between the communication at issue and the prima facie violation." A75 (citing *In re Sealed Case*, 754 F.2d 395, 399 (D.C. Cir. 1985); *In re Grand Jury*, 475

F.3d 1299, 1305 (D.C. Cir. 2007)); *see also Clark v. United States*, 289 U.S. 1, 15 (1933) ("A client who consults an attorney for advice that will serve him in the commission of a fraud" will not have recourse to the attorney-client privilege so long as there is "prima facie" evidence to support the existence of the fraud.).

The ALJ correctly observed that "[t]hree other district courts"—those in *Micron*, *Hynix* and *Infineon*—"have conducted extensive hearings and findings related to Rambus's spoliation of evidence." A76. The ALJ noted that "the facts surrounding the spoliation of evidence are unchanged," and concluded that "judicial resources need not be spent in recreating a record that has already been extensively developed by three other district courts." A76. The ALJ also noted that "Rambus's arguments against NVIDIA's motion to compel do no[t] dispute the factual findings of the district courts." A76.

Based on that undisputed factual record, the ALJ found "the record is sufficient to establish a prima facie showing of spoliation of evidence by Rambus." A78. The ALJ made the following findings of fact:

- Rambus hired Joel Karp as its Vice President of Intellectual Property, who began meeting with outside counsel in January of 1998 to discuss patent licensing and litigation options. A77.

- Rambus "actively consulted counsel in the development of its document destruction program as part of its litigation strategy," and "destroyed documents it reasonably knew or should have known were relevant [to] its intended litigation targets." A76 (citing *Infineon*, 222 F.R.D. at 282; *Micron Tech., Inc. v. Rambus Inc.*, 255 F.R.D. 135, 150-51 (D. Del. 2009)).

- The company thereafter implemented its new document retention policy "in part 'to allow Rambus to purge documents, including emails, from its files that might be discoverable in litigation.'" A77 (quoting *Micron,* 255 F.R.D. at 141).[9]

- In April of 1999, Karp instructed Rambus's outside counsel to begin "'clear[ing] out' the Rambus patent files for issued patents, including patents Rambus was preparing to assert in litigation, such as

[9] Notably, "Rambus had decided that it would go into litigation against manufacturers of SDRAM and control makers no later than July of 1998." Ad175. In June 1998, Rambus had already identified NVIDIA as a litigation target. A23291. Well before July 2000, Rambus had developed a detailed timeline for suing NVIDIA at the ITC. A23627-29.

the parent application of the Barth I patents." A78 (quoting *Micron*, 255 F.R.D. at 143).

- Pursuant to its chosen "litigation strategy," Rambus destroyed documents in a series of "Shred Days." A77-78 (citing *Infineon*, 222 F.R.D. at 291; *Micron*, 255 F.R.D. at 140-45).

The ALJ found the destroyed evidence was "relevant to NVIDIA's asserted defenses in this investigation, including documents related to its lack of domestic industry defense (licensing and contract negotiations), unclean hands (JEDEC meetings), and invalidity (prosecution history files and prior art)." A78-79. The ALJ concluded, on the strength of the evidence of intentional spoliation, "that the crime-fraud exception applies here." A79.[10]

Rambus offers three reasons for reversing the ALJ's crime-fraud determination: (1) the ALJ improperly relied on privileged documents in

---

[10] Though Rambus argues (at 64) that the ALJ "made no case-specific findings to support any piercing," the foregoing litany of factual findings demonstrates that the ALJ explained in detail why the record before him, which included exhibits presented in *Infineon* and *Micron*, as well as the courts' decisions in those cases, was entirely sufficient to support the piercing order in this case. *See* A78 ("Based on the foregoing findings, the record is sufficient to establish a prima facie showing of spoliation of evidence by Rambus. … Those documents [destroyed] included documents relevant to NVIDIA's asserted defenses *in this investigation* …." (emphasis added)).

deciding whether to apply the crime-fraud exception; (2) spoliation does not qualify as a basis for invoking the crime-fraud exception; and (3) the ALJ ultimately concluded there was no actionable spoliation in this case. R.O.B. 58-66. None of these rationales are persuasive.

Contrary to Rambus's suggestion (at 58), the ALJ's decision to order Rambus to disclose the material relevant to its document destruction is not "based" on allegedly privileged material. Rambus identifies Exhibits 9-14, 23-24, 32-33 and 38 as the documents the ALJ should not have relied on when he decided to pierce the attorney-client privilege. R.O.B. 59. But in every instance where the ALJ cites one or more of these exhibits, the principal sources of evidence in support of the relevant findings are either documents that Rambus admits are not privileged (e.g., Exhibits 7 & 8), published district court opinions (*Infineon* or *Micron*), or both. A76-78.

Rambus also cites *United States v. Zolin*, 491 U.S. 554, 570-72 (1989), for the uncontested rule that "a party opposing privilege must demonstrate through lawfully obtained *nonprivileged* material a reasonable basis to conclude that *in camera* review may reveal evidence to establish the applicability of the crime-fraud exception." R.O.B. 58.

The ALJ's decision in this case is consistent with *Zolin*. The ALJ relied on several earlier rulings that adhered to *Zolin* by first finding a reasonable basis to suspect improper document destruction based on an examination of non-privileged documents. A76 (citing the orders in *Hynix*, *Micron* and *Infineon*). In *Infineon*, for example, the court started with an examination of non-privileged materials and, with the holding of *Zolin* firmly in mind, found that those materials alone "rather strongly indicated that Rambus had explicitly linked development of its document retention policy and the shredding of documents with the company's preparations for patent litigation." 222 F.R.D. at 292. The court then proceeded to an *in camera* review of the privileged documents, which "confirmed what was established, in less detail, by the non-privileged evidence." *Id.* at 293. *See also*, *e.g.*, Order Compelling Production of Documents at 7, *Hynix Semiconductor, Inc. v. Rambus, Inc.,* No. 00-20905 (N.D. Cal. Jan. 31, 2005), Dkt. No. 729 ("non-privileged evidence submitted by Hynix in the original motion to dismiss establishes that Rambus anticipated commencing litigation against industry competitors"); Order Requesting Additional Briefing on Crime-Fraud Exception to the Attorney-Client Privilege at 2, *Hynix,* No.

00-20905 (N.D. Cal. Dec. 17, 2004), Dkt. No. 639 (Utilizing the two-part test set forth in [*Zolin*], this court determined, based on evidence presented by Hynix, that *in camera* review of the documents was appropriate."). Because the ALJ relied on the earlier courts' proper review of the same materials, the ALJ's decision is necessarily consistent with *Zolin*.

Rambus also argues that the ALJ "erred in characterizing spoliation as a crime or fraudulent scheme sufficient to compel the production of privileged documents." R.O.B. 62. The ALJ's decision is entirely in keeping with the purposes of the attorney-client privilege and the crime-fraud exception to that privilege. The "underlying rationale for the privilege has changed over time," but "courts long have viewed its central concern as one 'to encourage full and frank communication between attorneys and their clients and thereby promote broader public interests in the observance of law and administration of justice.'" *Zolin*, 491 U.S. at 562 (citation omitted). As the Supreme Court has recognized, "[t]he attorney-client privilege must necessarily protect the confidences of wrongdoers." *Id.* "[B]ut," the Court explains,

> the reason for that protection—the centrality of open client and attorney communication to the proper functioning of our adversary system of justice—ceas[es] to operate at a certain point, namely, where the desired advice refers *not to prior wrongdoing*, but to *future wrongdoing*.

Id. at 562-63 (internal quotes omitted) (emphasis in original).

It is precisely because the intentional destruction of documents impedes the proper functioning of our adversary system, operating as a fraud on the court and other litigants, that the D.C. Circuit has found that advice related to such conduct falls within the crime-fraud exception. *In re Sealed Case*, 754 F.2d 395. At issue in *In re Sealed Case* was an organization's "massive and systematic program to destroy and alter subpoenaed evidence or *evidence sought pursuant to civil discovery requests*." *Id*. at 397 (emphasis added)*; see also id*. at 400 ("deliberate and purposeful scheme to destroy extensive amounts of … discoverable materials") (internal quotes omitted). Finding the crime-fraud exception applicable to allegedly privileged documents related to that program, the court stated that the organization's "attempt to emasculate the court's ability to ascertain the truth necessarily strikes at the very foundations of the adversary system and the judicial process." *Id*. at 401. Indeed, the court deemed the intentional

destruction of documents to be "criminal and fraudulent activity" that "target[ed]" "the judicial process itself." *Id*. (document destruction was "an ongoing fraud in litigation").

The D.C. Circuit did note that "the grand jury's investigation into [the organization's] scheme of evidence destruction and concealment thereof involves possible *federal crimes*, not mere frauds between private litigants." *Id* (emphasis in original). The court gave every indication that "frauds between private litigants" would likewise justify piercing the privilege, however, because they also undermine the adversary system severely. The court wrote that the organization's "attempt to minimize the seriousness of its misconduct by characterizing it as incidental discovery abuse is inapposite," because "[m]isuse and abuse of the discovery process by litigants has been a matter of increasing concern to courts generally, and the deleterious effect on the judicial system cannot be overemphasized." *Id*. at 401 n.5.

That intentional spoliation in advance of or during litigation should be treated as fraud is also good policy. Fraud is defined as a "knowing misrepresentation of the truth" or "concealment of a material fact" "to induce another to act to his or her detriment." *Black's Law*

Dictionary 731 (9th ed. 2009). Intentional spoliation of discoverable materials, too, can stem from "'desire to suppress the truth'" and "'indicate[] fraud.'" *Gumbs v. Int'l Harvester, Inc.*, 718 F.2d 88, 96 (3d Cir. 1983) (citing 29 Am. Jur. 2d Evidence § 177). *See Lewy v. Remington Arms Co.*, 836 F.2d 1104, 1112 (8th Cir. 1988) (same); *Akiona v. United States*, 938 F.2d 158, 161 (9th Cir. 1991) (document destruction can indicate "bad faith"). The purpose of intentional spoliation is to skew the decision-making of courts and impede the adversarial method for getting to the truth. As the ALJ properly recognized, advice in connection with such conduct does not serve the aims of the attorney-client privilege and is not protected by it.

In its attempt to demonstrate error in the ALJ's decision on this point, Rambus cites only two cases: *Spalding Sports,* 203 F.3d 800*,* and *Silvestri v. General Motors Corp.*, 271 F.3d 583 (4th Cir. 2001). Neither dictates a different result. *Spalding Sports* involved no evidence of fraudulent intent. There, the patent owner asserted that the "invention record" was privileged, and the defendant invoked the crime-fraud exception. As evidence of fraud, the defendant argued that the patent owner had committed inequitable conduct by failing to cite a particular

prior art reference to the patent office. This Court noted that "inequitable conduct" "includes types of conduct less serious than 'knowing and willful' fraud," and held that "mere allegation of [the patent owner's] failure to cite a reference to the PTO will not suffice." *Spalding Sports*, 203 F.3d at 807-08 (internal quotes omitted). Here, however, the ALJ found "the evidence of Rambus' intention to destroy evidence for purposes of preparing the company for litigation overwhelming and the destruction was intentional and in bad faith." Ad180. Nothing in *Spalding Sports* undermines the ALJ's routine holding that intentional destruction of relevant documents in advance of litigation can support piercing the attorney-client privilege.

Silvestri v. General Motors Corp., 271 F.3d 583, meanwhile, fully supports the ALJ's holding. There, a plaintiff was allegedly injured by the failure of an air bag to deploy when he crashed into a utility pole. *Id.* at 586. The plaintiff, however, repaired the car without allowing the car manufacturer an opportunity to inspect the car. *Id.* at 587. The Fourth Circuit ruled that this spoliation of evidence warranted dismissal of the lawsuit. *Id.* at 592-95. Under the logic of *Silvestri*, Rambus's failure to preserve relevant documents warrants dismissal of

its claims. *Silvestri* provides no support for undermining the ALJ's far lesser sanction of piercing the attorney-client privilege in light of the improper document destruction.

Ultimately, Rambus wants Order 15 vacated because the ALJ concluded—incorrectly—that the spoliation was not actionable in this case. R.O.B. 63-65. Rambus's analysis confuses two separate questions—the showing necessary to pierce the attorney-client privilege under the crime-fraud exception *vs*. the showing necessary to demonstrate actionable spoliation—and fails as a result.

It is worth noting that a client need not actually have succeeded in his criminal or fraudulent scheme for the exception to apply; even in the event of a failed attempt at a crime or fraud, the privilege provides no protection. *In re Grand Jury Subpoena*, 731 F.2d 1032, 1039 (2d Cir. 1984). Rambus *did* succeed in its scheme, however, and there is accordingly no basis for vacating Order 15. As NVIDIA has demonstrated, N.O.B. 59-68, the record here permits only one conclusion: Rambus unlawfully destroyed documents in bad faith. Indeed, the ALJ found the evidence of "Rambus' intention to destroy evidence for purposes of preparing the company for litigation

overwhelming and that the destruction was intentional and in bad faith." Ad180. The ALJ declined to bar enforcement of the relevant patents on the basis of that spoliation due solely to an erroneous shifting of the burden to NVIDIA to show prejudice. *See* N.O.B. 60-61. Order 15 was correct when issued and subsequent events confirm that the privilege was properly pierced.

Finally, should the Court affirm the *Micron* ruling, the ALJ's decision to pierce the assertion of privilege would necessarily be correct. The *Micron* district court found Rambus's patents are unenforceable due to its bad-faith destruction of relevant documents. *See* N.O.B. 66. If the Court affirms that ruling, then the ALJ necessarily was correct to pierce the privilege under the lower "prima facie" standard.

CONCLUSION

The ITC's finding of no violation of Section 337 based on the Ware patents should be affirmed.

May 9, 2011

Respectfully submitted,

RUFFIN B. CORDELL
ANDREW R. KOPSIDAS
FISH & RICHARDSON P.C.
1425 K Street, NW, 11th Floor
Washington, DC 20005
Tel. No. 202-626-6449

Attorneys for Customer Appellants

MARK S. DAVIES
ORRICK, HERRINGTON
& SUTCLIFFE LLP
1152 15th Street, NW
Washington, DC 20005
Tel. No. 202-339-8400

JOSHUA ROSENKRANZ
ORRICK, HERRINGTON
& SUTCLIFFE LLP
51 West 52nd Street
New York, NY 10019

I. NEEL CHATTERJEE
ORRICK, HERRINGTON
& SUTCLIFFE LLP
1000 Marsh Road
Menlo Park, CA 94025

KAREN JOHNSON-MCKEWAN
ORRICK, HERRINGTON
& SUTCLIFFE LLP
405 Howard Street
San Francisco, CA 94105

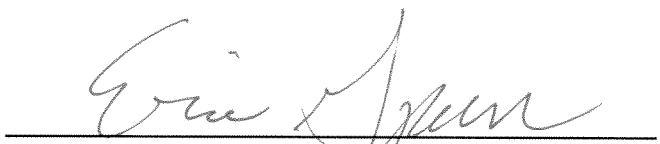
Attorneys for Appellant
NVIDIA Corporation

DECLARATION OF AUTHORITY OF ERIC GRUNSPAN

1. I am employee of the law firm of Orrick, Herrington & Sutcliffe LLP, counsel for Intervenor, NVIDIA Corporation, in the above-entitled action. I make this declaration pursuant to Federal Circuit Rule 47.3(d) and 28 U.S. Code Section 1746.

2. I am authorized to sign the foregoing Brief of Intervenors NVIDIA Corporation et al. and Certificate of Interest on behalf of Ruffin Cordell, attorney of record for Customer Intervenors in this action.

I declare under penalty of perjury that the foregoing is true and correct. Executed on May 9, 2011 in Washington, D.C.



Eric Grunspan

Addendum

Rambus, Inc. v. International Trade Commission, *et al.*
Fed. Cir. Appeal No. 2010-1483

Index to Addendum

JX-1

U 1705874

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

June 19, 2008

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF:

U.S. PATENT: *7,177,998*
ISSUE DATE: *February 13, 2007*

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

P. SWAIN
Certifying Officer

US007177998B2

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

(10) **Patent No.: US 7,177,998 B2**
(45) **Date of Patent: Feb. 13, 2007**

(54) **METHOD, SYSTEM AND MEMORY CONTROLLER UTILIZING ADJUSTABLE READ DATA DELAY SETTINGS**

(75) Inventors: **Frederick A. Ware**, Los Altos, CA (US); **Ely K. Tsern**, Los Altos, CA (US); **Richard E. Perego**, San Jose, CA (US); **Craig E. Hampel**, San Jose, CA (US)

(73) Assignee: **Rambus Inc.**, Los Altos, 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.

(21) Appl. No.: **11/335,029**

(22) Filed: **Jan. 18, 2006**

(65) **Prior Publication Data**

US 2006/0129776 A1 Jun. 15, 2006

Related U.S. Application Data

(63) Continuation of application No. 11/219,096, filed on Sep. 1, 2005, which is a continuation of application No. 11/094,137, filed on Mar. 31, 2005, which is a continuation of application No. 10/732,533, filed on Dec. 11, 2003, which is a continuation of application No. 09/841,911, filed on Apr. 24, 2001, now Pat. No. 6,675,272.

(51) **Int. Cl.**
G06F 13/20 (2006.01)

(52) **U.S. Cl.** **711/167**; 711/168; 711/169

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,950,735 A 4/1976 Patel
4,183,095 A 1/1980 Ward
4,280,221 A 7/1981 Chun et al.
4,315,308 A 2/1982 Jackson
4,330,852 A 5/1982 Redwine et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 0 379 772 A 8/1990

(Continued)

OTHER PUBLICATIONS

Rambus, Inc. Data Sheet, Preliminary Information, 8/9-Mbit (1M x 8/9) and 16/18-Mbit (2M x 8/9) RDRAM, Mar. 1996, pp. 1-30.

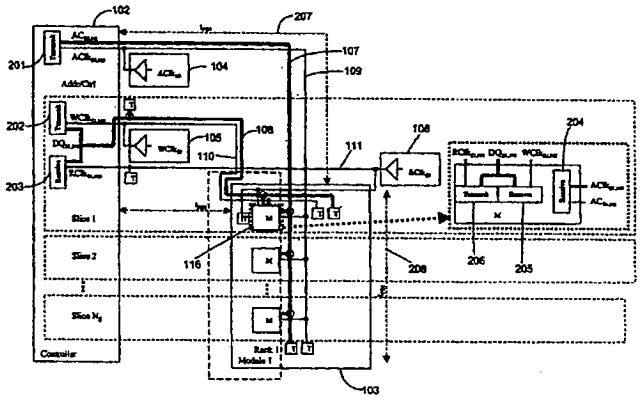
(Continued)

Primary Examiner—T Nguyen
(74) *Attorney, Agent, or Firm*—Shemwell Mahamedi LLP

(57) **ABSTRACT**

A method, system and memory controller that uses adjustable read data delay settings. The memory controller includes control transmit circuitry, data reception circuitry and timing circuitry. The control circuitry transmits a control signal to multiple memory devices via a shared control signal path. The data reception circuitry receives data signals from the memory devices via respective data signal paths. The timing circuitry delays reception of data signals on each of the data signal paths by a respective time interval that is based, at least in part, on a time required for the control signal to propagate on the control signal path from the memory controller to a respective one of the memory devices.

24 Claims, 34 Drawing Sheets



U.S. PATENT DOCUMENTS

4,337,523	A	6/1982	Hotta et al.	
4,445,204	A	4/1984	Nishiguchi	
4,499,536	A	2/1985	Gemma et al.	
4,567,545	A	1/1986	Mettler, Jr.	
4,637,018	A	1/1987	Flora et al.	
4,646,270	A	2/1987	Voss	
4,712,190	A	12/1987	Guglielmi et al.	
4,719,602	A	1/1988	Hag et al.	
4,755,937	A	7/1988	Glier	
4,763,249	A	8/1988	Bomba et al.	
4,792,926	A	12/1988	Roberts	
4,792,929	A	12/1988	Olson et al.	
4,799,199	A	1/1989	Scales, III et al.	
4,800,530	A	1/1989	Itoh et al.	
4,821,226	A	4/1989	Christopher et al.	
4,825,411	A	4/1989	Hamano	
4,845,664	A	7/1989	Aichelmann, Jr. et al.	
4,845,677	A	7/1989	Chappell et al.	
4,849,937	A	7/1989	Yoshimoto	
4,866,675	A	9/1989	Kawashima	
4,866,676	A	9/1989	Crisp et al.	
4,866,677	A	9/1989	Sakurai	
4,866,678	A	9/1989	Pinkham et al.	
4,866,679	A	9/1989	Chambers	
4,866,680	A	9/1989	Scherbatskoy	
4,928,265	A	5/1990	Higuchi et al.	
4,937,734	A	6/1990	Bechtolsheim	
4,945,516	A	7/1990	Kashiyama	
4,953,128	A	8/1990	Kawai et al.	
5,001,672	A	3/1991	Ebbers et al.	
5,077,693	A	12/1991	Hardee et al.	
5,083,296	A	1/1992	Hara et al.	
5,111,386	A	5/1992	Fujishima et al.	
5,117,389	A	5/1992	Yiu	
5,124,589	A	6/1992	Shiomi et al.	
5,140,688	A	8/1992	White et al.	
5,179,687	A	1/1993	Hidaka et al.	
5,260,905	A	11/1993	Mori	
5,276,858	A	1/1994	Oak et al.	
5,301,278	A	4/1994	Bowater et al.	
5,305,278	A	4/1994	Inoue	
5,311,483	A	5/1994	Takasugi	
5,319,755	A	6/1994	Farmwald et al.	
5,323,358	A	6/1994	Toda et al.	
5,327,390	A	7/1994	Takasugi	
5,339,276	A	8/1994	Takasugi	
5,341,341	A	8/1994	Fukuzo	
5,345,573	A	9/1994	Bowden, III et al.	
5,365,489	A	11/1994	Jeong	
5,379,438	A	1/1995	Bell et al.	
5,381,376	A	1/1995	Kim et al.	
5,381,538	A	1/1995	Amini et al.	
5,384,745	A	1/1995	Konishi et al.	
5,386,385	A	1/1995	Stephens et al.	
5,390,149	A	2/1995	Vogley et al.	
5,392,239	A	2/1995	Margulis et al.	
5,404,338	A	4/1995	Murai et al.	
5,404,463	A	4/1995	McGarvey	
5,444,667	A	8/1995	Obara	
5,455,803	A	10/1995	Kodama	
5,475,690	A	12/1995	Burns et al.	
5,504,874	A	4/1996	Galles et al.	
5,511,024	A	4/1996	Ware et al.	
5,533,204	A	7/1996	Tipley	
5,548,788	A	8/1996	Amini et al.	
5,553,248	A	9/1996	Melo et al.	
5,578,940	A	11/1996	Dillon et al.	
5,606,717	A	2/1997	Farmwald et al.	
5,611,058	A	3/1997	Moore et al.	
5,638,531	A	6/1997	Crump et al.	
5,649,161	A	7/1997	Andrade et al.	
5,655,113	A	8/1997	Leung et al.	
5,663,661	A	9/1997	Dillon et al.	
5,680,361	A	10/1997	Ware et al.	
5,708,297	A	1/1998	Clayton	
5,742,798	A	4/1998	Goldrian	
5,748,914	A	5/1998	Barth et al.	
5,764,963	A	6/1998	Ware et al.	
5,778,419	A	7/1998	Hansen et al.	
5,796,624	A	8/1998	Sridhar et al.	
5,844,855	A	12/1998	Ware et al.	
5,867,541	A	2/1999	Tanaka et al.	
5,892,981	A	4/1999	Wiggers et al.	
5,928,343	A	7/1999	Farmwald et al.	
5,933,379	A	8/1999	Park et al.	
5,943,573	A	8/1999	Wen	
5,952,691	A	9/1999	Yamaguchi	
5,987,576	A *	11/1999	Johnson et al.	711/167
6,005,776	A	12/1999	Holman et al.	
6,016,282	A	1/2000	Keeth	
6,034,878	A	3/2000	Osaka et al.	
6,049,467	A	4/2000	Tamarkin et al.	
6,065,092	A	5/2000	Roy	
6,067,594	A	5/2000	Perino et al.	
6,075,730	A	6/2000	Barth et al.	
6,111,757	A	8/2000	Dell et al.	
6,125,419	A	9/2000	Umemura et al.	
6,154,417	A	11/2000	Kim et al.	
6,154,821	A	11/2000	Barth et al.	
6,160,754	A	12/2000	Suh et al.	
6,172,895	B1	1/2001	Brown et al.	
6,185,644	B1	2/2001	Farmwald et al.	
6,233,157	B1	5/2001	Yoon et al.	
6,253,266	B1	6/2001	Ohanian	
6,260,097	B1	7/2001	Farmwald et al.	
6,266,285	B1	7/2001	Farmwald et al.	
6,266,730	B1	7/2001	Perino et al.	
6,266,737	B1	7/2001	Ware et al.	
6,292,877	B1	9/2001	Ryan	
6,304,937	B1	10/2001	Farmwald et al.	
6,314,051	B1	11/2001	Farmwald et al.	
6,321,282	B1	11/2001	Horowitz et al.	
6,336,205	B1	1/2002	Kurokawa et al.	
6,343,352	B1	1/2002	Davis et al.	
6,356,260	B1	3/2002	Montalbo	
6,359,815	B1 *	3/2002	Sato et al.	365/198
6,388,886	B1	5/2002	Tobita	
6,388,934	B1	5/2002	Tobita	
6,401,167	B1	6/2002	Barth et al.	
6,449,159	B1	9/2002	Haba	
6,449,727	B1	9/2002	Toda	
6,477,592	B1	11/2002	Chen et al.	
6,493,789	B2	12/2002	Ware et al.	
6,496,897	B2	12/2002	Ware et al.	
6,502,161	B1	12/2002	Perego et al.	
6,516,365	B2	2/2003	Horowitz et al.	
6,526,469	B1	2/2003	Drehmel et al.	
6,539,454	B2	3/2003	Mes	
6,545,875	B1	4/2003	Perino et al.	
6,590,781	B2	7/2003	Kollipara et al.	
6,618,736	B1	9/2003	Menage	
6,640,292	B1	10/2003	Barth et al.	
6,643,752	B1	11/2003	Donnelly et al.	
6,643,787	B1	11/2003	Zerbe et al.	
6,646,953	B1 *	11/2003	Stark	365/233
6,657,871	B2	12/2003	Perino et al.	
6,675,272	B2	1/2004	Ware et al.	
6,680,866	B2	1/2004	Kajiimoto	
6,681,288	B2	1/2004	Ware et al.	
6,684,263	B2	1/2004	Horowitz et al.	
6,760,857	B1	7/2004	Lau et al.	
6,765,800	B2	7/2004	Haba et al.	
6,833,984	B1	12/2004	Belgacem	
6,839,266	B1	1/2005	Garrett, Jr. et al.	

6,839,393	B1	1/2005	Sidiropoulos	
6,853,557	B1	2/2005	Haba et al.	
6,873,939	B1	3/2005	Zerbe et al.	
6,898,085	B2	5/2005	Haba et al.	
6,912,680	B1 *	6/2005	Keeth	714/736
6,950,956	B2	9/2005	Zerbe et al.	
2001/0047450	A1	11/2001	Gillingham et al.	
2002/0174311	A1	11/2002	Ware et al.	
2004/0003194	A1	1/2004	Bodas et al.	
2004/0054845	A1	3/2004	Ware et al.	
2004/0170072	A1	9/2004	Ware et al.	
2005/0169097	A1	8/2005	Ware et al.	
2006/0007761	A1	1/2006	Ware et al.	
2006/0039174	A1	2/2006	Ware et al.	
2006/0056244	A1	3/2006	Ware et al.	
2006/0069895	A1	3/2006	Ware et al.	
2006/0077731	A1	4/2006	Ware et al.	

FOREIGN PATENT DOCUMENTS

EP	0 709 786	A	5/1996
EP	0735492		10/1996
JP	200035831	A	2/2000
JP	200035831	A	2/2000
WO	WO99/41667		8/1999
WO	WO 99/50852	A1	10/1999
WO	WO 00/54164	A1	9/2000

OTHER PUBLICATIONS

Samsung Electronics, SDRAM Module, Preliminary KMM377S1620CT2, Rev. 1, Nov. 1998, 12 pages.

IBM, 184 Pin DIMM Design Updates/Ramifications for Unbuffered and Registered DDR DIMMs, JC-42.5, Dec. 1999, pp. 1-12.

Yasunobu Nakase, et al., Source-Synchronization and Timing Vernier Techniques for 1.2-GB/s SLDRAM Interface, IEEE Journal of Solid-State Circuits, vol. 34, No. 4, Apr. 1999, pp. 494-501.

IBM, Micron Technology, and Reliance Computer Corporation, DDR SDRAM Registered DIMM Design Specification, Revision 0.6, Nov. 1999, 62 pages.

Rambus, Inc., Direct Rambus Short Channel Layout Guide, Version 0.8, Mar. 2000, 33 pages.

Lluis Paris, et al., WP 24.3 A 800MB/s 72 Mb SLDRAM with Digitally-Calibrated DLL, 1999 IEEE International Solid-State Circuits Conference, 0-7803-5129-0/99, 10 pages.

Mike Seibert, DRAM Memory Enabling, Micron Technology, Inc., Competitive DDR Memory Sub-systems, Platform Conference, Jul. 18-19, 2000, 67 pages.

IEEE Standard for Scalable Coherent Interface (SCI), IEEE Std 1596-1992, 270 pages.

John Poulton, Signaling in High-Performance Memory Systems, ISSCC 1999, slides 1-59 on 30 pages.

Rambus, Inc. Direct RAC Data Sheet, Advance Information, Aug. 7, 1998, pp. 1-47.

IBM, Corp., Application Note, Direct Rambus Memory System Overview, Mar. 30, 1998, pp. 1-5.

Rambus, Inc., Rambus RIMM Module (with 64/72Mb RDRAMs) Data Sheet, Preliminary Information, Document DL0078 Version 0.90, Mar. 1999, pp. 1-11.

Rambus, Inc., Rambus RIMM Connector, Document DL 0069 Version 1.01, Jul. 1999, 14 pages.

Y.R. Kim, Memory Product Planning and Application, Samsung Electronics, DDR, Today and Tomorrow, Platform Conference, Jul. 18-19, 2000, 26 pages.

Peter Gillingham, MOSAID Technologies, Inc. SLDRAM Architectural and Functional Overview, SLDRAM Consortium, Aug. 29, 1997, 13 pages.

SLDRAM Inc., 4M x 18 SLDRAM, Draft/Advance, Jul. 9, 1998, pp. 1-69.

Peter Gillingham, MOSAID Technologies, Inc. SLDRAM Architectural and Functional Overview, SLDRAM Consortium, Aug. 29, 1997, 14 pages.

Peter Gillingham, MOSAID Technologies and Bill Vogley, Texas Instruments, SLDRAM: High-Performance, Open-Standard Memory, IEEE Micro, Nov./Dec. 1997, pp. 29-39.

Rambus, Inc., Rambus SO-RIMM Module (with 128/144Mb RDRAMs), Advance Information, Document DL0076 Version 0.7, Feb. 2000, pp. 1-12.

IEEE Draft Standard for a High-Speed Memory Interface (SyncLink), IEEE Draft 0.99 IEEE P 1596.7-199X, 1996, 66 pages.

Changsik Yoo, DRAM Design 3, Samsung Electronics, High Speed DRAM Interface, 35 pages.

Wong, D., et al., "Inserting Active Delay Elements to Achieve Wave Pipelining," IEEE 1989, pp. 270-273.

European Search Report and Written Opinion in European Patent Application 02009032.0-2212-, search completed Jun. 27, 2005, 9 pgs.

European Search Report and Written Opinion for EP 1 653374 A3 (Application No. 05022021.9) , Jul. 26, 2006, 6 pgs.

* cited by examiner

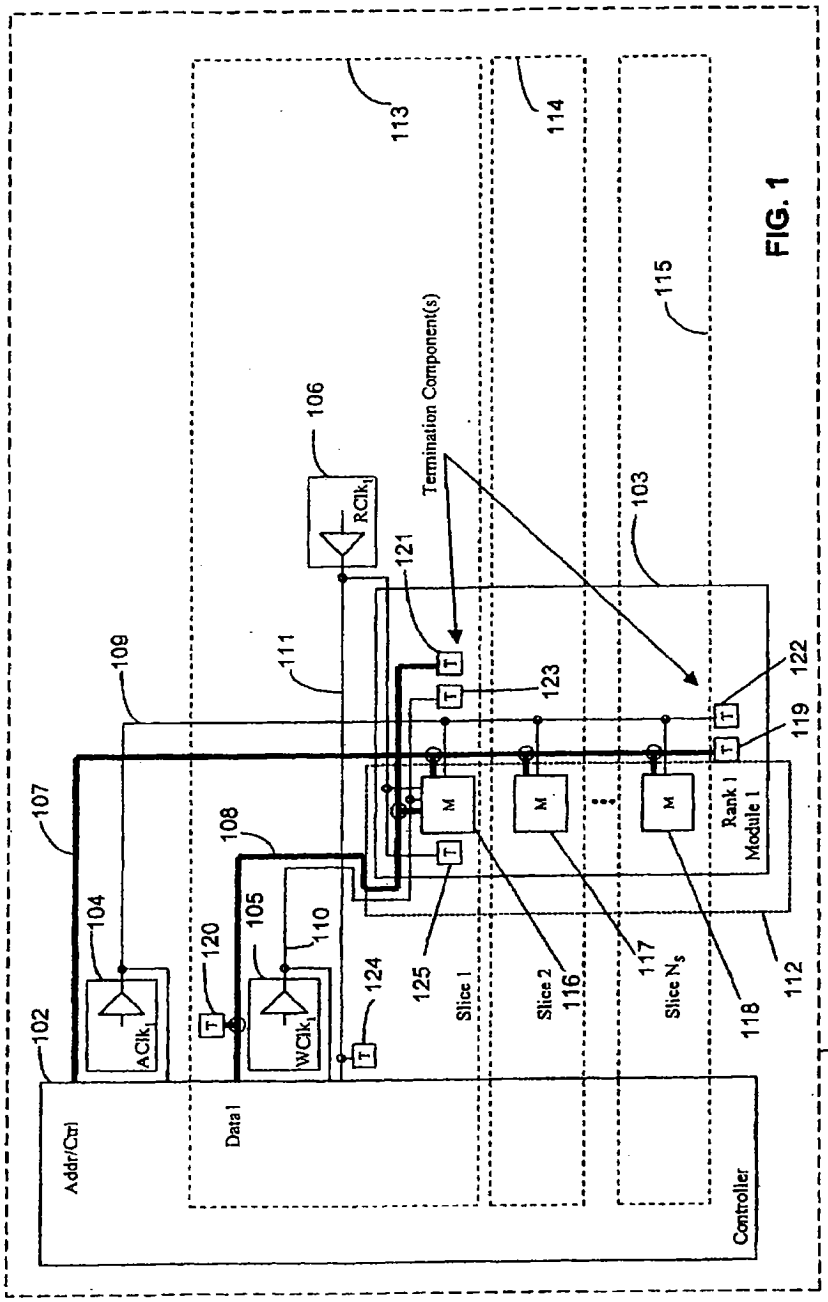


FIG. 1

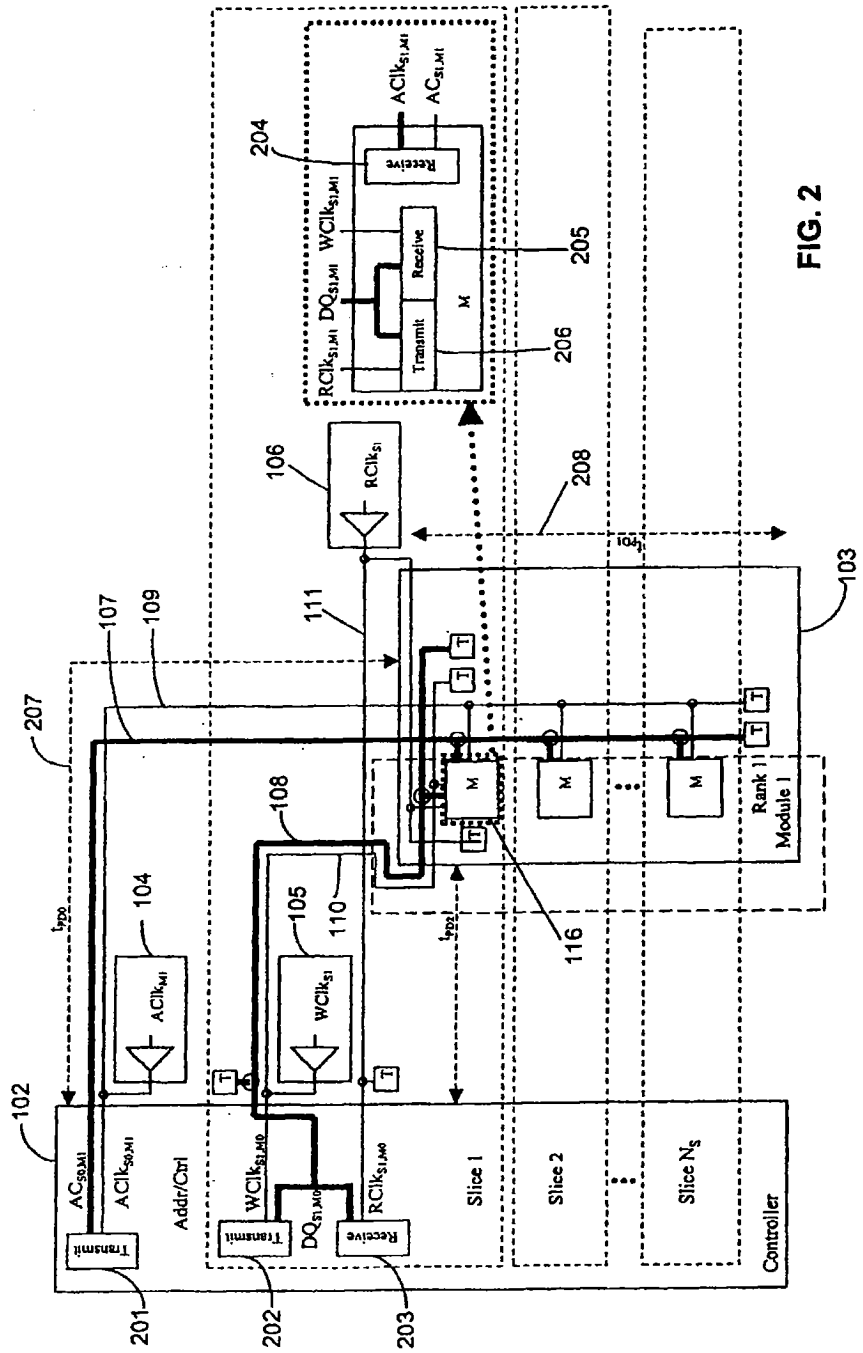


FIG. 2

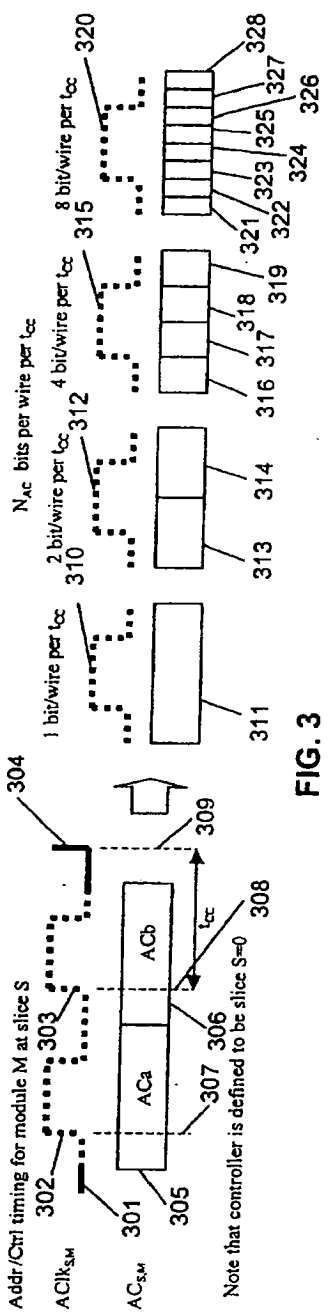


FIG. 3

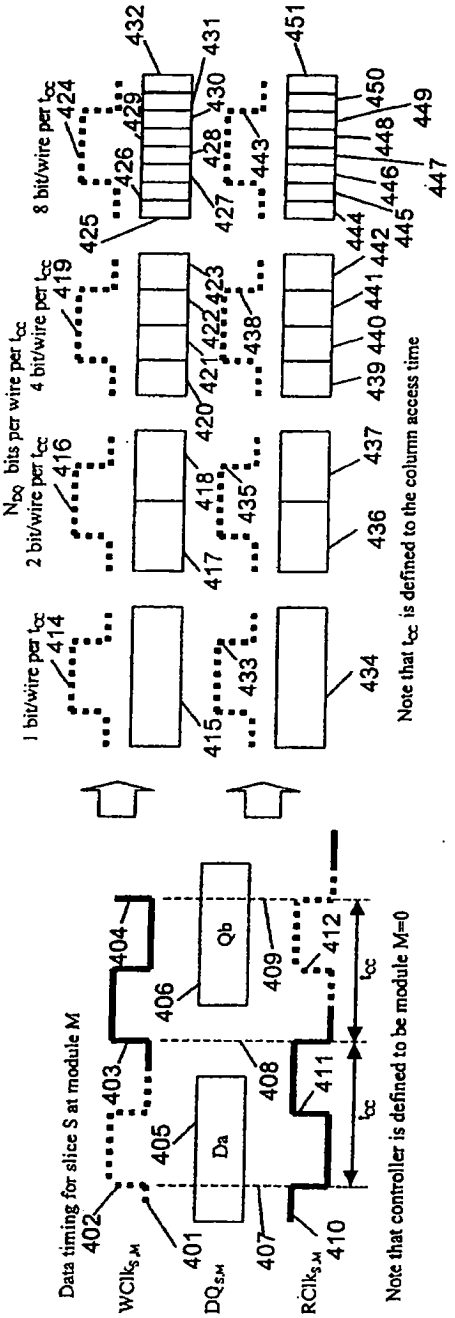


FIG. 4

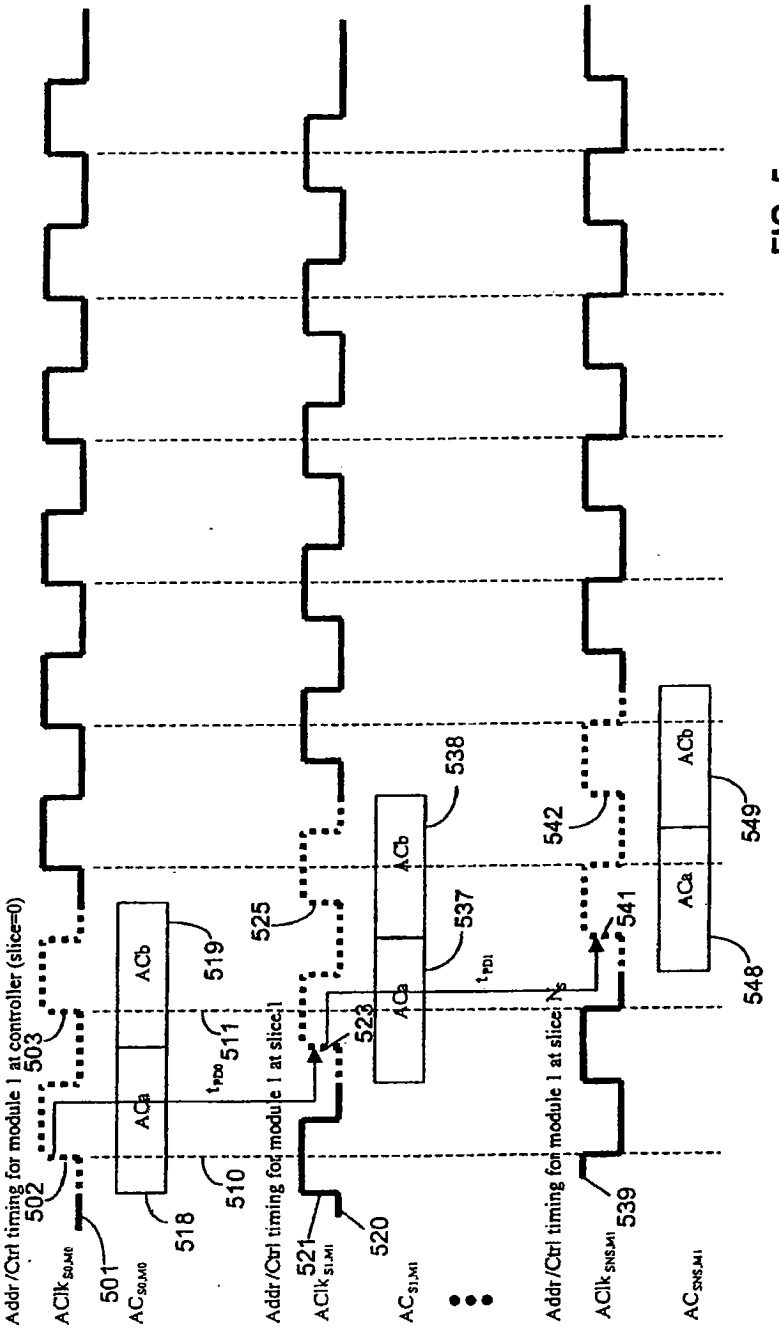


FIG. 5

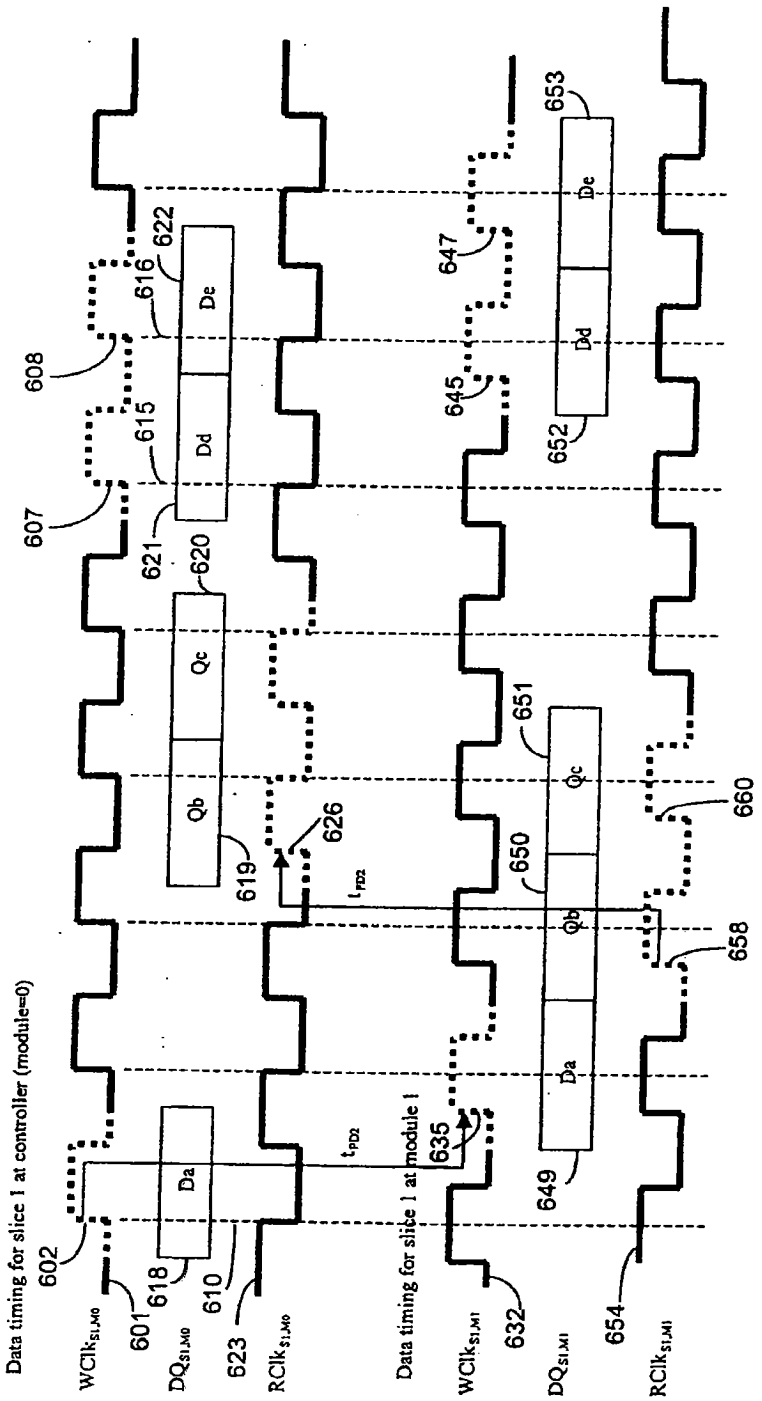


FIG. 6

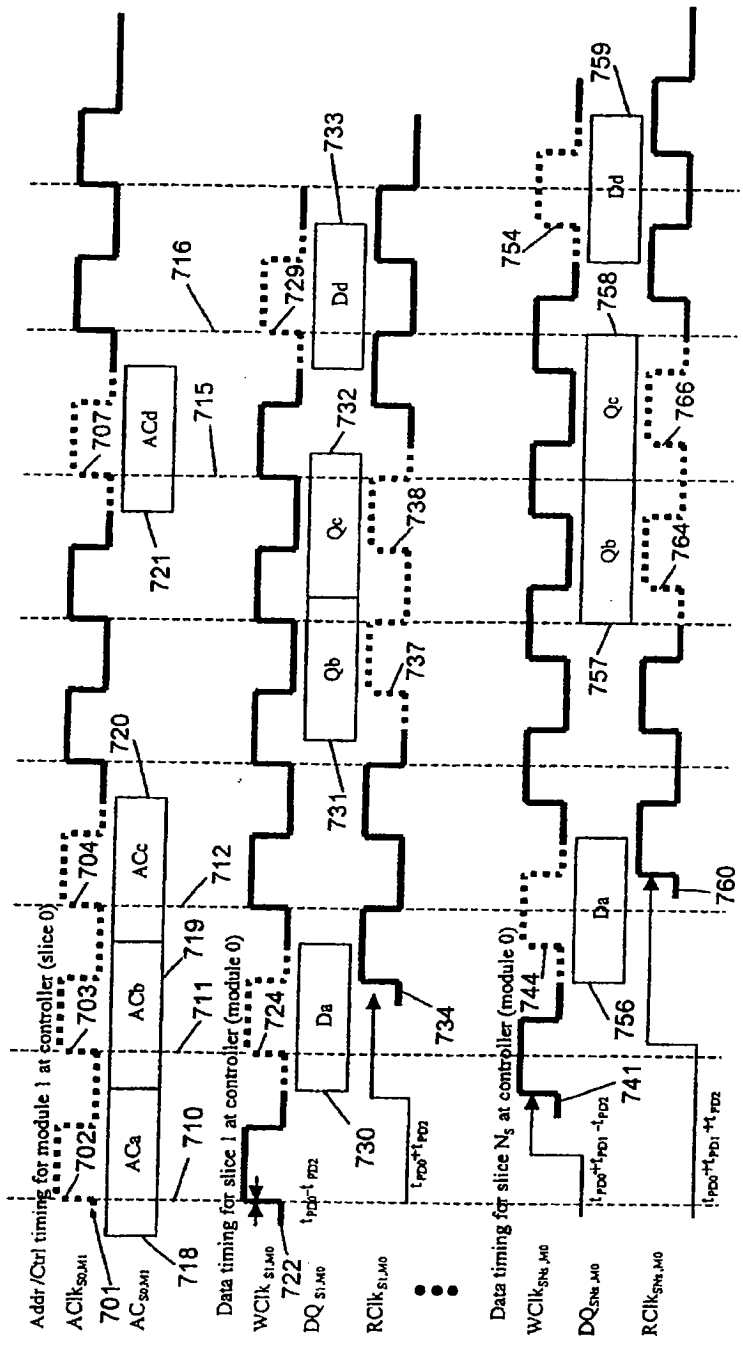


FIG. 7

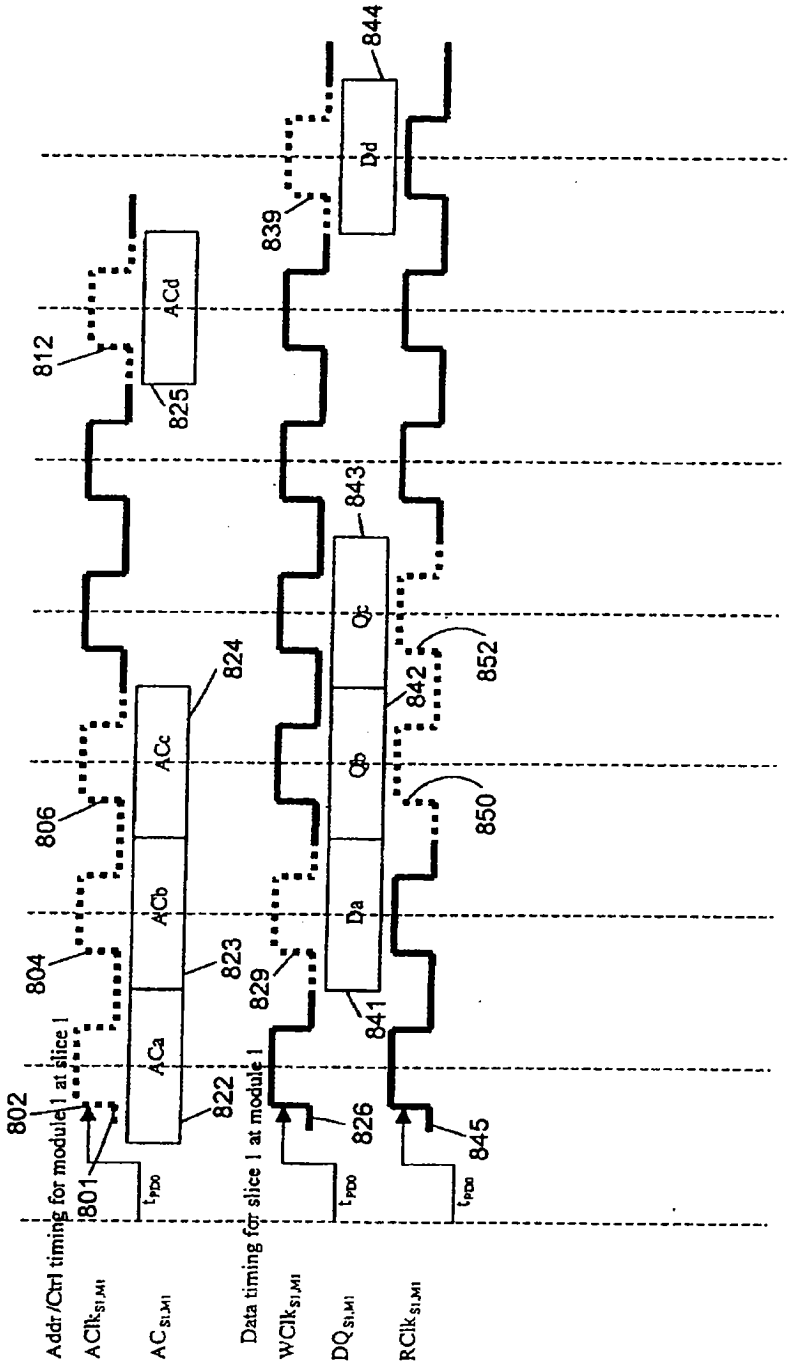


FIG. 8

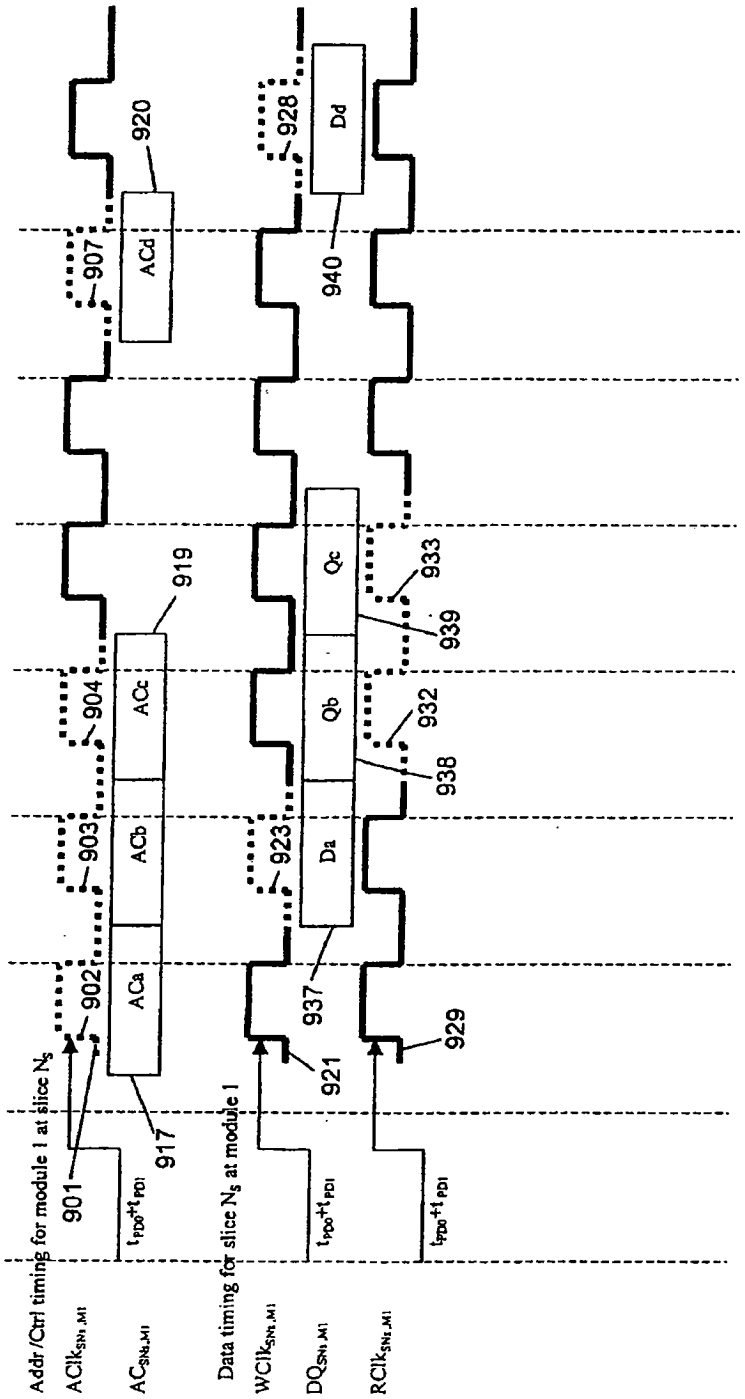


FIG. 9

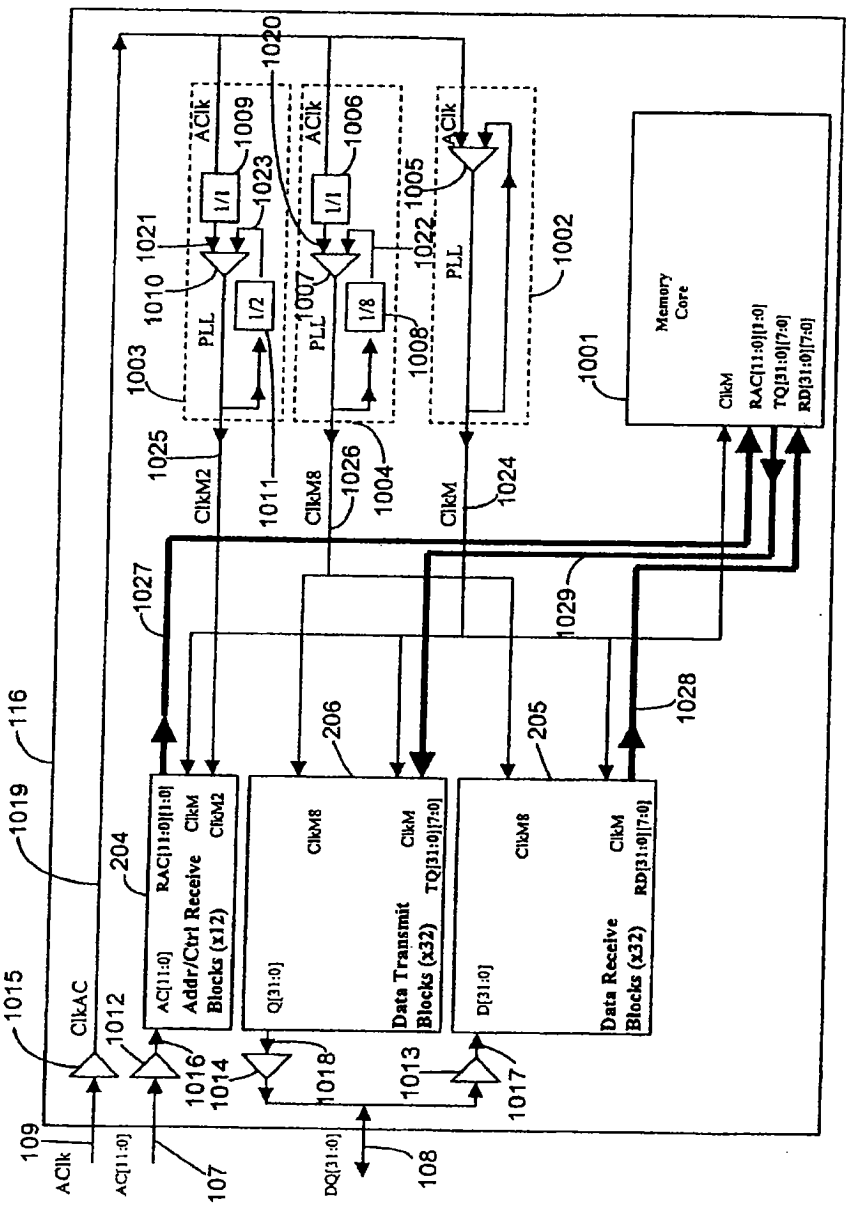


FIG. 10

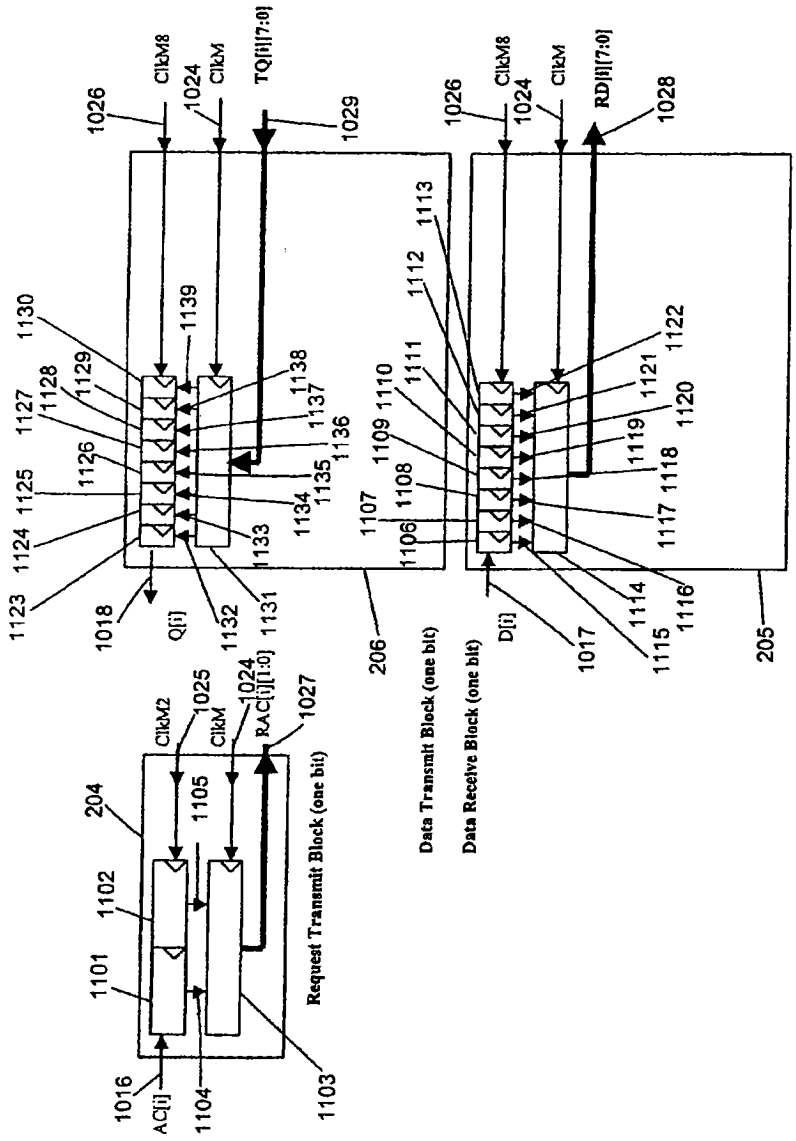


FIG. 11

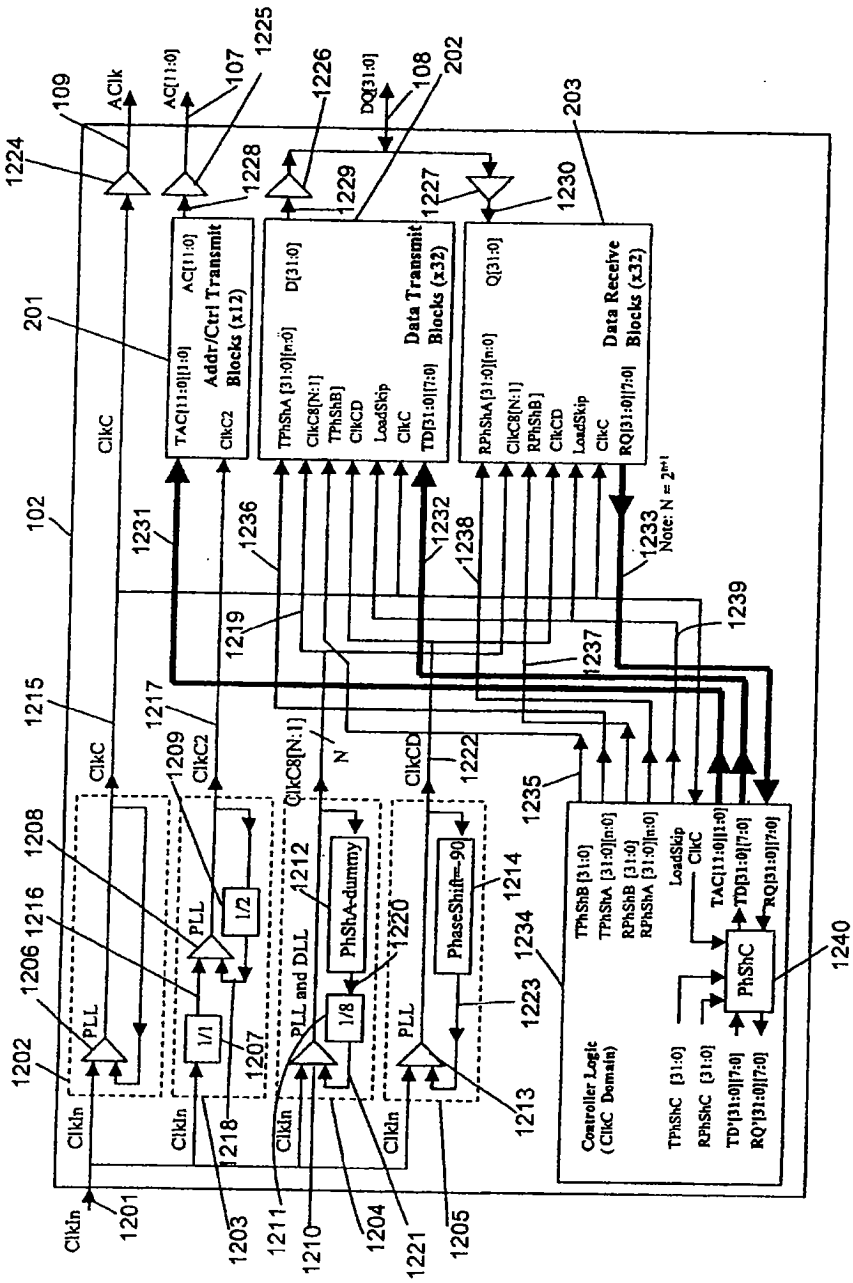


FIG. 12

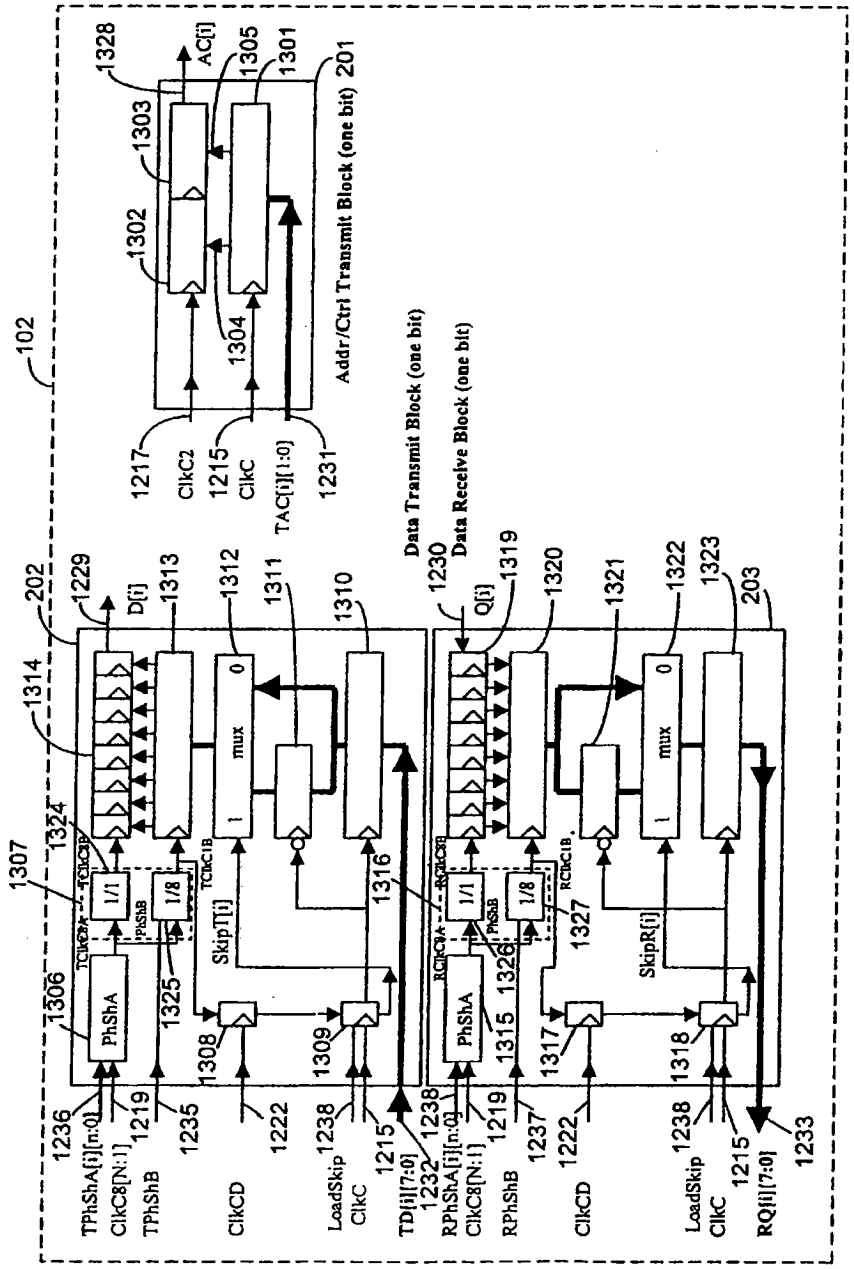


FIG. 13

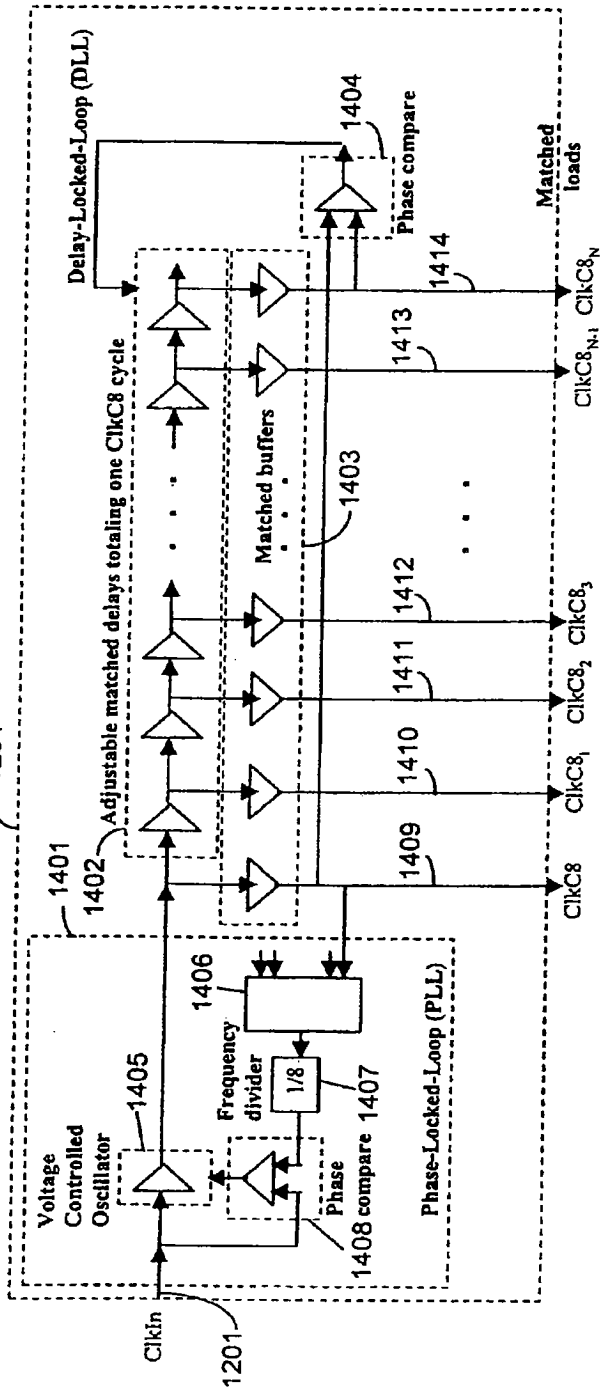


FIG. 14

FIG. 15

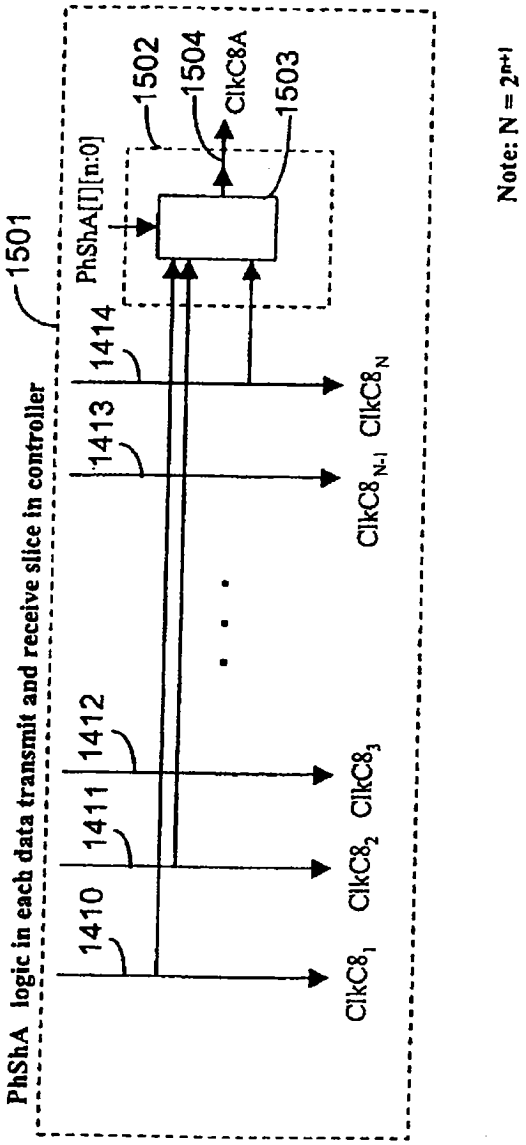
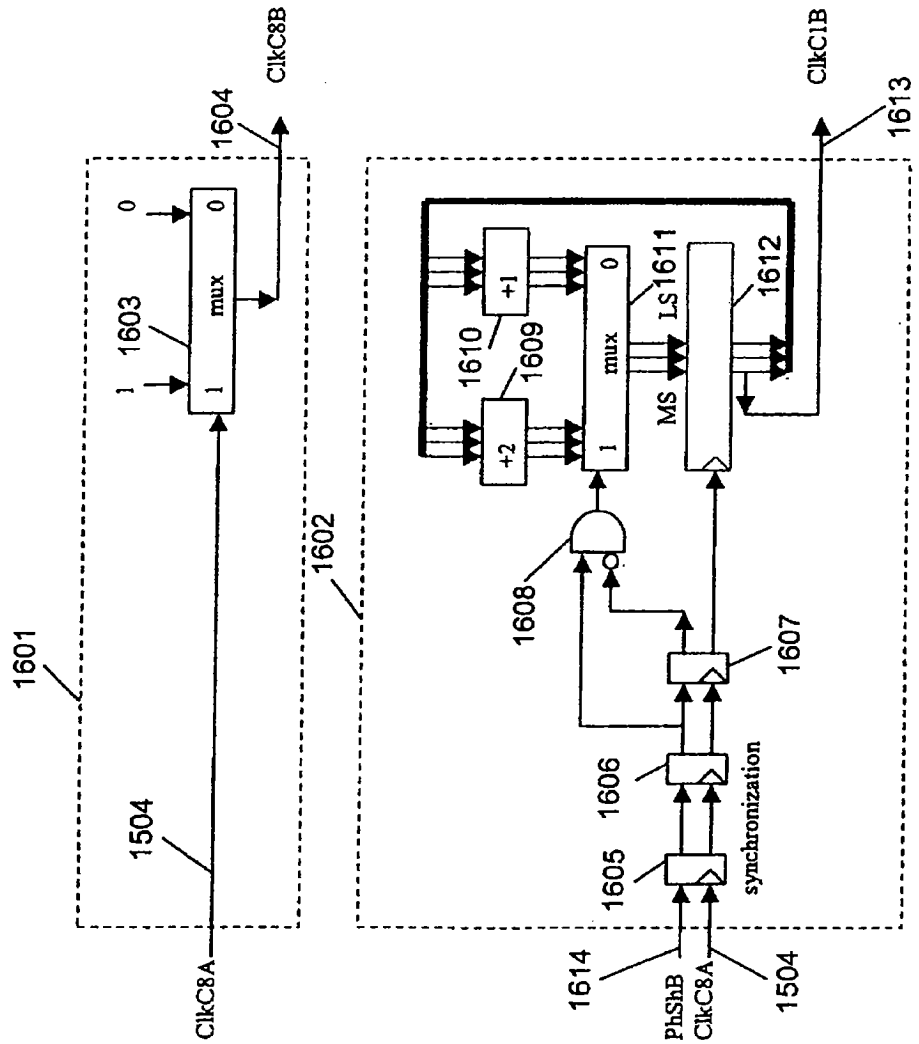


FIG. 16



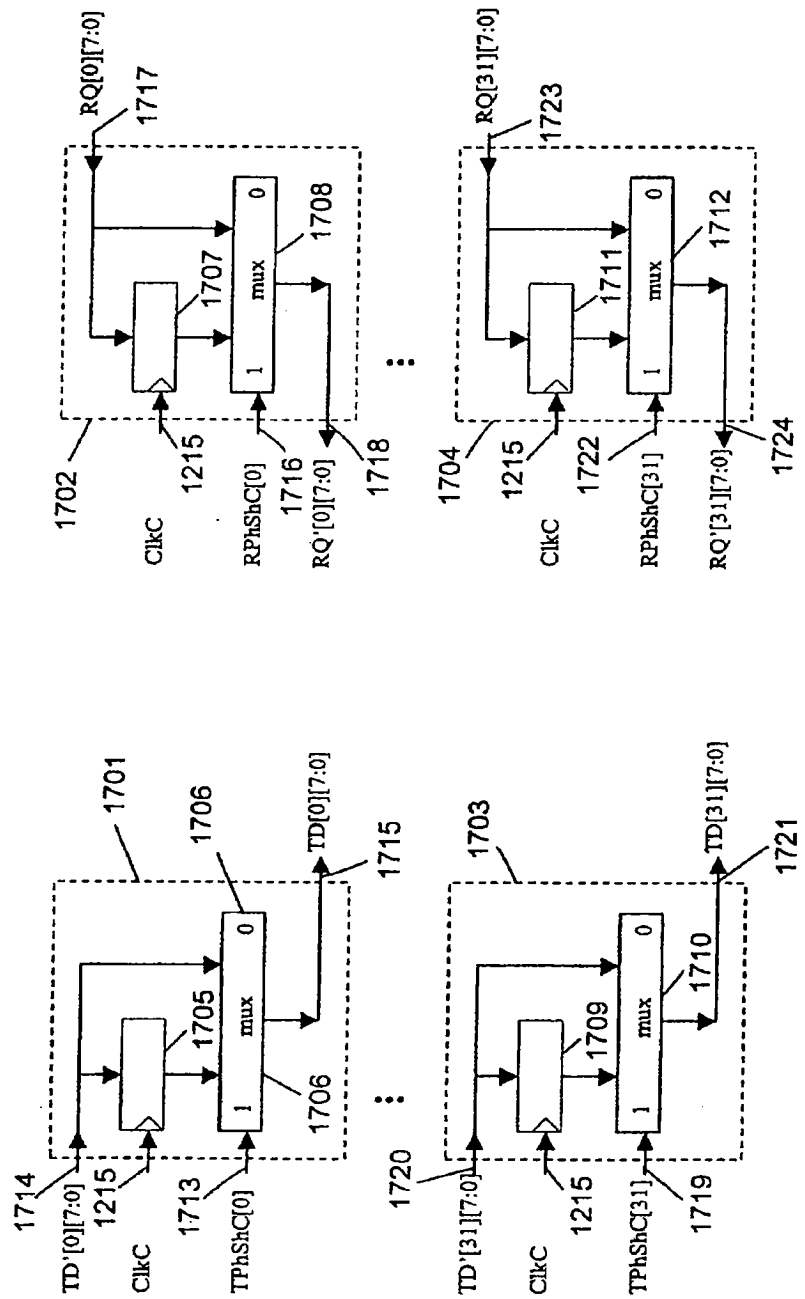


FIG. 17

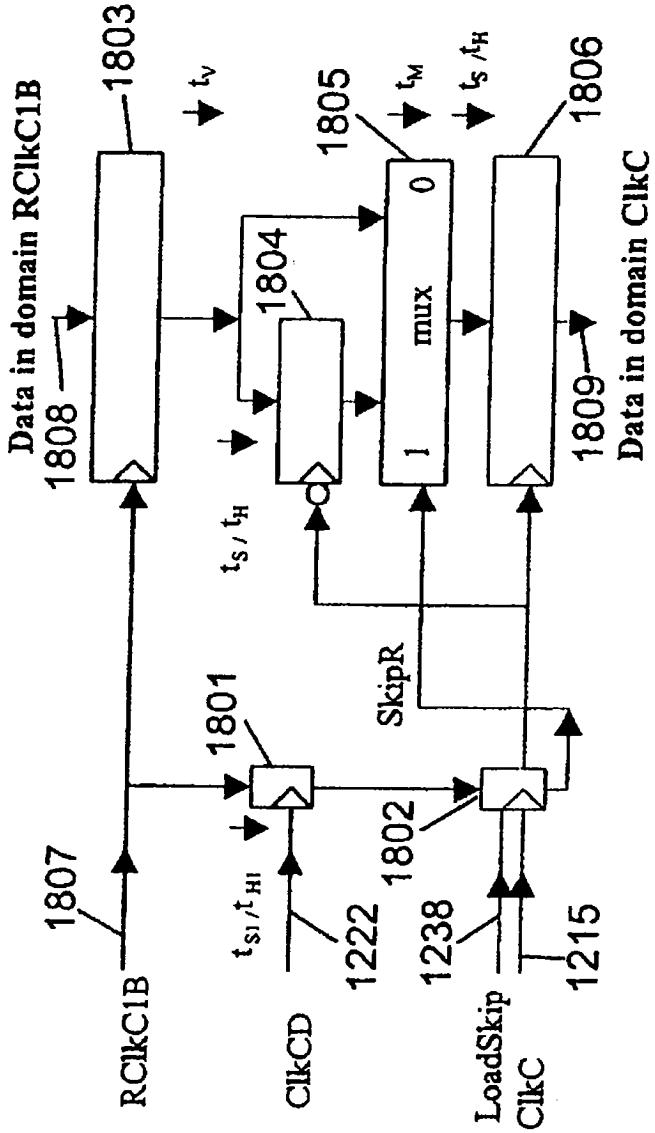


FIG. 18

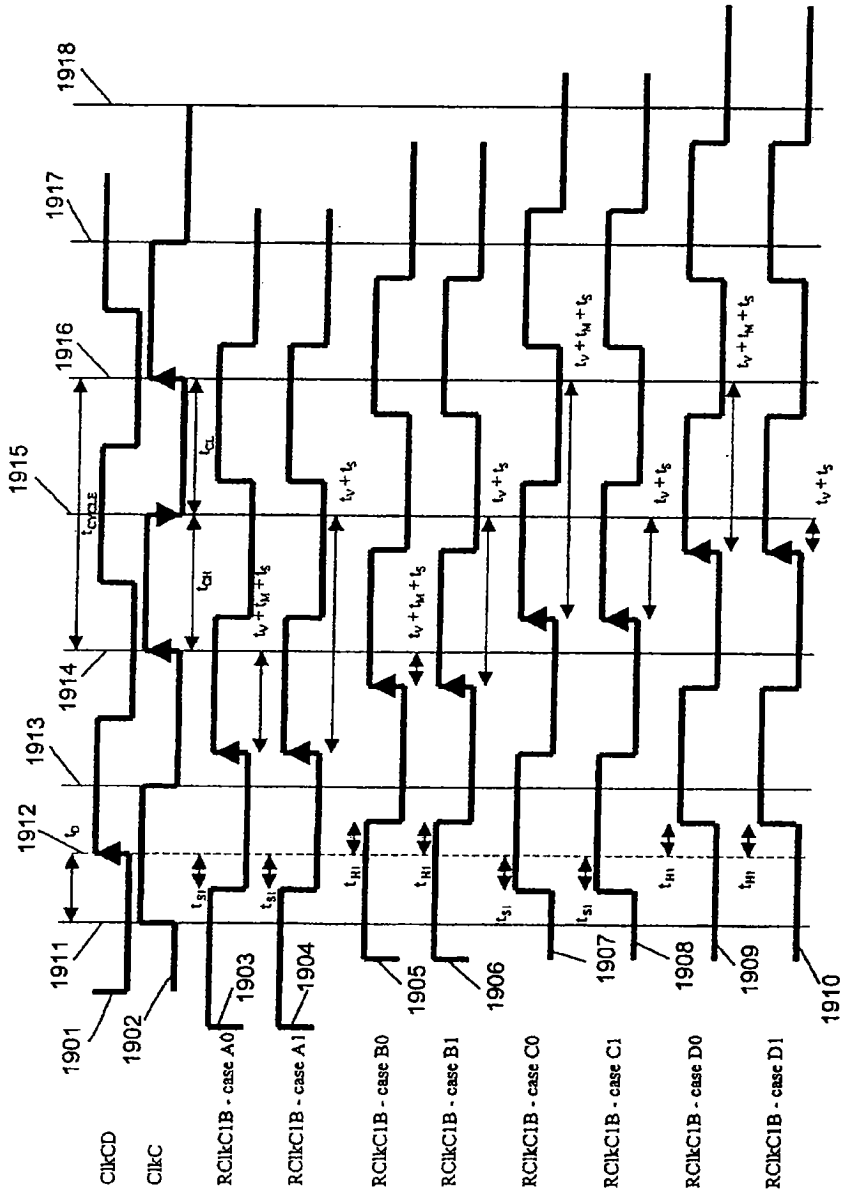


FIG. 19

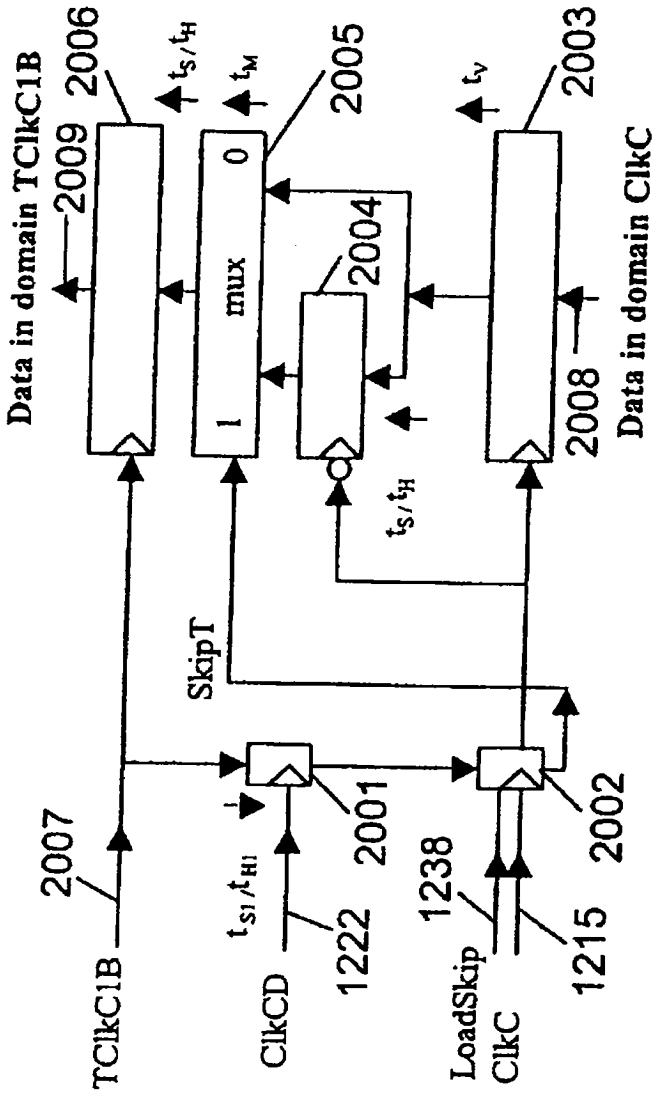


FIG. 20

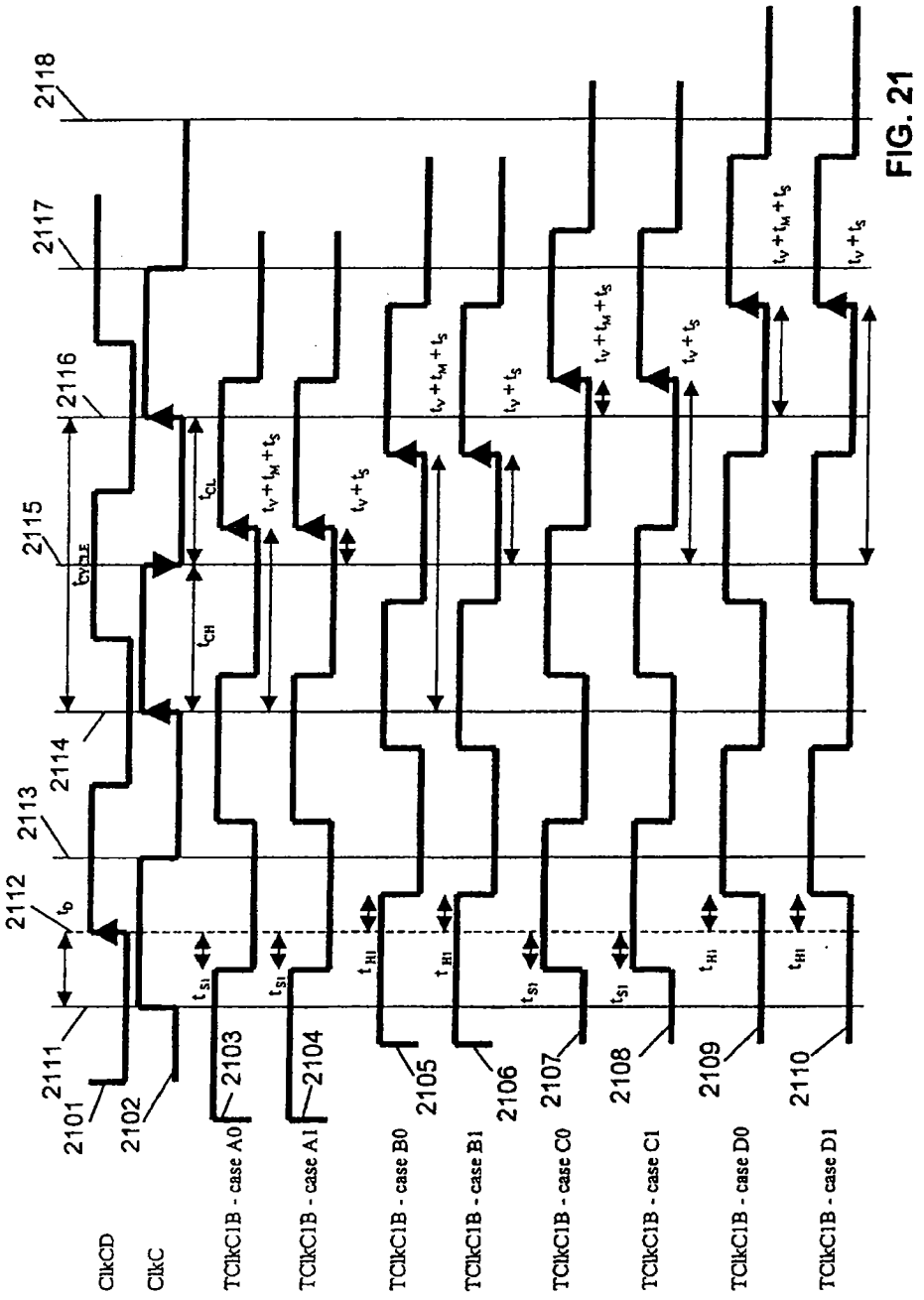


FIG. 21

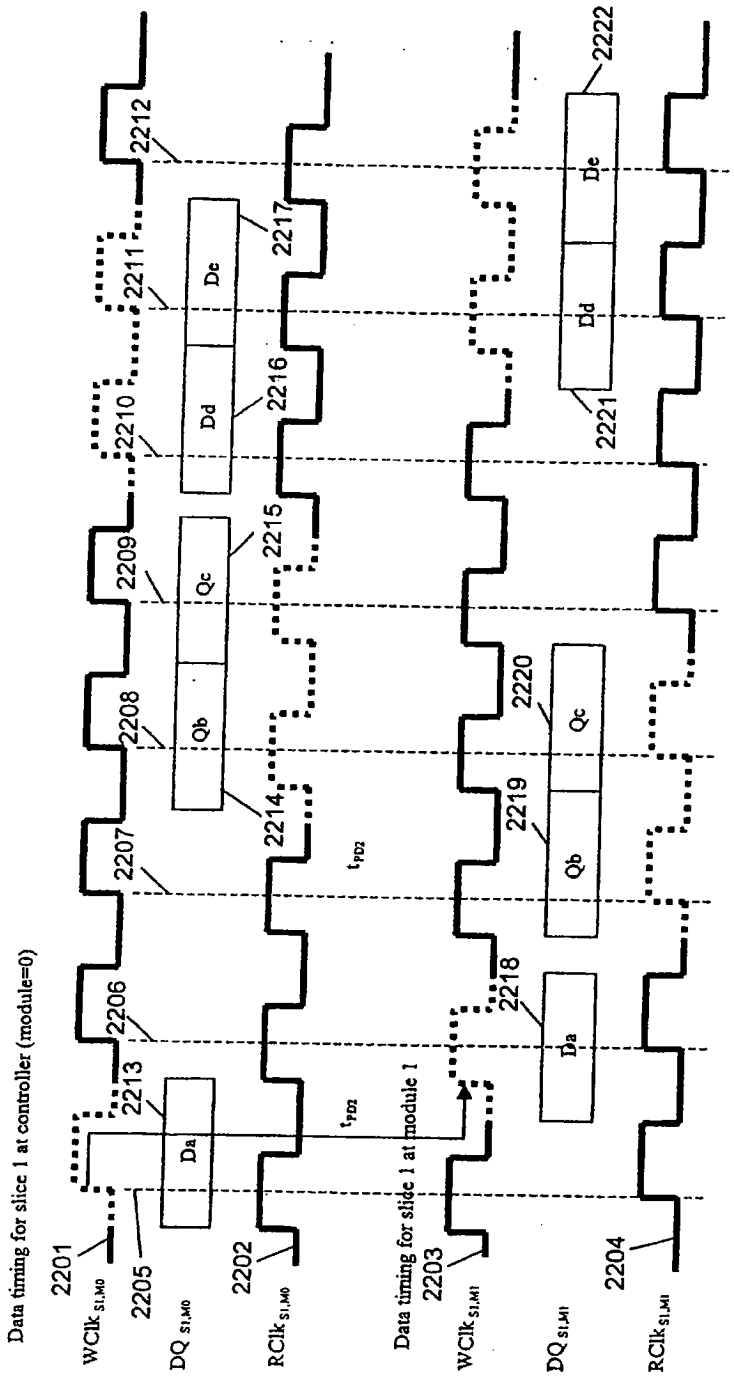


FIG. 22

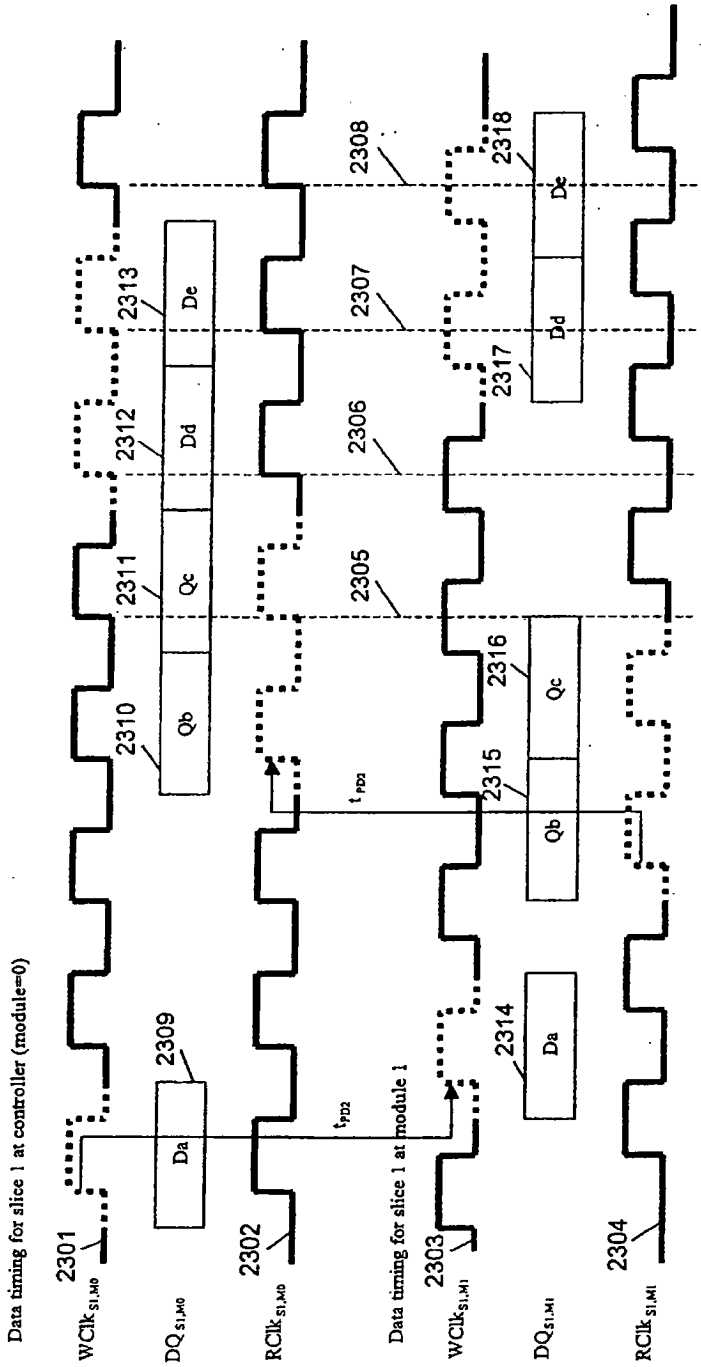


FIG. 23

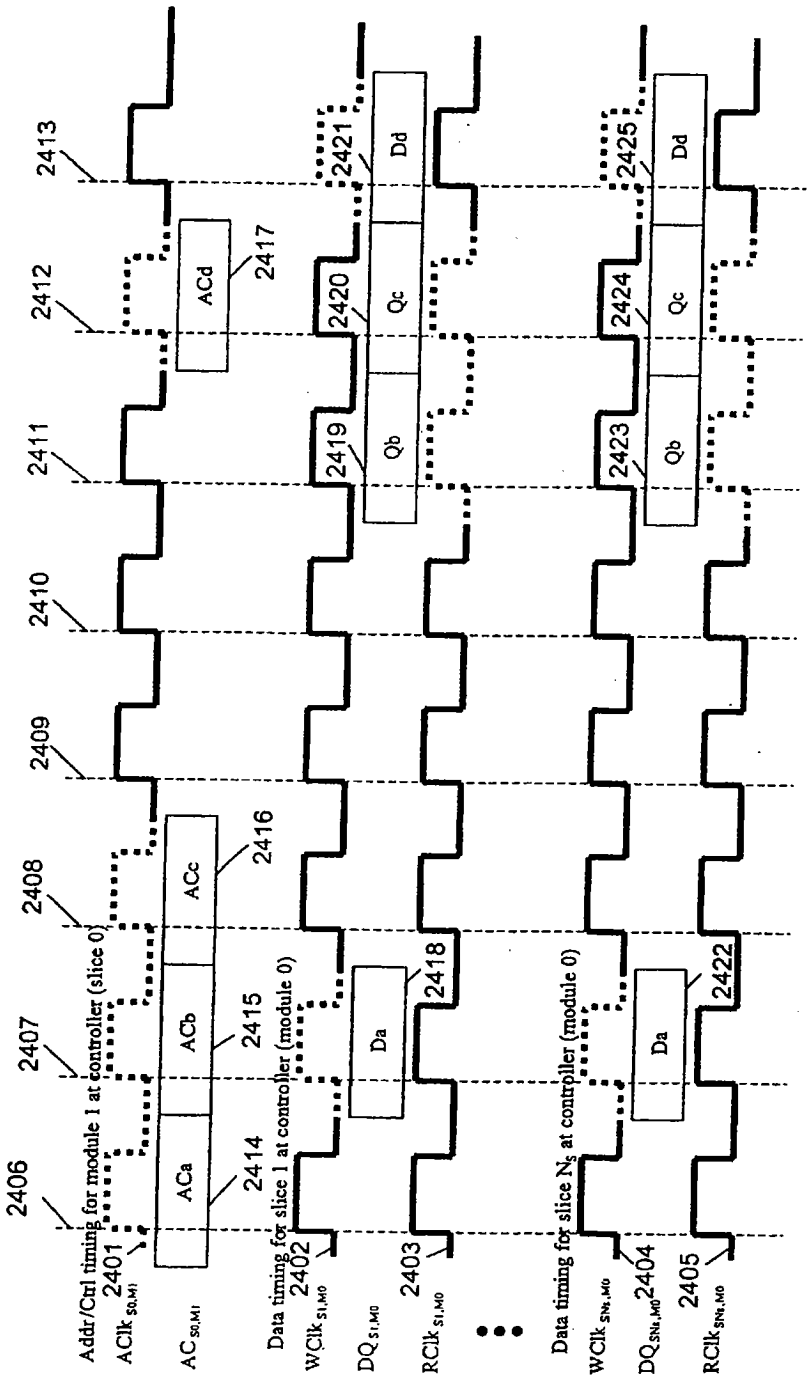


FIG. 24

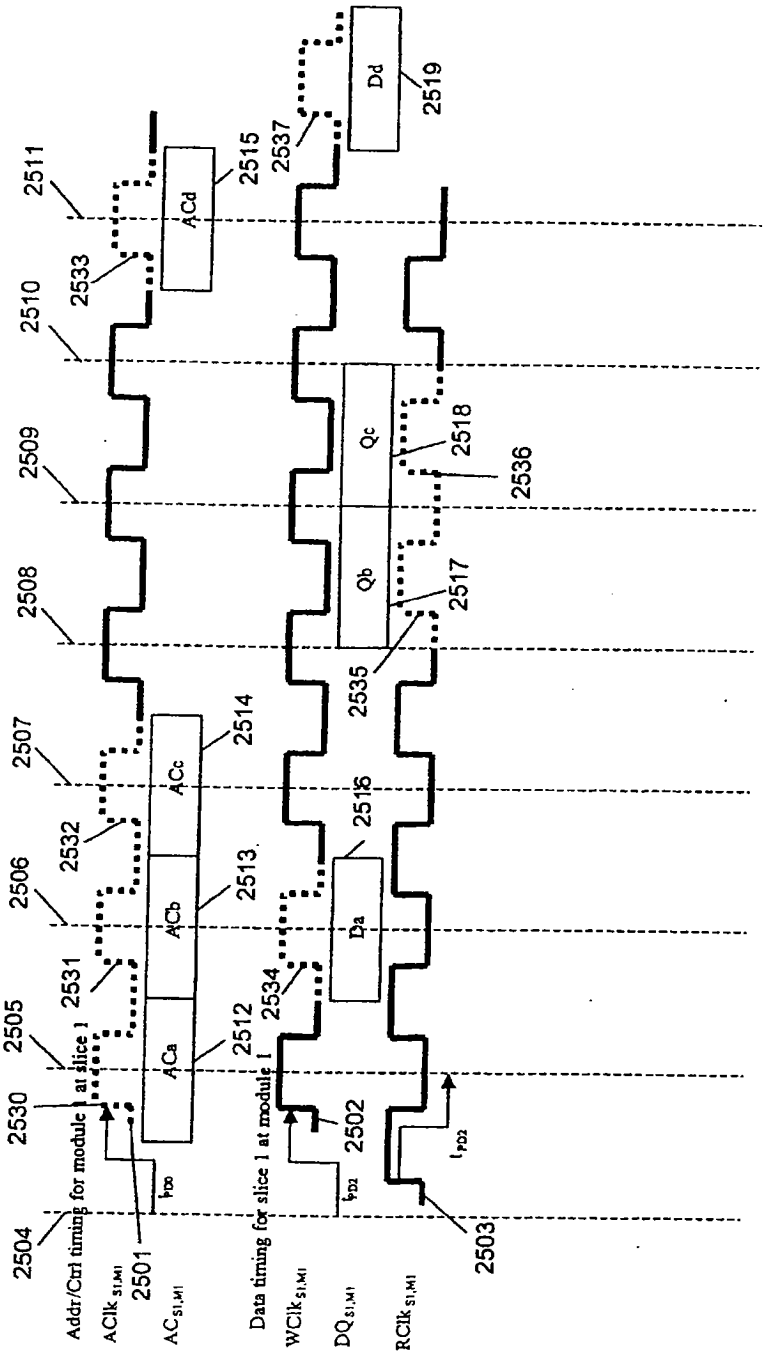


FIG. 25

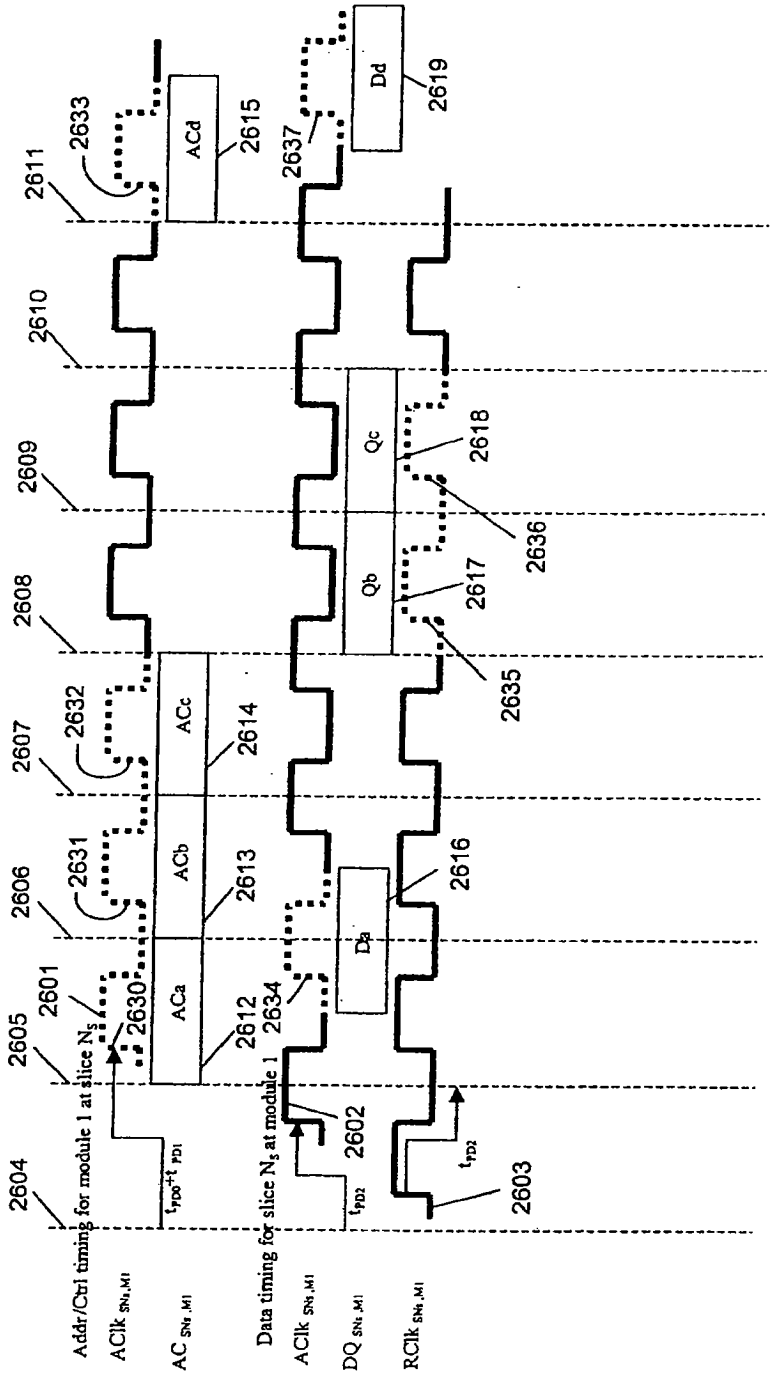


FIG. 26

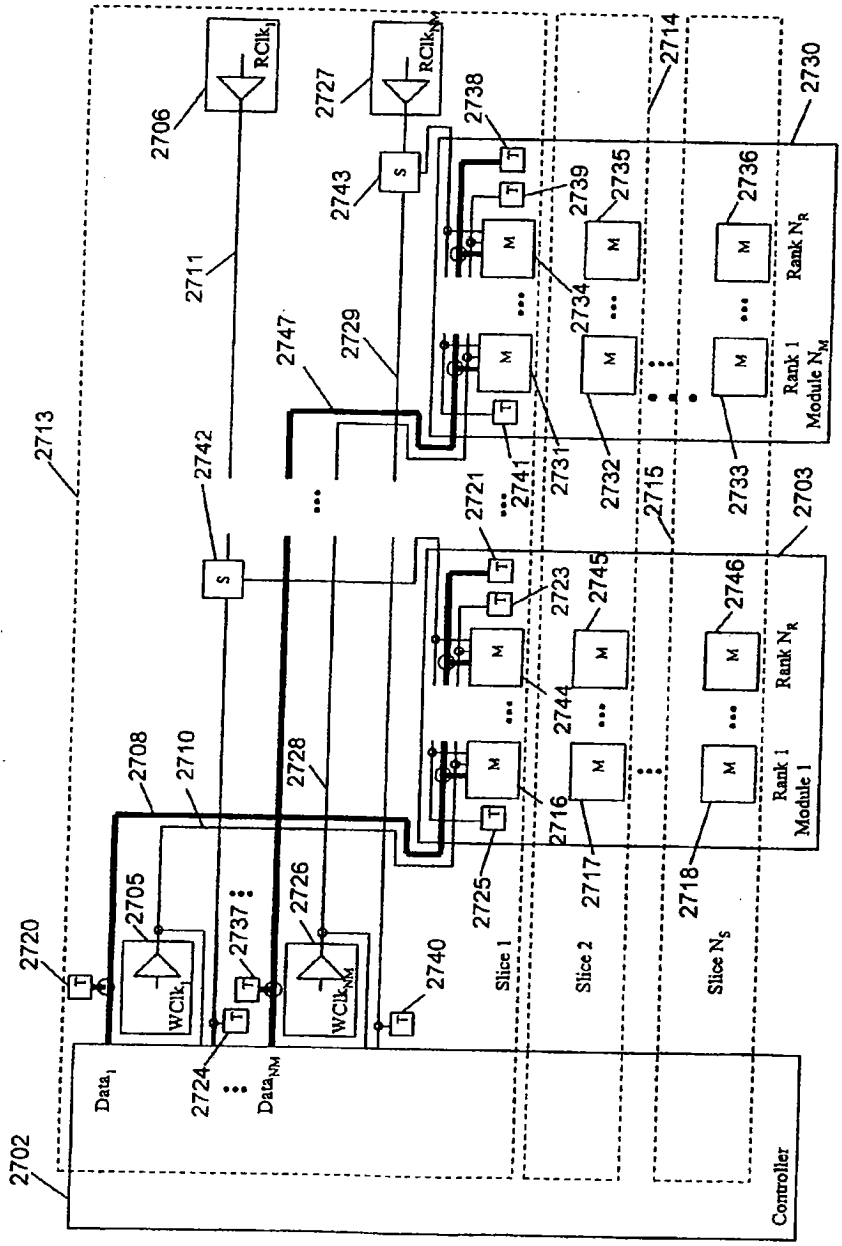


FIG. 27

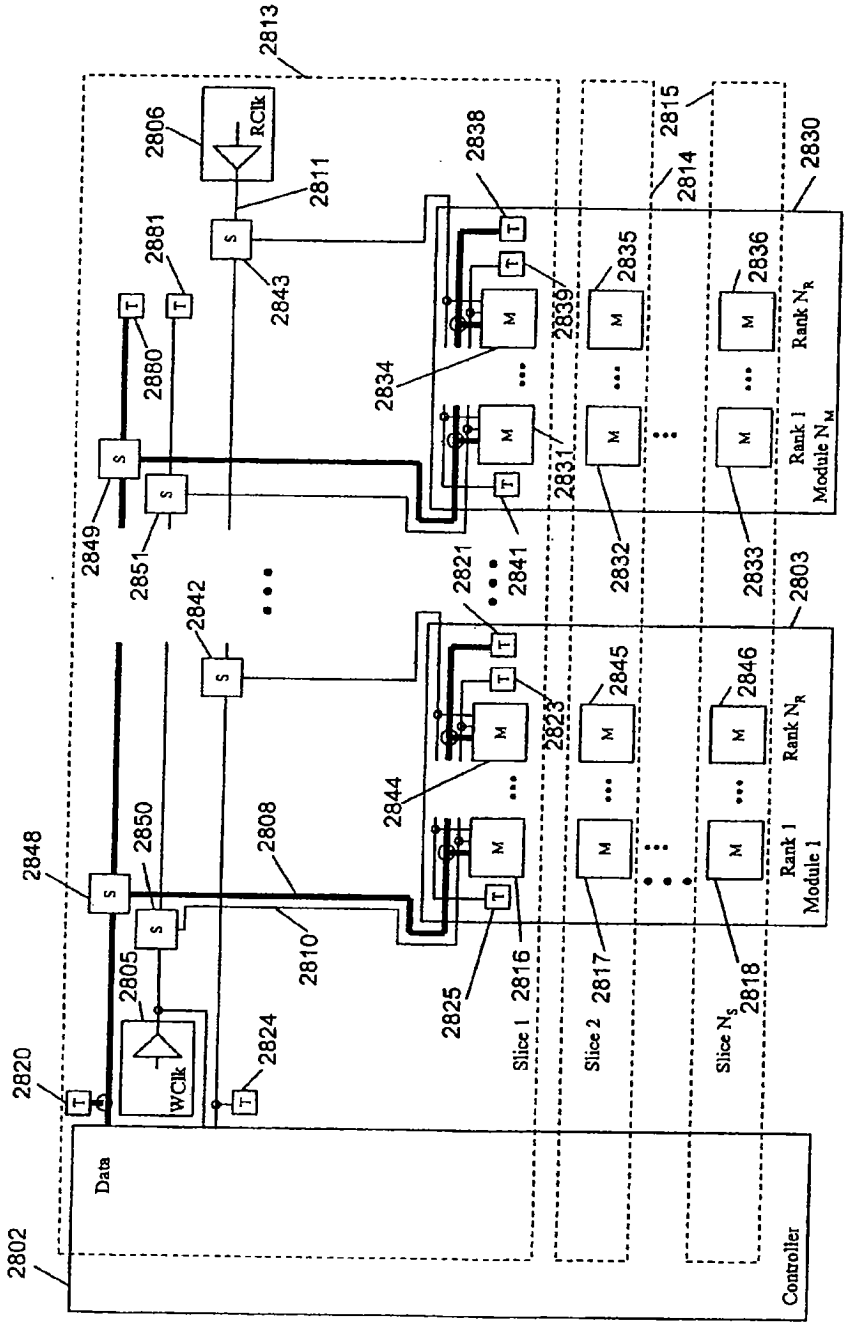


FIG. 28

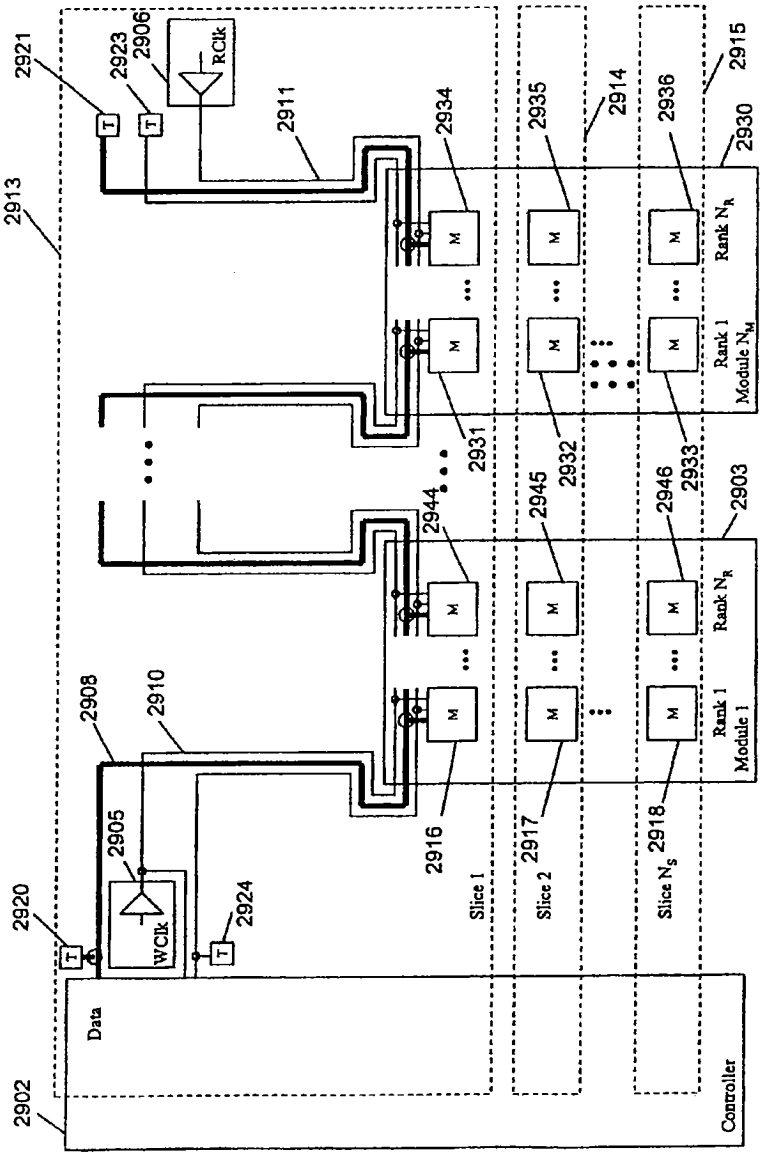


FIG. 29

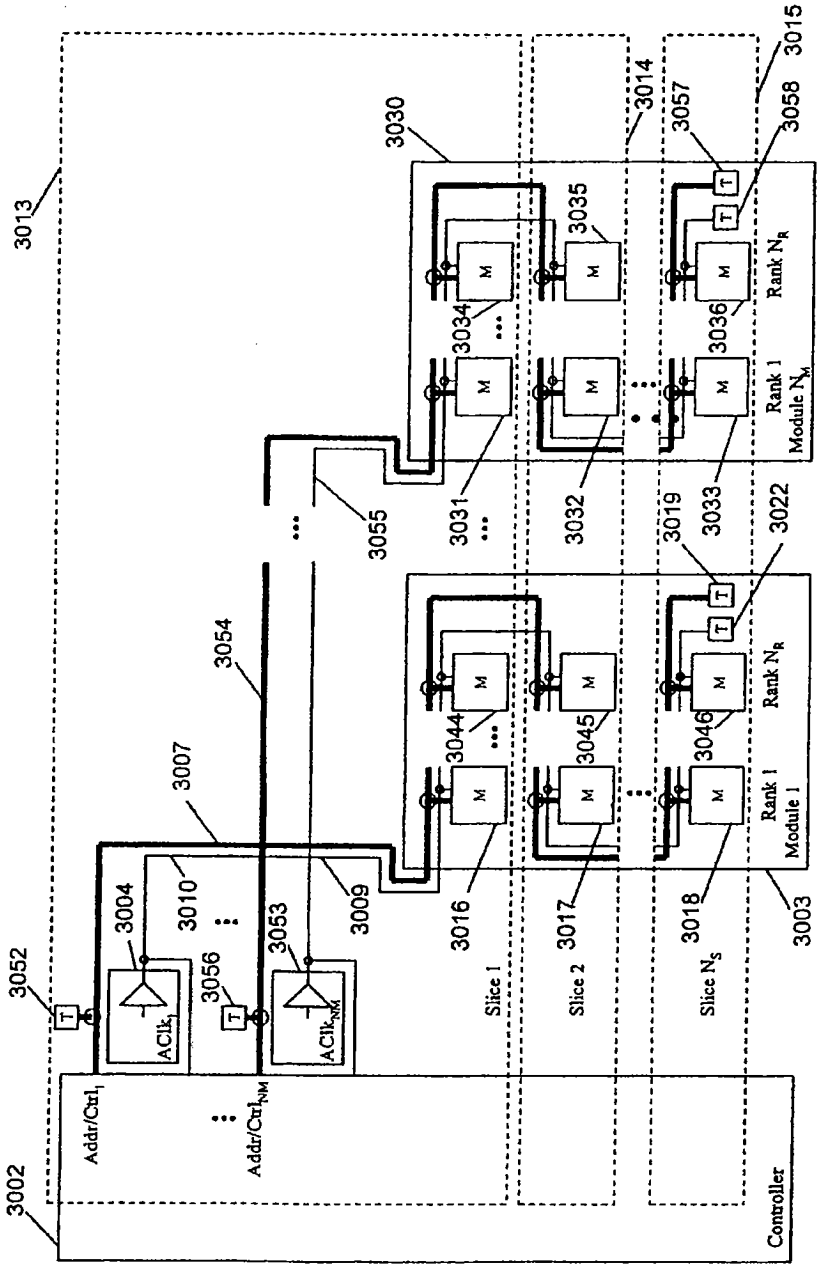


FIG. 30

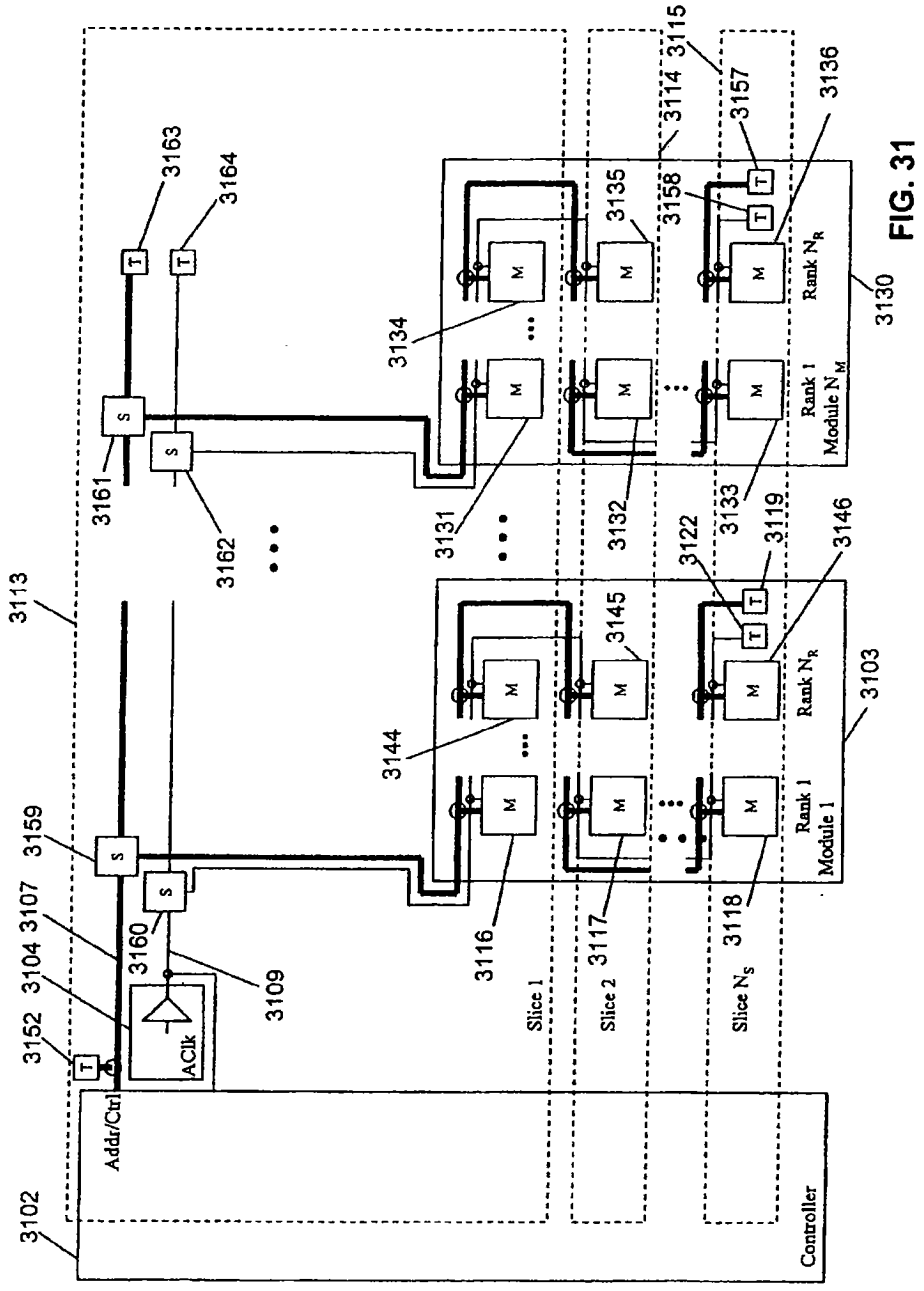


FIG. 31

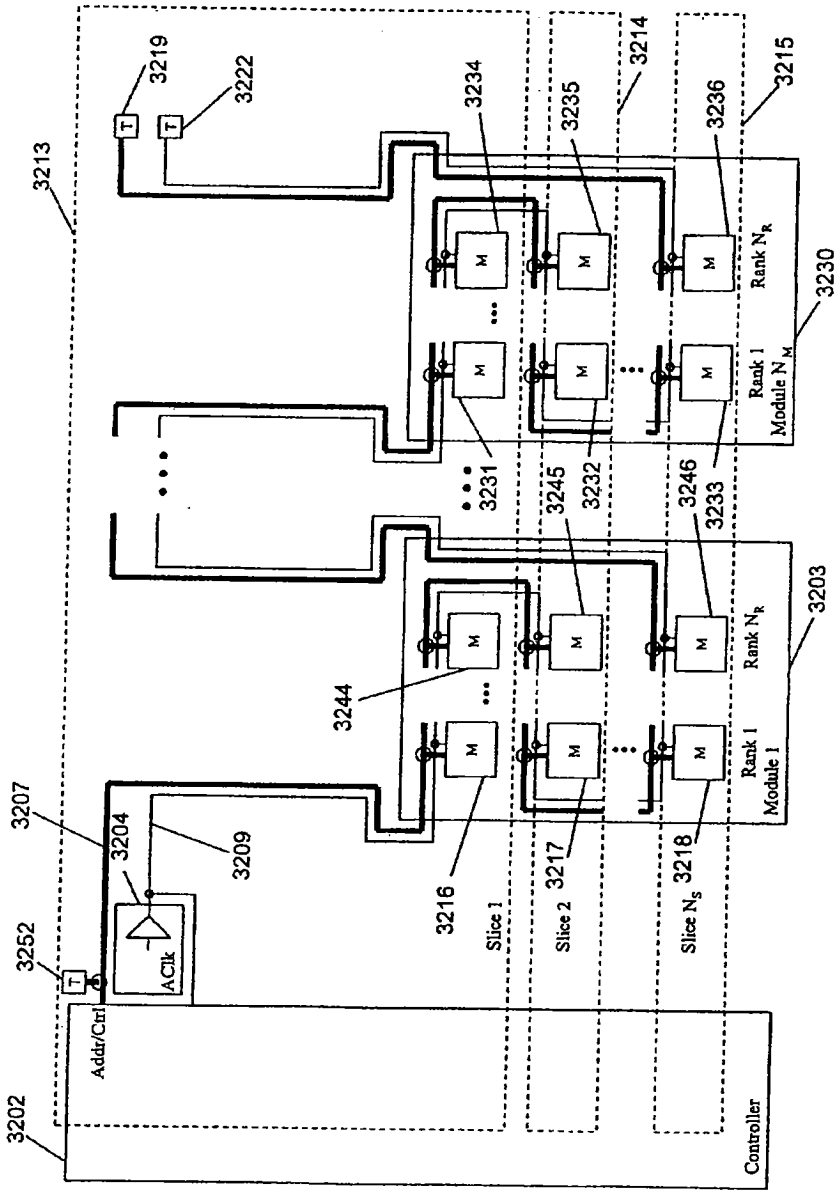


FIG. 32

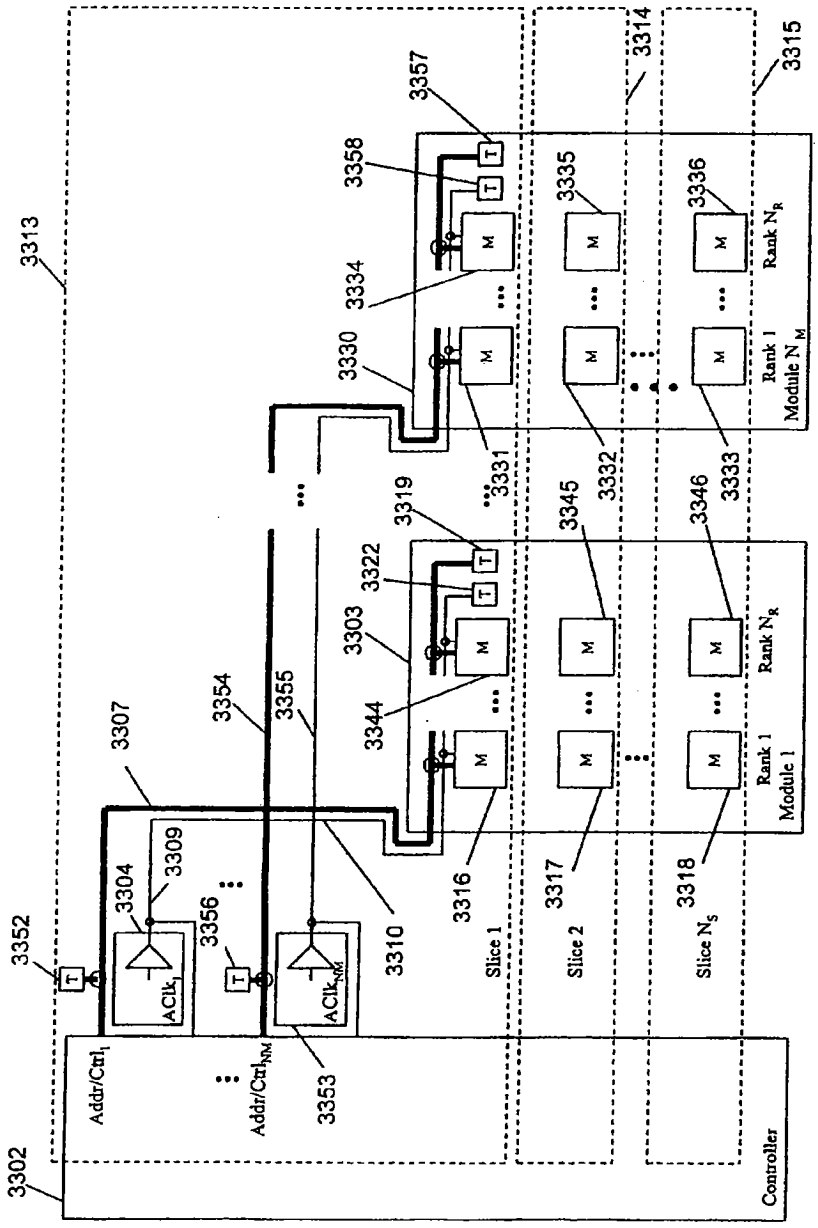


FIG. 33

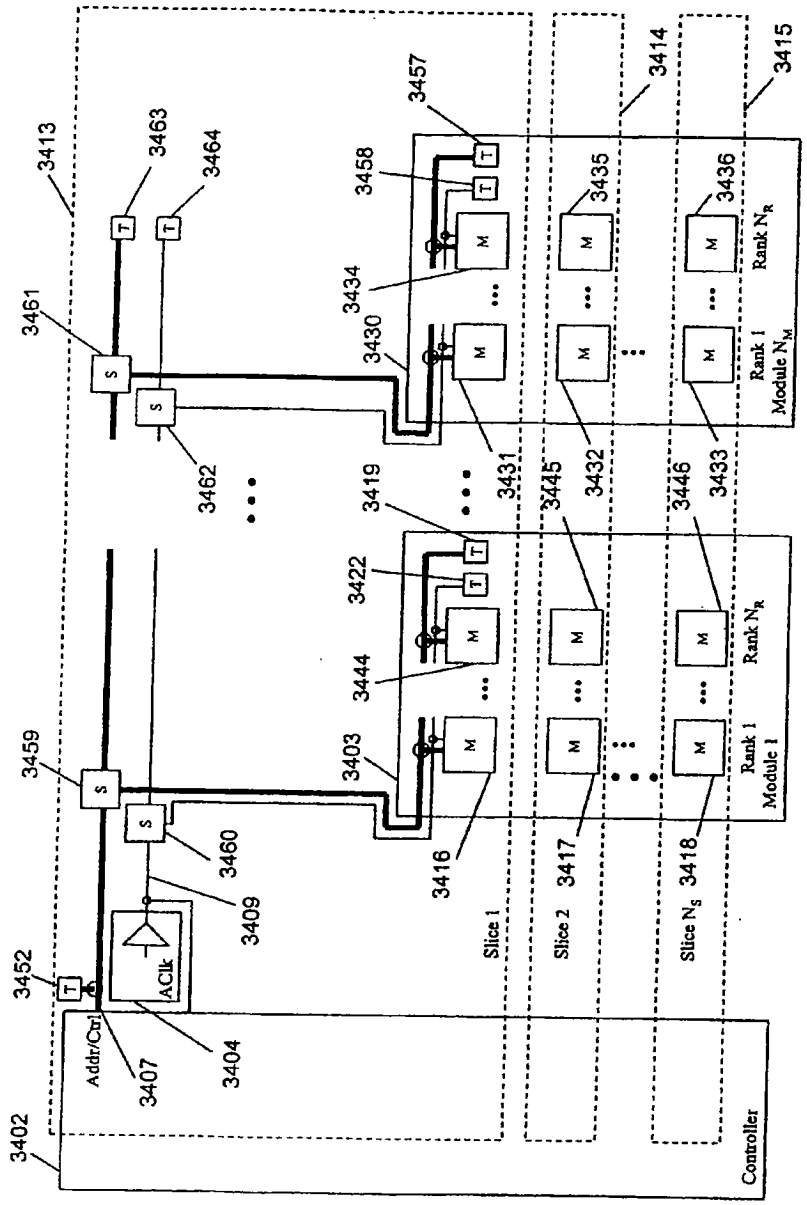


FIG. 34

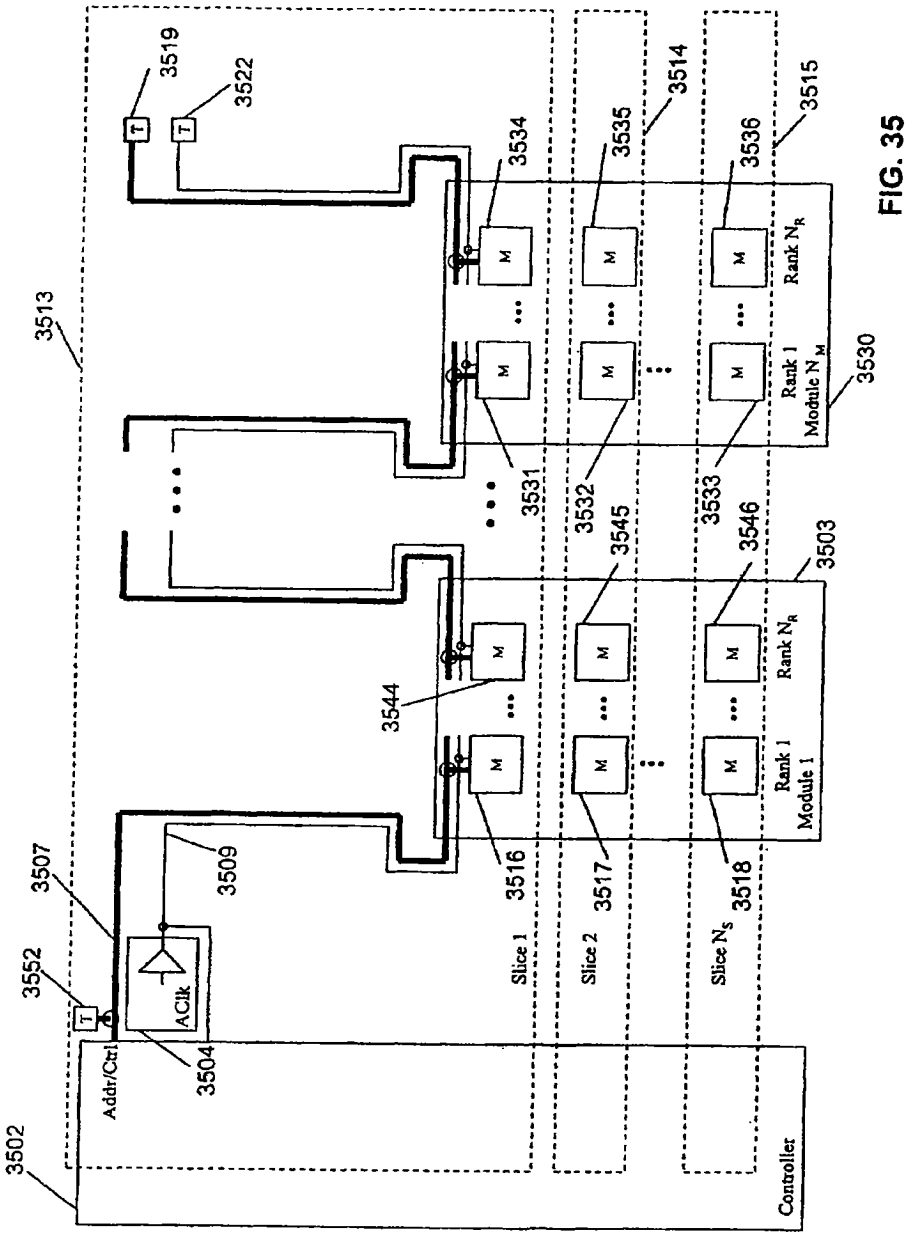


FIG. 35

METHOD, SYSTEM AND MEMORY CONTROLLER UTILIZING ADJUSTABLE READ DATA DELAY SETTINGS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 11/219,096 filed Sep. 1, 2005, which is a continuation of U.S. patent application Ser. No. 11/094,137 filed Mar. 31, 2005, which is a continuation of U.S. patent application Ser. No. 10/732,533 filed Dec. 11, 2003, which is a continuation of U.S. patent application Ser. No. 09/841,911, filed Apr. 24, 2001, now U.S. Pat. No. 6,675,272. U.S. Pat. No. 6,675,272 and each of the foregoing patent applications is hereby incorporated by reference.

TECHNICAL FIELD

The invention relates generally to information storage and retrieval and, more specifically, to coordinating memory components.

BACKGROUND

As computers and data processing equipment have grown in capability, users have developed applications that place increasing demands on the equipment. Thus, there is a continually increasing need to process more information in a given amount of time. One way to process more information in a given amount of time is to process each element of information in a shorter amount of time. As that amount of time is shortened, it approaches the physical speed limits that govern the communication of electronic signals. While it would be ideal to be able to move electronic representations of information with no delay, such delay is unavoidable. In fact, not only is the delay unavoidable, but, since the amount of delay is a function of distance, the delay varies according to the relative locations of the devices in communication.

Since there are limits to the capabilities of a single electronic device, it is often desirable to combine many devices, such as memory components, to function together to increase the overall capacity of a system. However, since the devices cannot all exist at the same point in space simultaneously, consideration must be given to operation of the system with the devices located diversely over some area.

Traditionally, the timing of the devices' operation was not accelerated to the point where the variation of the location of the devices was problematic to their operation. However, as performance demands have increased, traditional timing paradigms have imposed barriers to progress.

One example of an existing memory system uses DDR (double data rate) memory components. The memory system includes a memory controller and a memory module. A propagation delay occurs along an address bus between the memory controller and the memory module. Another propagation delay occurs along the data bus between the memory controller and the memory module.

The distribution of the control signals and a control clock signal in the memory module is subject to strict constraints. Typically, the control wires are routed so there is an equal length to each memory component. A "star" or "binary tree" topology is typically used, where each spoke of the star or each branch of the binary tree is of equal length. The intent is to eliminate any variation of the timing of the control

signals and the control clock signal between different memory components of a memory module, but the balancing of the length of the wires to each memory component compromises system performance (some paths are longer than they need to be). Moreover, the need to route wires to provide equal lengths limits the number of memory components and complicates their connections.

In such DDR systems, a data strobe signal is used to control timing of both data read and data write operations. The data strobe signal is not a periodic timing signal, but is instead only asserted when data is being transferred. The timing signal for the control signals is a periodic clock. The data strobe signal for the write data is aligned to the clock for the control signals. The strobe for the read data is delayed by delay relative to the control clock equal to the propagation delay along the address bus plus the propagation delay along the data bus. A pause in signaling must be provided when a read transfer is followed by a write transfer to prevent interference along various signal lines used. Such a pause reduces system performance.

Such a system is constrained in several ways. First, because the control wires have a star topology or a binary tree routing, reflections occur at the stubs (at the ends of the spokes or branches). The reflections increase the settling time of the signals and limit the transfer bandwidth of the control wires. Consequently, the time interval during which a piece of information is driven on a control wire will be longer than the time it takes a signal wavefront to propagate from one end of the control wire to the other. Additionally, as more modules are added to the system, more wire stubs are added to each conductor of the data bus, thereby adding reflections from the stubs. This increases the settling time of the signals and further limits the transfer bandwidth of the data bus.

Also, because there is a constraint on the relationship between the propagation delays along the address bus and the data bus in this system, it is hard to increase the operating frequency without violating a timing parameter of the memory component. If a clock signal is independent of another clock signal, those clock signals and components to which they relate are considered to be in different clock domains. Within a memory component, the write data receiver is operating in a different clock domain from the rest of the logic of the memory component, and the domain crossing circuitry will only accommodate a limited amount of skew between these two domains. Increasing the signaling rate of data will reduce this skew parameter (when measured in time units) and increase the chance that a routing mismatch between data and control wires on the board will create a timing violation.

Also, most DDR systems have strict limits on how large the address bus and data bus propagation delays may be (in time units). These are limits imposed by the memory controller and the logic that is typically included for crossing from the controller's read data receiver clock domain into the clock domain used by the rest of the controller. There is also usually a limit (expressed in clock cycles) on how large the sum of these propagation delays can be. If the motherboard layout makes this sum too large (when measured in time units), the signal rate of the system may have to be lowered, thereby decreasing performance.

In another example of an existing memory system, the control wires and data bus are connected to a memory controller and are routed together past memory components on each memory module. One clock is used to control the timing of write data and control signals, while another clock is used to control the timing of read data. The two clocks are

aligned at the memory controller. Unlike the previous prior art example, these two timing signals are carried on separate wires.

In such an alternate system, several sets of control wires and a data bus may be used to intercouple the memory controller to one or more of the memory components. The need for separate sets of control wires introduces additional cost and complexity, which is undesireable. Also, if a large capacity memory system is needed, the number of memory components on each data bus will be relatively large. This will tend to limit the maximum signal rate on the data bus, thereby limiting performance.

Thus, a technique is needed to coordinate memory operations among diversely-located memory components.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a memory system having a single rank of memory components with which an embodiment of the invention may be implemented.

FIG. **2** is a block diagram illustrating clocking details for one slice of a rank of memory components of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention.

FIG. **3** is a timing diagram illustrating address and control timing notations used in timing diagrams of other Figures.

FIG. **4** is a timing diagram illustrating data timing notations used in timing diagrams of other Figures.

FIG. **5** is a timing diagram illustrating timing of signals communicated over the address and control bus (Addr/Ctrl or $AC_{S,M}$) in accordance with an embodiment of the invention.

FIG. **6** is a timing diagram illustrating timing of signals communicated over the data bus ($DQ_{S,M}$) in accordance with an embodiment of the invention.

FIG. **7** is a timing diagram illustrating system timing at a memory controller component in accordance with an embodiment of the invention.

FIG. **8** is a timing diagram illustrating alignment of clocks $AClk_{S1,M1}$, $WClk_{S1,M1}$, and $RClk_{S1,M1}$ at the memory component in slice **1** of rank **1** in accordance with an embodiment of the invention.

FIG. **9** is a timing diagram illustrating alignment of clocks $AClk_{SNs,M1}$, $WClk_{SNs,M1}$, and $RClk_{SNs,M1}$ at the memory component in slice N_S of rank **1** in accordance with an embodiment of the invention.

FIG. **10** is a block diagram illustrating further details for one slice of a rank of memory components of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention.

FIG. **11** is a block diagram illustrating the clocking elements of one slice of a rank of the memory components of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention.

FIG. **12** is a block diagram illustrating details for the memory controller component of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention.

FIG. **13** is a block diagram illustrating the clocking elements of a memory controller component of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention.

FIG. **14** is a logic diagram illustrating details of the ClkC8 block of the memory controller component such as that illustrated in FIG. **12** in accordance with an embodiment of the invention.

FIG. **15** is a block diagram illustrating how the ClkC8 [N:1] signals are used in the transmit and receive blocks of the memory controller component such as that illustrated in FIG. **12** in accordance with an embodiment of the invention.

FIG. **16** is a block diagram illustrating a circuit for producing a ClkC8B clock and a ClkC1B clock based on the ClkC8A clock in accordance with an embodiment of the invention.

FIG. **17** is a block diagram illustrating details of the PhShC block in accordance with an embodiment of the invention.

FIG. **18** is a block diagram illustrating the logic details of the skip logic in a controller slice of the receive block of a memory controller component in accordance with an embodiment of the invention.

FIG. **19** is a timing diagram illustrating the timing details of the skip logic in a controller slice of the receive block of a memory controller component in accordance with an embodiment of the invention.

FIG. **20** is a block diagram illustrating the logic details of the skip logic in a controller slice of the transmit block of a memory controller component in accordance with an embodiment of the invention.

FIG. **21** is a timing diagram illustrating the timing details of the skip logic in a controller slice of the transmit block of a memory controller component in accordance with an embodiment of the invention.

FIG. **22** is a timing diagram illustrating an example of a data clocking arrangement in accordance with an embodiment of the invention.

FIG. **23** is a timing diagram illustrating an example of a data clocking arrangement in accordance with an embodiment of the invention.

FIG. **24** is a timing diagram illustrating timing at the memory controller component for the example of the data clocking arrangement illustrated in FIG. **23** in accordance with an embodiment of the invention.

FIG. **25** is a timing diagram illustrating timing at a first slice of a rank of memory components for the example of the data clocking arrangement illustrated in FIG. **23** in accordance with an embodiment of the invention.

FIG. **26** is a timing diagram illustrating timing a last slice of a rank of memory components for the example of the data clocking arrangement illustrated in FIG. **23** in accordance with an embodiment of the invention.

FIG. **27** is a block diagram illustrating a memory system that may comprise multiple ranks of memory components and multiple memory modules in accordance with an embodiment of the invention.

FIG. **28** is a block diagram illustrating a memory system that may comprise multiple ranks of memory components and multiple memory modules in accordance with an embodiment of the invention.

FIG. **29** is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules in accordance with an embodiment of the invention.

FIG. **30** is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a dedicated control/address bus per memory module in accordance with an embodiment of the invention.

FIG. **31** is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared among the memory modules in accordance with an embodiment of the invention.

FIG. 32 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared by all the memory modules in accordance with an embodiment of the invention.

FIG. 33 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a dedicated, sliced control/address bus per memory module in accordance with an embodiment of the invention.

FIG. 34 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared by all the memory modules in accordance with an embodiment of the invention.

FIG. 35 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared by all the memory modules in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

A method and apparatus for coordinating memory operations among diversely-located memory components is described. In accordance with an embodiment of the invention, wave-pipelining is implemented for an address bus coupled to a plurality of memory components. The plurality of memory components are configured according to coordinates relating to the address bus propagation delay and the data bus propagation delay. A timing signal associated with address and/or control signals which duplicates the propagation delay of these signals is used to coordinate memory operations. The address bus propagation delay, or common address bus propagation delay, refers to the delay for a signal to travel along an address bus between the memory controller component and a memory component. The data bus propagation delay refers to the delay for a signal to travel along a data bus between the memory controller component and a memory component.

According to one embodiment of the invention, a memory system includes multiple memory modules providing multiple ranks and multiple slices of memory components. Such a system can be understood with reference to FIG. **27**. The memory system of FIG. **27** includes memory module **2703** and memory module **2730**. Memory module **2703** includes a rank that includes memory components **2716–2618** and another rank that includes memory components **2744–2746**.

The memory system is organized into slices across the memory controller component and the memory modules. The memory system of FIG. **27** includes a slice **2713** that includes a portion of memory controller **2702**, a portion of memory module **2703** including memory components **2716** and **2744**, and a portion of memory module **2730** including memory components **2731** and **2734**. The memory system of FIG. **27** includes another slice **2714** that includes another portion of memory controller **2702**, another portion of memory module **2703** including memory components **2717** and **2745**, and another portion of memory module **2730** including memory components **2732** and **2735**. The memory system of FIG. **27** further includes yet another slice **2715** that includes yet another portion of memory controller **2702**, yet another portion of memory module **2703** including memory components **2718** and **2746**, and yet another portion of memory module **2730** including memory components **2733** and **2736**.

The use of multiple slices and ranks, which may be implemented using multiple modules, allows efficient interconnection of a memory controller and several memory components while avoiding degradation of performance that can occur when a data bus or address bus has a large number of connections to it. With a separate data bus provided for each slice, the number of connections to each data bus can be kept to a reasonable number. The separate data buses can carry different signals independently of each other. A slice can include one or more memory components per module. For example, a slice can include one memory component of each rank. Note that the term slice may be used to refer to the portion of a slice excluding the memory controller. In this manner, the memory controller can be viewed as being coupled to the slices. The use of multiple modules allows memory components to be organized according to their path lengths to a memory controller. Even slight differences in such path lengths can be managed according to the organization of the memory components into ranks. The organization of memory components according to ranks and modules allows address and control signals to be distributed efficiently, for example through the sharing of an address bus within a rank or module.

In one embodiment, a slice can be understood to include several elements coupled to a data bus. As one example, these elements can include a portion of a memory controller component, one or more memory components on one module, and, optionally, one or more memory components on another module. In one embodiment, a rank can be understood to include several memory components coupled by a common address bus. The common address bus may optionally be coupled to multiple ranks on the module or to multiple modules. The common address bus can connect a memory controller component to each of the slices of a rank in succession, thereby allowing the common address bus to be routed from a first slice of the rank to a second slice of the rank and from the second slice of the rank to a third slice of the rank. Such a configuration can simplify the routing of the common address bus.

For discussion purposes, a simplified form of a memory system is first discussed in order to illustrate certain concepts, whereas a more complex memory system that includes a plurality of modules and ranks is discussed later in the specification.

FIG. **1** is a block diagram illustrating a memory system having a single rank of memory components with which an embodiment of the invention may be implemented. Memory system **101** comprises memory controller component **102** and memory module **103**. Address clock **104** provides an address clock signal that serves as a timing signal associated with the address and control signals that propagate along address bus **107**. Address clock **104** provides its address clock signal along address clock conductor **109**, which is coupled to memory controller component **102** and to memory module **103**. The address and control signals are sometimes referred to as simply the address signals or the address bus. However, since control signals may routed according to a topology common to address signals, these terms, when used, should be understood to include address signals and/or control signals.

Write clock **105** provides a write clock signal that serves as a timing signal associated with the data signals that propagate along data bus **108** during write operations. Write clock **105** provides its write clock signal along write clock conductor **110**, which is coupled to memory controller component **102** and memory module **103**. Read clock **106** provides a read clock signal that serves as a timing signal

associated with the data signals that propagate along data bus **108** during read operations. Read clock **106** provides its read clock signal along read clock conductor **111**, which is coupled to memory controller component **102** and memory module **103**.

Termination component **120** is coupled to data bus **108** near memory controller component **102**. As one example, termination component **120** may be incorporated into memory controller component **102**. Termination component **121** is coupled to data bus **108** near memory module **103**. Termination component **121** is preferably incorporated into memory module **103**. Termination component **123** is coupled to write clock conductor **110** near memory component **116** of memory module **103**. Termination component **123** is preferably incorporated into memory module **103**. Termination component **124** is coupled to read clock conductor **111** near memory controller component **102**. As an example, termination component **124** may be incorporated into memory controller component **102**. Termination component **125** is coupled to read clock conductor **111** near memory component **116** of memory module **103**. Termination component **125** is preferably incorporated into memory module **103**. The termination components may utilize active devices (e.g., transistors or other semiconductor devices) or passive devices (e.g. resistors, capacitors, or inductors). The termination components may utilize an open connection. The termination components may be incorporated in one or more memory controller components or in one or more memory components, or they may be separate components on a module or on a main circuit board.

Memory module **103** includes a rank **112** of memory components **116**, **117**, and **118**. The memory module **103** is organized so that each memory component corresponds to one slice. Memory component **116** corresponds to slice **113**, memory component **117** corresponds to slice **114**, and memory component **118** corresponds to slice **115**. Although not shown in FIG. **1**, the specific circuitry associated with the data bus, write clock and associated conductors, and read clock and associated conductors that are illustrated for slice **113** is replicated for each of the other slices **114** and **115**. Thus, although such circuitry has not been illustrated in FIG. **1** for simplicity, it is understood that such dedicated circuitry on a slice-by-slice basis is preferably included in the memory system shown.

Within memory module **103**, address bus **107** is coupled to each of memory components **116**, **117**, and **118**. Address clock conductor **109** is coupled to each of memory components **116**, **117**, and **118**. At the terminus of address bus **107** within memory module **103**, termination component **119** is coupled to address bus **107**. At the terminus of address clock conductor **109**, termination component **122** is coupled to address clock conductor **109**.

In the memory system of FIG. **1**, each data signal conductor connects one controller data bus node to one memory device data bus node. However, it is possible for each control and address signal conductor to connect one controller address/control bus node to an address/control bus node on each memory component of the memory rank. This is possible for several reasons. First, the control and address signal conductors pass unidirectional signals (the signal wavefront propagates from the controller to the memory devices). It is easier to maintain good signal integrity on a unidirectional signal conductor than on a bidirectional signal conductor (like a data signal conductor). Second, the address and control signals contain the same information for all memory devices. The data signals will be different for all memory devices. Note that there might be some control signals (such as write enable signals) which are different for each memory device—these are treated as unidirectional data signals, and are considered to be part of the data bus for the purposes of this distinction. For example, in some instances, the data bus may include data lines corresponding to a large number of bits, whereas in some applications only a portion of the bits carried by the data bus may be written into the memory for a particular memory operation. For example, a 16-bit data bus may include two bytes of data where during a particular memory operation only one of the two bytes is to be written to a particular memory device. In such an example, additional control signals may be provided along a similar path as that taken by the data signals such that these control signals, which control whether or not the data on the data bit lines is written, traverse the system along a path with a delay generally matched to that of the data such that the control signals use in controlling the writing of the data is aptly timed. Third, routing the address and control signals to all the memory devices saves pins on the controller and memory module interface.

As a result, the control and address signals will be propagated on wires that will be longer than the wires used to propagate the data signals. This enables the data signals to use a higher signaling rate than the control and address signals in some cases.

To avoid impairment of the performance of the memory system, the address and control signals may be wave-pipelined in accordance with an embodiment of the invention. The memory system is configured to meet several conditions conducive to wave-pipelining. First, two or more memory components are organized as a rank. Second, some or all address and control signals are common to all memory components of the rank. Third, the common address and control signals propagate with low distortion (e.g. controlled impedance). Fourth, the common address and control signals propagate with low intersymbol-interference (e.g. single or double termination).

Wave-pipelining occurs when Tbit<Twire, where the timing parameter Twire is defined to be the time delay for a wavefront produced at the controller to propagate to the termination component at the end of the wire carrying the signal, and the timing parameter Tbit is defined to be the time interval between successive pieces (bits) of information on the wire. Such pieces of information may represent individual bits or multiple bits encoded for simultaneous transmission. Wave-pipelined signals on wires are incident-wave sampled by receivers attached to the wire. This means that sampling will generally take place before the wavefront has reflected from the end of the transmission line (e.g., the wire).

It is possible to extend the applicability of the invention from a single rank to multiple ranks of memory components in several ways. First, multiple ranks of memory components may be implemented on a memory module. Second, multiple memory modules may be implemented in a memory system. Third, data signal conductors may be dedicated, shared, or "chained" to each module. Chaining involves allowing a bus to pass through one module, connecting with the appropriate circuits on that module, whereas when it exits that particular module it may then enter another module or reach termination. Examples of such chaining of conductors are provided and described in additional detail in FIGS. **29**, **32**, and **35** below. Fourth, common control and address signal conductors may be dedicated, shared, or chained to each module. Fifth, data signal conductors may be terminated transmission lines or terminated stubs on each module. For this discussion, trans-

mission lines are understood to represent signal lines that have sufficient lengths such that reflections and other transmission line characteristics must be considered and accounted for in order to assure proper signal transmission over the transmission lines. In contrast, terminated stubs are understood to be of such limited length that the parasitic reflections and other transmission line characteristics associated with such stubs can generally be ignored. Sixth, common control and address signal conductors may be terminated transmission lines or terminated stubs on each module. Permitting the shared address and control signals to be wave-pipelined allows their signaling rate to be increased, thereby increasing the performance of the memory system.

FIG. **2** is a block diagram illustrating clocking details for one slice of a rank of memory components of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention. The memory controller component **102** includes address transmit block **201**, which is coupled to address bus **107** and address clock conductor **109**. The memory controller component **102** also includes, on a per-slice basis, data transmit block **202** and data receive block **203**, which are coupled to data bus **108**. Data transmit block **202** is coupled to write clock conductor **110**, and data receive block **203** is coupled to read clock conductor **111**.

Within each memory component, such as memory component **116**, an address receive block **204**, a data receive block **205**, and a data transmit block **206** are provided. The address receive block **204** is coupled to address bus **107** and address clock conductor **109**. The data receive block **205** is coupled to data bus **108** and write clock conductor **110**. The data transmit block **206** is coupled to data bus **108** and read clock conductor **111**.

A propagation delay **207**, denoted t_{PD0}, exists along address bus **107** between memory controller component **102** and memory module **103**. A propagation delay **208**, denoted t_{PD1}, exists along address bus **107** within memory module **103**.

The basic topology represented in FIG. **2** has several attributes. It includes a memory controller. It includes a single memory module. It includes a single rank of memory components. It includes a sliced data bus (DQ), with each slice of wires connecting the controller to a memory component. It includes a common address and control bus (Addr/Ctrl or AC) connecting the controller to all the memory components. Source synchronous clock signals flow with data, control, and address signals. Control and address signals are unidirectional and flow from controller to memory components. Data signals are bi-directional and may flow from controller to memory components (write operation) or may flow from memory components to controller (read operation). There may be some control signals with the same topology as data signals, but which flow only from controller to memory components. Such signals may be used for masking write data in write operations, for example. These may be treated as unidirectional data signals for the purpose of this discussion. The data, address, control, and clock wires propagate with low distortion (e.g., along controlled impedance conductors). The data, address, control, and clock wires propagate with low inter-symbol interference (e.g., there is a single termination on unidirectional signals and double termination on bi-directional signals). These attributes are listed to maintain clarity. It should be understood that the invention is not constrained to be practiced with these attributes and may be practiced so as to include other system topologies.

In FIG. **2**, there is a two dimensional coordinate system based on the slice number of the data buses and the memory components ($S=\{0,1, \ldots N_S\}$) and the module number ($M=\{0,1\}$). Here a slice number of '0' and a module number of '0' refer to the controller. This coordinate system allows signals to be named at different positions on a wire. This coordinate system will also allow expansion to topologies with more than one memory rank or memory module.

FIG. **2** also shows the three clock sources (address clock **104**, which generates the ACIk signal, write clock **105**, which generates the WCIk signal, and read clock **106**, which generates the RCIk signal) which generate the clocking reference signals for the three types of information transfer. These clock sources each drive a clock wire that is parallel to the signal bus with which it is associated. Preferably, the positioning of the clock sources within the system is such that the physical position on the clock line at which the clock source drives the corresponding clock signal is proximal to the related driving point for the bus line such that the propagation of the clock for a particular bus generally tracks the propagation of the related information on the associated bus. For example, the positioning of the address clock (ACIk clock **104**) is preferably close to the physical position where the address signals are driven onto the address bus **107**. In such a configuration, the address clock will experience similar delays as it propagates throughout the circuit as those delays experienced by the address signals propagating along a bus that follows generally the same route as the address clock signal line.

The clock signal for each bus is related to the maximum bit rate on the signals of the associated bus. This relationship is typically an integer or integer ratio. For example, the maximum data rate may be twice the frequency of the data clock signals. It is also possible that one or two of the clock sources may be "virtual" clock sources; the three clock sources will be in an integral-fraction-ratio (N/M) relationship with respect to one another, and any of them may be synthesized from either of the other two using phase-locked-loop (PLL) techniques to set the frequency and phase. Virtual clock sources represent a means by which the number of actual clock sources within the circuit can be minimized. For example, a WCIk clock might be derived from an address clock (ACIk) that is received by a memory device such that the memory device is not required to actually receive a WCIk clock from an external source. Thus, although the memory device does not actually receive a unique, individually-generated WCIk clock, the WCIk clock generated from the ACIk clock is functionally equivalent. The phase of a synthesized clock signal will be adjusted so it is the same as if it were generated by a clock source in the positions shown.

Any of the clock signals shown may alternatively be a non-periodic signal (a strobe control signal, for example) which is asserted only when information is present on the associated bus. As was described above with respect to clock sources, the non-periodic signal sources are preferably positioned, in a physical sense, proximal to the appropriate buses to which they correspond such that propagation delays associated with the non-periodic signals generally match those propagation delays of the signals on the buses to which they correspond.

FIG. **3** is a timing diagram illustrating address and control timing notations used in timing diagrams of other Figures. In FIG. **3**, a rising edge **302** of the ACIk signal **301** occurs at a time **307** during transmission of address information ACa **305**. A rising edge **303** of the ACIk signal occurs at a time **308** during transmission of address information ACb **306**.

Time **308** occurs at a time t_{CC} before the time **309** of the next rising edge **304** of AClk signal **301**. The time tCC represents a cycle time of a clock circuit of a memory controller component. Dashed lines in the timing diagrams are used to depict temporal portions of a signal coincident with address information or datum information. For example, the AClk signal **301** includes a temporal portion corresponding to the presence of address information ACa **305** and another temporal portion corresponding to the presence of address information ACb **306**. Address information can be transmitted over an address bus as an address signal.

If one bit per wire occurs per t_{CC}, address bit **311** is transmitted during cycle **310**. If two bits per wire occur per t_{CC}, address bits **313** and **314** are transmitted during cycle **312**. If four bits per wire occur per t_{CC}, address bits **316**, **317**, **318**, and **319** are transmitted during cycle **315**. If eight bits per wire occur per t_{CC}, address bits **321**, **322**, **323**, **324**, **325**, **326**, **327**, and **328** are transmitted during cycle **320**. Note that the drive and sample points for each bit window may be delayed or advanced by an offset (up to one bit time, which is t_{CC}/N_{AC}), depending upon the driver and sampler circuit techniques used. The parameters N_{AC} and N_{DQ} represent the number of bits per t_{CC} for the address/control and data wires, respectively. In one embodiment, a fixed offset is used. An offset between the drive/sample points and the bit windows should be consistent between the driving component and the sampling component. It is preferable that in a particular system, any offset associated with the drive point for a bus is consistent throughout the entire system. Similarly, any understood sampling offset with respect to the bus should also be consistent. For example, if data is expected to be driven at a point generally corresponding to a rising edge of a related clock signal for one data bus line, that understood offset (or lack thereof) is preferably consistently used for all data lines. Note that the offset associated with driving data onto the bus may be completely different than that associated with sampling data carried by the bus. Thus, continuing with the example above, the sample point for data driven generally coincident with a rising edge may be 180 degrees out of phase with respect to the rising edge such that the valid window of the data is better targeted by the sample point.

FIG. **4** is a timing diagram illustrating data timing notations used in timing diagrams of other Figures. In FIG. **4**, a rising edge **402** of the WClk signal **401** occurs at a time **407** during transmission of write datum information Da **405**. A rising edge **403** of the WClk signal **401** occurs at a time **408**. A rising edge **404** of the WClk signal **401** occurs at a time **409** during transmission of read datum information Qb **406**. Time **407** is separated from time **408** by a time t_{CC}, and time **408** is separated from time **409** by a time t_{CC}. The time t_{CC} represents the duration of a clock cycle. RClk signal **410** includes rising edge **411** and rising edge **412**. These rising edges may be used as references to clock cycles of RClk signal **410**. For example, transmission of write datum information Da **405** occurs during a clock cycle of RClk signal **410** that includes rising edge **411**, and transmission of read datum information Qb **406** occurs during a clock cycle of RClk signal **410** that includes rising edge **412**. As is apparent to one of ordinary skill in the art, the clock cycle time associated with the address clock may differ from the clock cycle time associated with the read and/or write clocks.

Write datum information is an element of information being written and can be transmitted over a data bus as a write data signal. Read datum information is an element of information being read and can be transmitted over a data bus as a read data signal. As can be seen, the notation Dx is used to represent write datum information x, while the notation Qy is used to represent read datum information y. Signals, whether address signals, write data signals, read data signals, or other signals can be applied to conductor or bus for a period of time referred to as an element time interval. Such an element time interval can be associated with an event occurring on a conductor or bus that carries a timing signal, where such an event may be referred to as a timing signal event. Examples of such a timing signal include a clock signal, a timing signal derived from another signal or element of information, and any other signal from which timing may be derived. In a memory access operation, the time from when an address signal begins to be applied to an address bus to when a data signal corresponding to that address signal begins to be applied to a data bus can be referred to as an access time interval.

If one bit per wire occurs per t_{CC}, datum bit **415** is transmitted during cycle **414**. If two bits per wire occur per t_{CC}, data bits **417** and **418** are transmitted during cycle **416**. If four bits per wire occur per t_{CC}, data bits **420**, **421**, **422**, and **423** are transmitted during cycle **419**. If eight bits per wire occur per t_{CC}, data bits **425**, **426**, **427**, **428**, **429**, **430**, **431**, and **432** are transmitted during cycle **424**. Note that the drive and sample points for each bit window may be delayed or advanced by an offset (up to one bit time, which is t_{CC}/N_{DQ}), depending upon the driver and sampler circuit techniques used. In one embodiment, a fixed offset is used. An offset between the drive/sample points and the bit windows should be consistent between the driving component and the sampling component. For example, if the data window is assumed to be positioned such that data will be sampled on the rising edge of the appropriate clock signal at the controller, a similar convention should be used at the memory device such that valid data is assumed to be present at the rising edge of the corresponding clock at that position within the circuit as well.

If one bit per wire occurs per t_{CC}, datum bit **434** is transmitted during cycle **433**. If two bits per wire occur per t_{CC}, data bits **436** and **437** are transmitted during cycle **435**. If four bits per wire occur per t_{CC}, data bits **439**, **440**, **441**, and **442** are transmitted during cycle **438**. If eight bits per wire occur per t_{CC}, data bits **444**, **445**, **446**, **447**, **448**, **449**, **450**, and **451** are transmitted during cycle **443**. Note that the drive and sample points for each bit window may be delayed or advanced by an offset (up to one bit time, which is t_{CC}/N_{DQ}), depending upon the driver and sampler circuit techniques used. In one embodiment, a fixed offset is used. An offset between the drive/sample points and the bit windows should be consistent between the driving component and the sampling component. As stated above, it is preferable that in a particular system, any offset associated with the drive point or sampling point for a bus is consistent throughout the entire system.

The column cycle time of the memory component represents the time interval required to perform successive column access operations (reads or writes). In the example shown, the AClk, RClk, and WClk clock signals are shown with a cycle time equal to the column cycle time. As is apparent to one of ordinary skill in the art, the cycle time of the clock signals used in the system may be different from the column cycle time in other embodiments.

Alternatively, any of the clocks could have a cycle time that is different than the column cycle time. The appropriate-speed clock for transmitting or receiving signals on a bus can always be synthesized from the clock that is distributed with the bus as long as there is an integer or integral-fraction-ratio between the distributed clock and the synthesized clock. As

mentioned earlier, any of the required clocks can be synthesized from any of the distributed clocks from the other buses.

This discussion will assume a single bit is sampled or driven on each wire during each t_{CC} interval in order to keep the timing diagrams as simple as possible. However, the number of bits that are transmitted on each signal wire during each t_{CC} interval can be varied. The parameters N_{AC} and N_{DQ} represent the number of bits per t_{CC} for the address/control and data wires, respectively. The distributed or synthesized clock is multiplied up to create the appropriate clock edges for driving and sampling the multiple bits per t_{CC}. Note that the drive and sample points for each bit window may be delayed or advanced by an offset (up to one bit time, which is t_{CC}/N_{AC} or t_{CC}/N_{DQ}), depending upon the driver and sampler circuit techniques used. In one embodiment, a fixed offset is used. An offset between the drive/sample points and the bit windows should be consistent between the driving component and the sampling component. Once again, as stated above, it is preferable that in a particular system, any offset associated with the drive point or sampling point for a bus is consistent throughout the entire system.

FIG. **5** is a timing diagram illustrating timing of signals communicated over the address and control bus (Addr/Ctrl or $AC_{S,M}$) in accordance with an embodiment of the invention. This bus is accompanied by a clock signal $ACLk_{S,M}$ which sees essentially the same wire path as the bus. The subscripts (S,M) indicate the bus or clock signal at a particular module M or a particular slice S. The controller is defined to be slice zero.

The waveform for ACLk clock signal **501** depicts the timing of the ACLk clock signal at the memory controller component. A rising edge **502** of ACLk clock signal **501** occurs at time **510** and is associated with the transmission of address information ACa **518**. A rising edge **503** of ACLk clock signal **501** occurs at time **511** and is associated with the transmission of address information ACb **519**.

The waveform for ACLk clock signal **520** depicts the timing of the ACLk clock signal at a memory component located at slice one. The ACLk signal **520** is delayed a delay of by t_{PD0} from signal **501**. For example, the rising edge **523** of signal **520** is delayed by a delay of t_{PD0} from edge **502** of signal **501**. The address information ACa **537** is associated with the rising edge **523** of signal **520**. The address information ACb **538** is associated with the rising edge **525** of signal **520**.

The waveform for ACLk clock signal **539** depicts the timing of the ACLk clock signal at the memory component located at slice N_S. The ACLk signal **539** is delayed by a delay of t_{PD1} from signal **520**. For example, the rising edge **541** of signal **539** is delayed by a delay of t_{PD1} from edge **523** of signal **520**. The address information ACa **548** is associated with the rising edge **541** of signal **539**. The address information ACb **549** is associated with the rising edge **542** of signal **539**.

The clock signal ACLk is shown with a cycle time that corresponds to the column cycle time. As previously mentioned, it could also have a shorter cycle time as long as the frequency and phase are constrained to allow the controller and memory components to generate the necessary timing points for sampling and driving the information on the bus. Likewise, the bus is shown with a single bit per wire transmitted per t_{CC} interval. As previously mentioned, more than one bit could be transferred in each t_{CC} interval since the controller and memory components are able to generate the necessary timing points for sampling and driving the information on the bus. Note that the actual drive point for the bus (the point at which data signals, address signals, and/or control signals are applied to the bus) may have an offset from what is shown (relative to the rising and falling edges of the clock)—this will depend upon the design of the transmit and receive circuits in the controller and memory components. In one embodiment, a fixed offset is used. An offset between the drive/sample points and the bit windows should be consistent between the driving component and the sampling component. As reiterated above, it is preferable that in a particular system, any offset associated with the drive point or sampling point for a bus is consistent throughout the entire system. It should be noted in FIG. **5** is that there is a delay t_{PD0} in the clock $ACLk_{S,M}$ and bus $AC_{S,M}$ as they propagate from the controller to the first slice. As indicated, ACLk signal **520** is shifted in time and space from ACLk signal **501**. Also note that there is a second delay t_{PD1} in the clock $ACLk_{S,M}$ and bus $AC_{S,M}$ as they propagate from the first slice to the last slice N_S. There will be a delay of $t_{PD1}/(N_S-1)$ as the clock and bus travel between each slice. Note that this calculation assumes generally equal spacing between the slices, and, if such physical characteristics are not present in the system, the delay will not conform to this formula. Thus, as indicated, ACLk signal **539** is shifted in time and space from ACLk signal **520**. As a result, the N_S memory components will each be sampling the address and control bus at slightly different points in time.

FIG. **6** is a timing diagram illustrating timing of signals communicated over the data bus ($DQ_{S,M}$) in accordance with an embodiment of the invention. This bus is accompanied by two clock signals $RClk_{S,M}$ and $WClk_{S,M}$ which see essentially the same wire path as the bus. The subscripts (S,M) indicate the bus or clock signal at a particular module M and a particular slice S. The controller is defined to be module zero. The two clocks travel in opposite directions. $WClk_{S,M}$ accompanies the write data which is transmitted by the controller and received by the memory components. $RClk_{S,M}$ accompanies the read data which is transmitted by the memory components and received by the controller. In the example described, read data (denoted by "Q") and write data (denoted by "D") do not simultaneously occupy the data bus. Note that in other embodiments, this may not be the case where additional circuitry is provided to allow for additive signaling such that multiple waveforms carried over the same conductor can be distinguished and resolved.

The waveform of WClk clock signal **601** depicts the timing of the WClk clock signal at the memory controller component. Rising edge **602** occurs at time **610** and is associated with write datum information Da **618**, which is present at slice one of module zero. Rising edge **607** occurs at time **615**, and is associated with write datum information Dd **621**, which is present at slice one of module zero. Rising edge **608** occurs at time **616**, and is associated with write datum De **622**, which is present at slice one of module zero.

The waveform of RClk clock signal **623** depicts the timing of the RClk clock signal at the memory controller component (at module zero). Rising edge **626** is associated with read datum information Qb **619**, which is present at the memory controller component (at slice one of module zero). Rising edge is associated with read datum information Qc **620**, which is present at the memory controller component (at slice one of module zero).

The waveform of WClk clock signal **632** depicts the timing of the WClk clock signal at the memory component at slice one of module one. Rising edge **635** is associated with write datum information Da **649**, which is present at slice one of module one. Rising edge **645** is associated with

write datum information Dd **652**, which is present at slice one of module one. Rising edge **647** is associated with write datum information De **653**, which is present at slice one of module one.

The waveform of RClk clock signal **654** depicts the timing of the RClk clock signal at the memory component of slice one of module one. Rising edge **658** is associated with read datum information Qb **650**, which is present at slice one of module one. Rising edge **660** is associated with read datum information Qd **651**, which is present at slice one of module one.

The clock signals are shown with a cycle time that corresponds to t_{CC}. As previously mentioned, they could also have a shorter cycle time as long as the frequency and phase are constrained to allow the controller and memory components to generate the necessary timing points for sampling and driving the information on the bus. Likewise, the bus is shown with a single bit per wire. As previously mentioned, more than one bit could be transferred in each t_{CC} interval since the controller and memory components are able to generate the necessary timing points for sampling and driving the information on the bus. Note that the actual drive point for the bus may have an offset from what is shown (relative to the rising and falling edges of the clock)—this will depend upon the design of the transmit and receive circuits in the controller and memory components. In one embodiment, a fixed offset is used. An offset between the drive/sample points and the bit windows should be consistent between the driving component and the sampling component.

It should be noted in FIG. **6** is that there is a delay t_{PD2} in the clock $WClk_{S,M}$ and bus $DQ_{S,M}$ (with the write data) as they propagate from the controller to the slices of the first module. Thus, WClk clock signal **632** is shifted in time and space from WClk clock signal **601**. Also note that there is an approximately equal delay t_{PD2} in the clock $RClk_{S,M}$ and bus $DQ_{S,M}$ (with the read data) as they propagate from the slices of the first module to the controller. Thus, RClk clock signal **623** is shifted in time and space from RClk clock signal **654**.

As a result, the controller and the memory components must have their transmit logic coordinated so that they do not attempt to drive write data and read data at the same time. The example in FIG. **6** shows a sequence in which there are write-read-read-write-write transfers. It can be seen that read-read and write-write transfers may be made in successive t_{CC} intervals, since the data in both intervals is traveling in the same direction. However, gaps (bubbles) are inserted at the write-read and read-write transitions so that a driver only turns on when the data driven in the previous interval is no longer on the bus (it has been absorbed by the termination components at either end of the bus wires).

In FIG. **6**, the read clock $RClk_{S,M}$ and the write clock $WClk_{S,M}$ are in phase at each memory component (however the relative phase of these clocks at each memory component will be different from the other memory components—this will be shown later when the overall system timing is discussed). Note that this choice of phase matching is one of several possible alternatives that could have been used. Some of the other alternatives will be described later.

As a result of matching the read and write clocks at each memory component (slice), the t_{CC} intervals with read data Qb **650** will appear to immediately follow the t_{CC} intervals with write data Da **649** at the memory components (bottom of FIG. **6**), but there will be a gap of $2*t_{PD2}$ between the read data interval Qb **619** and write data interval Da **618** at the controller (top of FIG. **6**). There will be a second gap of $(2*t_{CC}-2*t_{PD2})$ between the read data Qc **620** and the write data Dd **621** at the controller. There will be a gap of $(2*t_{CC})$ between the read data Qc **651** and the write data Dd **621**. Note that the sum of the gaps at the memory components and the controller will be $2*t_{CC}$.

The overall system timing will be described next. The example system phase aligns the $AClk_{S,M}$, $RClk_{S,M}$, and $WClk_{S,M}$ clocks at each memory component (the slice number varies from one through N_S, and the module number is fixed at one). This has the benefit of allowing each memory component to operate in a single clock domain, avoiding any domain crossing issues. Because the address and control clock $AClk_{S,M}$ flows past each memory component, the clock domain of each memory slice will be offset slightly from the adjacent slices. The cost of this phasing decision is that the controller must adjust the read and write clocks for each slice to different phase values—this means there will be $1+(2*N_S)$ clock domains in the controller, and crossing between these domains efficiently becomes very important. Other phase constraints are possible and will be discussed later.

FIG. **7** is a timing diagram illustrating system timing at a memory controller component in accordance with an embodiment of the invention. As before, the controller-sends a write-read-read-write sequence of operations on the control and address bus $AClk_{S0,M1}$. The Da write datum information is sent on the $WClk_{S1,M0}$ and $WClk_{SNs,M0}$ buses so that it will preferably arrive at the memory component of each slice one cycle after the address and control information ACa. This is done by making the phase of the $WClk_{S1,M0}$ clock generally equivalent to $(t_{PD0}-t_{PD2})$ relative to the phase of the $AClk_{S0,M1}$ clock (positive means later, negative means earlier). This will cause them to be in phase at the memory component of the first slice of the first module. Likewise, the phase of the $WClk_{SNs,M0}$ clock is adjusted to be generally equivalent to $(t_{PD0}+t_{PD1}-t_{PD2})$ relative to the phase of the $AClk_{S0,M1}$ clock. Note that some tolerance is preferably built into the system such that the phase adjustment of the clock to approximate the propagation delays can vary slightly from the desired adjustment while still allowing for successful system operation.

In a similar fashion, the phase of the $RClk_{S1,M0}$ clock is adjusted to be generally equivalent to $(t_{PD0}+t_{PD2})$ relative to the phase of the $AClk_{S0,M1}$ clock. This will cause them to be in phase at the memory component of the last slice of the first module. Likewise, the phase of the $RClk_{SNs,M0}$ clock is adjusted according to the expression $(t_{PD0}+t_{PD1}+t_{PD2})$ relative to the phase of the $AClk_{S0,M1}$ clock to cause the $RClk_{SNs,M0}$ clock and the $AClk_{S0,M1}$ clock to be in phase at the memory component of the last slice of the first module.

The waveform of AClk clock signal **701** depicts the AClk clock signal at the memory controller component, which is denoted as being at slice zero. Rising edge **702** occurs at time **710** and is associated with address information ACa **718**, which is present at slice zero. Rising edge **703** occurs at time **711** and is associated with address information ACb **719**, which is present at slice zero. Rising edge **704** occurs at time **712** and is associated with address information ACc **720**, which is present at slice zero. Rising edge **707** occurs at time **715** and is associated with address information ACd **721**, which is present at slice zero.

The waveform of WClk clock signal **722** depicts the WClk clock signal for the memory component at slice one when that WClk clock signal is present at the memory controller component at module zero. Rising edge **724** occurs at time **711** and is associated with write datum information Da **730**, which is present. Rising edge **729**

occurs at time **716** and is associated with write datum information Dd **733**, which is present.

The waveform of RClk clock signal **734** depicts the RClk clock signal for the memory component of slice one when that RClk clock signal is present at the memory controller component at module zero. Rising edge **737** is associated with read datum information Qb **731**, which is present. Rising edge **738** is associated with read datum information Qc **732**, which is present.

The waveform of WClk clock signal **741** depicts the WClk clock signal for the memory component at slice N_S when that WClk clock signal is present at the memory controller component at module zero. Write datum information Da **756** is associated with edge **744** of signal **741**. Write datum information Dd **759** is associated with edge **754** of signal **741**.

The waveform of RClk clock signal **760** depicts the RClk clock signal for the memory component at slice N_S when that RClk clock signal is present at the memory controller component at module zero. Read datum information Qb **757** is associated with edge **764** of signal **760**. Read datum information Qc **758** is associated with edge **766** of signal **760**.

FIG. **8** is a timing diagram illustrating alignment of clocks $AClk_{S1,M1}$, $WClk_{S1,M1}$, and $RClk_{S1,M1}$ at the memory component in slice **1** of rank **1** in accordance with an embodiment of the invention. All three clocks are delayed by t_{PD0} relative to the $AClk_{S0,M1}$ clock produced at the controller.

The waveform of ACIk clock signal **801** depicts the ACIk clock signal for the memory component at slice one of module one. Address information ACa **822** is associated with edge **802** of signal **801**. Address information ACb **823** is associated with edge **804** of signal **801**. Address information ACc **824** is associated with edge **806** of signal **801**. Address information ACd **825** associated with edge **812** of signal **801**.

The waveform of WClk clock signal **826** depicts the WClk clock signal for the memory component at slice one of module one. Write datum information Da **841** is associated with edge **829** of signal **826**. Write datum information Dd **844** is associated with edge **839** of signal **826**.

The waveform of RClk clock signal **845** depicts the RClk clock signal for the memory component at slice one of module one. Read datum information Qb **842** is associated with edge **850** of signal **845**. Read datum information Qc **843** is associated with edge **852** of signal **845**.

FIG. **9** is a timing diagram illustrating alignment of clocks $AClk_{SNs,M1}$, $WClk_{SNs,M1}$, and $RClk_{SNs,M1}$ at the memory component in slice N_S of rank one of module one in accordance with an embodiment of the invention. All three clocks are delayed by $(t_{PD0}+t_{PD1})$ relative to the $AClk_{S0,M1}$ clock produced at the controller.

The waveform of ACIk clock signal **901** depicts the ACIk clock signal for the memory component at slice N_S at module one. Rising edge **902** of signal **901** is associated with address information ACa **917**. Rising edge **903** of signal **901** is associated with address information ACb. Rising edge **904** of signal **901** is associated with address information ACc **919**. Rising edge **907** of signal **901** is associated with address information ACd **920**.

The waveform of WClk clock signal **921** depicts the WClk clock signal for the memory component at slice N_S at module one. Rising edge **923** of signal **921** is associated with write datum information Da **937**. Rising edge **928** of signal **921** is associated with write datum information Dd **940**.

The waveform RClk clock signal **929** depicts the RClk clock signal for the memory component at slice N_S at module one. Rising edge **932** of signal **929** is associated with read datum information Qb **938**. Rising edge **933** of signal **929** is associated with read datum information Qc **939**.

Note that in both FIGS. **8** and **9** there is a one t_{CC} cycle delay between the address/control information (ACa **917** of FIG. **9**, for example) and the read or write information that accompanies it (Da **937** of FIG. **9** in this example) when viewed at each memory component. This may be different for other technologies; i.e. there may be a longer access delay. In general, the access delay for the write operation at the memory component should be equal or approximately equal to the access delay for the read operation in order to maximize the utilization of the data bus.

FIGS. **10** through **18** illustrate the details of an exemplary system which uses address and data timing relationships which are nearly identical to what has been described in FIGS. **5** through **9**. In particular, all three clocks are in-phase on each memory component. This example system has several differences relative to this earlier description, however. First, two bits per wire are applied per t_{CC} interval on the AC bus (address/control bus, or simply address bus). Second, eight bits per wire are applied per t_{CC} interval on the DQ bus. Third, a clock signal accompanies the AC bus, but the read and write clocks for the DQ bus are synthesized from the clock for the AC bus.

FIG. **10** is a block diagram illustrating further details for one memory rank (one or more slices of memory components) of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention. The internal blocks of the memory components making up this rank are connected to the external AC or DQ buses. The serialized data on these external buses is converted to or from parallel form on internal buses which connect to the memory core (the arrays of storage cells used to hold information for the system). Note that FIG. **10** shows all 32 bits of the DQ bus connecting to the memory rank—these 32 bits are divided up into multiple, equal-sized slices and each slice of the bus is routed to one memory component. Thus, slices are defined based on portions of the DQ bus routed to separate memory components. The example shown in FIG. **10** illustrates a memory component, or device, that supports the entire set of 32 data bits for a particular example system. In other embodiments, such a system may include two memory devices, where each memory device supports half of the 32 data bits. Thus, each of these memory devices would include the appropriate data transmit blocks, data receive blocks, and apportionment of memory core such that they can individually support the portion of the overall data bus for which they are responsible. Note that the number of data bits need not be 32, but may be varied.

The ACIk signal is the clock which accompanies the AC bus. It is received and is used as a frequency and phase reference for all the clock signals generated by the memory component. The other clocks are ClkM2, ClkM8, and ClkM. These are, respectively, 2×, 8×, and 1× the frequency of ACIk. The rising edges of all clocks are aligned (no phase offset). The frequency and phase adjustment is typically done with some type of phase-locked-loop (PLL) circuit, although other techniques are also possible. A variety of different suitable PLL circuits are well known in the art. The feedback loop includes the skew of the clock drivers needed to distribute the various clocks to the receive and transmit blocks as well as the memory core. The memory core is assumed to operate in the ClkM domain.

Memory component **116** comprises memory core **1001**, PLL **1002**, PLL **1003**, and PLL **1004**. ACIk clock signal **109** is received by buffer **1015**, which provides clock signal **1019**

to PLLs **1002**, **1003**, and **1004**. Various PLL designs are well known in the art, however some PLLs implemented in the example embodiments described herein require minor customization to allow for the specific functionality desired. Therefore, in some embodiments described herein, the particular operation of the various blocks within the PLL are described in additional detail. Thus, although some of the PLL constructs included in the example embodiments described herein are not described in extreme detail, it is apparent to one of ordinary skill in the art that the general objectives to be achieved by such PLLs are readily recognizable through a variety of circuits well known to those skilled in the art. PLL **1002** includes phase comparator and voltage controlled oscillator (VCO) **1005**. PLL **1002** provides clock signal ClkM **1024** to memory core **1001**, address/control receive block **204**, data receive block **205**, and data transmit block **206**.

PLL **1003** comprises prescaler **1009**, phase comparator and VCO **1010**, and divider **1011**. Prescaler **1009** may be implemented as a frequency divider (such as that used to implement divider **1011**) and provides a compensating delay with no frequency division necessary. Prescaler **1009** provides a signal **1021** to phase comparator and VCO **1010**. The phase comparator in VCO **1010** is represented as a triangle having two inputs and an output. The functionality of the phase comparator **1010** is preferably configured such that it produces an output signal that ensures that the phase of the feedback signal **1023**, which is one of its inputs, is generally phase aligned with a reference signal **1021**. This convention is preferably applicable to similar structures included in other PLLs described herein. Divider **1011** provides a feedback signal **1023** to phase comparator and VCO **1010**. PLL **1003** provides clock signal ClkM2 **1025** to address/control receive block **204**.

PLL **1004** comprises prescaler **1006**, phase comparator and VCO **1007**, and divider **1008**. Prescaler **1006** may be implemented as a frequency divider (such as that used to implement divider **1011**) and provides a compensating delay with no frequency division necessary. Prescaler **1006** provides a signal **1020** to phase comparator and VCO **1007**. Divider **1008** provides a feedback signal **1022** to phase comparator and VCO **1007**. PLL **1004** provides clock signal ClkM8 **1026** to data receive block **205** and data transmit block **206**.

The address bus **107** is coupled via buffers **1012** to address/control receive block **204** via coupling **1016**. The data outputs **1018** of data transmit block **206** are coupled to data bus **108** via buffers **1014**. The data bus **108** is coupled to data inputs **1017** of data receive block **205** via buffers **1013**.

Address/control receive block **204** provides address information to the memory core **1001** via internal address bus **1027**. Data receive blocks **205** provides write data to memory core **1001** via internal write data bus **1028**. Memory core **1001** provides read data to data transmit blocks **206** via internal read data bus **1029**.

FIG. **11** is a block diagram illustrating logic used in the receive and transmit blocks of FIG. **10** in accordance with an embodiment of the invention. In this Figure, for clarity, the elements for only one bit of each bus are illustrated. It is understood that such elements may be replicated for each bit of the bus.

Address/control receive block **204** comprises registers **1101**, **1102**, and **1103**. Address bus conductor **1016** is coupled to registers **1101** and **1102**, which together form a shift register, and which are clocked by ClkM2 clock signal **1025** and coupled to register **1103** via couplings **1104** and **1105**, respectively. Register **1103** is clocked by ClkM clock signal **1024** and provides address/control information to internal address bus **1027**. The representation of registers **1101** and **1102** in FIG. **11** is preferably understood to imply that they form a shift register such that data entering register **1101** during one cycle is transferred into register **1102** during the subsequent cycle as new data enters register **1101**. In the particular embodiment shown in FIG. **11**, the movement of data is controlled by the clock signal ClkM2 **1025**. Thus, if clock ClkM2 **1025** operates at twice the frequency of clock ClkM **1024**, the receive block **204** generally operates as a serial-to-parallel shift register, where two consecutive serial bits are grouped together in a two-bit parallel format before being output onto signal lines RAC **1027**. Thus, other similar representations in the figures where a number of registers are grouped together in a similar configuration preferably are understood to include the interconnections required to allow data to be serially shifted along the path formed by the registers. Examples include the registers **1123**–**1130** included in transmit block **206** and the registers **1106**–**1113** included in receive block **205**. As a result, the serial information on the input **1016** is converted to parallel form on the output **1027**.

Data receive block **205** comprises registers **1106**, **1107**, **1108**, **1109**, **1110**, **1111**, **1112**, **1113**, and **1114**. Data input **1017** is coupled to registers **1106**, **1107**, **1108**, **1109**, **1110**, **1111**, **1112**, and **1113**, which are clocked by ClkM8 clock signal **1026** and coupled to register **1114** via couplings **1115**, **1116**, **1117**, **1118**, **1119**, **1120**, **1121**, and **1122**, respectively. Register **1114** is clocked by ClkM clock signal **1024** and provides write data to internal write data bus **1028**. As a result, the serial information on the input **1017** is converted to parallel form on the output **1028**.

Data transmit block **206** comprises registers **1123**, **1124**, **1125**, **1126**, **1127**, **1128**, **1129**, **1130**, and **1131**. Read data from internal read data bus **1029** is provided to register **1131**, which is clocked by ClkM clock **1024** and coupled to registers **1123**, **1124**, **1125**, **1126**, **1127**, **1128**, **1129**, and **1130** via couplings **1132**, **1133**, **1134**, **1135**, **1136**, **1137**, **1138**, and **1139**. Registers **1123**, **1124**, **1125**, **1126**, **1127**, **1128**, **1129**, and **1130** are clocked by ClkM8 clock **1026** and provide data output **1018**. As a result, the parallel information on the input **1029** is converted to serial form on the output **1018**.

Shown are the register elements needed to sample the address/control and write data, and to drive the read data. It is assumed in this example that two bits are transferred per address/control (AC[i]) wire in each t_{CC} interval, and that eight bits are transferred per read data (Q[i]) wire or write data (D[i]) wire in each t_{CC} interval. In addition to the primary clock ClkM (with a cycle time of t_{CC}), there are two other aligned clocks that are generated. There is ClkM2 (with a cycle time of $t_{CC}/2$) and ClkM8 (with a cycle time of $t_{CC}/8$). These higher frequency clocks shift information in to or out from the memory component. Once in each t_{CC} interval the serial data is transferred to or from a parallel register clocked by ClkM.

Note that ClkM2 and ClkM8 clocks are frequency locked and phase locked to the ClkM clock. The exact phase alignment of the two higher frequency clocks will depend upon the circuit implementation of the driver and sampler logic. There may be small offsets to account for driver or sampler delay. There may also be small offsets to account for the exact position of the bit valid windows on the AC and DQ buses relative to the ClkM clock.

Note also that in the memory component, the ClkM2 or ClkM8 clocks could be replaced by two or eight clocks each with a cycle time of t_{CC}, but offset in phase in equal

increments across the entire t_{CC} interval. The serial register, which in transmit block **204** includes registers **1101–1102**, in transmit block **206** includes registers **1123–1130**, and in data receive block **205** includes registers **1106–1113**, would be replaced by a block of two or eight registers, each register loaded with a different clock signal so that the bit windows on the AC and DQ buses are properly sampled or driven. For example, in the transmit block **204**, two individual registers would be included, where one register is clocked by a first clock signal having a particular phase and the second register is clocked by a different clock signal having a different phase, where the phase relationship between these two clock signals is understood such that the equivalent serial-to-parallel conversion can be achieved as that described in detail above. Another possibility is to use level-sensitive storage elements (latches) instead of edge sensitive storage elements (registers) so that the rising and falling edges of a clock signal cause different storage elements to be loaded.

Regardless of how the serialization is implemented, there are multiple bit windows per t_{CC} interval on each wire, and multiple clock edges per t_{CC} interval are created in the memory component in order to properly drive and sample these bit windows.

FIG. **12** is a block diagram illustrating details for the memory controller component of a memory system such as that illustrated in FIG. **1** in accordance with an embodiment of the invention. The memory controller component **102** comprises PLLs **1202**, **1203**, **1204**, and **1205**, address/control transmit blocks **201**, data transmit blocks **202**, data receive blocks **203**, and controller logic core **1234**. PLL **1202** comprises phase comparator and VCO **1206**. PLL **1202** receives ClkIn clock signal **1201** and provides ClkC clock signal **1215** to controller logic core **1234** and to buffer **1224**, which outputs AClk clock signal **109**.

PLL **1203** comprises prescaler **1207**, phase comparator and VCO **1208**, and divider **1209**. Prescaler **1207** may be implemented as a frequency divider and provides a compensating delay with no frequency division necessary. Prescaler **1207** receives ClkIn clock signal **1201** and provides signal **1216** to phase comparator and VCO **1208**. Divider **1209** provides feedback signal **1218** to phase comparator and VCO **1208**, which provides ClkC2 clock output **1217** to address/control transmit blocks **201**.

PLL **1204** comprises phase comparator and VCO **1210**, dummy phase offset selector **1212**, and divider **1211**. Dummy phase offset selector **1212** inserts an amount of delay to mimic the delay inherent in a phase offset selector and provides signal **1220** to divider **1211**, which provides feedback signal **1221** to phase comparator and VCO **1210**. Phase comparator and VCO **1210** receives ClkIn clock input **1201** and provides ClkC8 clock output **1219** to data transmit blocks **202** and data receive blocks **203**.

PLL **1205** comprises phase shifting circuit **1214** and phase comparator and VCO **1213**. Phase shifting circuit **1214** provides feedback signal **1223** to phase comparator and VCO **1213**. Phase comparator and VCO **1213** receives ClkIn clock signal **1201** and provides ClkCD clock signal **1222** to data transmit blocks **202** and data receive blocks **203**.

Controller logic core **1234** provides TPhShB signals **1235** and TPhShA signals **1236** to data transmit blocks **202**. Controller logic core **1234** provides RPhShB signals **1237** and RPhShA signals **1238** to data receive blocks **203**. Controller logic core **1234** provides LoadSkip signal **1239** to data transmit blocks **202** and data receive blocks **203**. Controller logic core **1234** comprises PhShC block **1240**. Functionality of the controller logic **1234** is discussed in additional detail with respect to FIG. **17** below.

Controller logic core **1234** provides address/control information to address/control transmit blocks **201** via internal address bus **1231**. Controller logic core **1234** provides write data to data transmit blocks **1232** via internal write data bus **1232**. Controller logic core **1234** receives read data from data receive blocks **203** via internal read data bus **1233**.

Address/control transmit blocks **201** are coupled via output **1228** to buffers **1225**, which drive AC bus **107**. Data transmit blocks **202** provide outputs **1229** to buffers **1226**, which drive DQ bus **108**. Buffers **1227** couple DQ bus **108** to inputs **1230** of data receive blocks **203**.

Each of address/control transmit blocks **201** is connected to the AC bus, and each of blocks **202** and **203** is connected to the DQ bus. The serialized data on these external buses is converted to or from parallel from internal buses which connect to the rest of the controller logic. The rest of the controller is assumed to operate in the ClkC clock domain.

In the embodiment shown, the ClkIn signal is the master clock for the whole memory subsystem. It is received and used as a frequency and phase reference for all the clock signals used by the controller. The other clocks are ClkC2, ClkC8, ClkC, and ClkCD. These are, respectively, 2×, 8×, 1×, and 1× the frequency of ClkIn. ClkC will have no phase offset relative to ClkIn, and ClkCD will be delayed by 90 degrees. ClkC2 has every other rising edge aligned with a rising edge of ClkIn.

Every eighth ClkC8 rising edge is aligned with a rising edge of ClkIn except for an offset which compensates for the delay of a frequency divider and phase offset selector in the transmit and receive blocks. There are "N" additional ClkC8 signals (ClkC8[N:1]) which are phase-shifted relative to the ClkC8 signal. These other ClkC8 phases are used to synthesize the transmit and receive clock domains needed to communicate with the memory components.

The frequency and phase adjustment is typically done with some type of phase-locked-loop (PLL) circuit, although other techniques are also possible. The feedback loop of the PLL circuit includes the skew of the clock drivers needed to distribute the various clocks to the receive and transmit blocks as well as the rest of the controller logic.

FIG. **13** is a block diagram illustrating the logic used in the receive and transmit blocks of FIG. **12** in accordance with an embodiment of the invention. Memory controller component **102** comprises address/control transmit blocks **201**, data transmit blocks **202**, and data receive blocks **203**. For clarity, the elements for only one bit are illustrated. It is understood that such elements may be replicated for each bit of the buses.

Address/control transmit blocks **201** comprise register **1301** and registers **1302** and **1303**. Internal address bus **1231** is coupled to register **1301**, which is clocked by ClkC clock **1215** and provides outputs to registers **1302** and **1303** via couplings **1304** and **1305**, respectively. Registers **1302** and **1303** are clocked by ClkC2 clock **1217** and provide output **1328** to the AC bus. As a result, the parallel information on the internal address bus **1231** is converted to serial form on the output **1228**. Additional functional description of the address/control transmit blocks **201** is provided with respect to FIG. **13** below.

Generally, the data transmit blocks **202** and data receive blocks **203** shown in FIG. **13** serve the function of performing serial-to-parallel or parallel-to-serial conversion of data (the type of conversion depending upon the direction of the data flow). Such blocks are similar to those present within the memory devices, however in the case of the transmit and

receive blocks included in the controller in this particular system, additional circuitry is required in order to obtain the appropriate clocking signals required to perform these serial-to-parallel and parallel-to-serial conversions. In the memory devices of this example, such clock adjustment circuitry is not required, as the clocks are understood to be phase aligned within the memory devices. However, within the controller such phase alignment cannot be guaranteed due to the assumption within the system that phase alignment within the memory devices will possibly cause phase mismatching in other portions of the system due to the physical positioning of the memory devices with respect to the controller. Thus, a memory device positioned a first distance from the controller will have a different set of characteristic delays with respect to signals communicated with the controller than a second memory device positioned at a second position. As such, individual clock adjustment circuitry would be required for such memory devices within the controller such that the controller is assured of properly capturing read data provided by each of the memory devices and to allow for the controller to properly drive write data intended to be received by each of the memory devices.

Within the transmit block **202**, data for transmission is received over the TD bus **1232** in parallel format. This data is loaded into the register **1310** based on the clock ClkC signal **1215**. Once loaded in the register **1310**, the data is either directly passed through the multiplexer **1312** to the register **1313** or caused to be delayed by a half clock cycle by traversing the path through the multiplexer **1312** that includes the register **1311** which is clocked by the falling edge of the ClkC signal. Such circuitry enables the data on the TD bus, which is in the ClkC clock domain, to be successfully transferred into the clock domain needed for its transmission. This clock domain is the TClkC1B clock domain, which has the same frequency as the ClkC clock, but is not necessarily phase aligned to the ClkC clock signal. Similar circuitry is included within the receive block **203** such that data received in the RClkC1B clock domain can be successfully transferred onto the RQ bus that operates in the ClkC clock domain.

Data transmit blocks **202** comprise PhShA block **1306**, clock divider circuit **1307**, registers **1308**, **1309**, **1310**, **1311**, and **1313**, multiplexer **1312**, and shift register **1314**. TPhShA signals **1236** and ClkC8 clock signals **1219** are provided to PhShA block **1306**. Additional detail regarding the PhShA block **1306** are provided with respect to FIG. **15** below. Clock divider circuit **1307** comprises 1/1 divider circuit **1324** and 1/8 divider circuit **1325**. TPhShB signals **1235** are provided to 1/8 divider circuit **1325**. An output of PhShA block **1306** is provided to inputs of 1/1 divider circuit **1324** and 1/8 divider circuit **1325**. An output of 1/1 divider circuit **1324** is provided to clock shift register **1314**. An output of 1/8 divider circuit **1325** is provided to clock register **1313** and as an input to register **1308**.

Register **1308** is clocked by ClkCD clock signal **1222** and provides an output to register **1309**. Register **1309** is clocked by ClkC clock signal **1215** and receives LoadSkip signal **1238** to provide an output to multiplexer **1312** and an output to clock registers **1310** and **1311**. Register **1310** receives write data from write data bus **1232** and provides an output to register **1311** and multiplexer **1312**. Register **1311** provides an output to multiplexer **1312**. Multiplexer **1312** provides an output to register **1313**. Register **1313** provides parallel outputs to shift register **1314**. Shift register **1314** provides output **1229**. As a result, the parallel information on the input **1232** is converted to serial form on the output **1229**.

Data receive blocks **203** comprise PhShA block **1315**, clock dividing circuit **1316**, registers **1317**, **1318**, **1320**, **1321**, and **1323**, shift register **1319**, and multiplexer **1322**. Clock dividing circuit **1316** comprises 1/1 divider circuit **1326** and 1/8 divider circuit **1327**. RPhShA signals **1238** and ClkC8 clock signal **1219** are provided to PhShA block **1315**, which provides an output to 1/1 divider circuit **1326** and 1/8 divider circuit **1327**. RPhShB signal **1237** is provided to an input of 1/8 divider circuit **1327**. The 1/1 divider circuit **1326** provides an output used to clock shift register **1319**. The 1/8 divider circuit **1327** provides an output used to clock register **1320** and used as an input to register **1317**. Register **1317** is clocked by ClkCD clock signal **1222** and provides an output to register **1318**. Register **1318** receives LoadSkip signal **1238** and is clocked by ClkC clock signal **1215**, providing an output to multiplexer **1322** and an output. used to clock registers **1321** and **1323**.

Shift register **1319** receives input **1230** and provides parallel outputs to register **1320**. Register **1320** provides an output to register **1321** and to multiplexer **1322**. Register **1321** provides an output to multiplexer **1322**. Multiplexer **1322** provides an output to register **1323**. Register **1323** provides an output to internal read data bus **1233**. As a result, the serial information on the input **1230** is converted to parallel form on the output **1233**.

Shown are the register and gating elements needed to drive address/control and write data, and to sample the read data. It is assumed in this example that two bits are transferred per address/control (AC[i]) wire in each t_{CC} interval, and that eight bits are transferred per read data (Q[i]) wire or write data (D[i]) wire in each t_{CC} interval. In addition to the primary clock ClkC (with a cycle time of t_{CC}), there are two other aligned clocks that are generated. There is ClkC2 (with a cycle time of $t_{CC}/2$) and ClkC8 (with a cycle time of $t_{CC}/8$). These higher frequency clocks shift information in to or out from the controller. Once in every t_{CC} interval the serial data is transferred to or from a parallel register clocked by ClkC.

Note that in the controller, the ClkC2 or ClkC8 clocks can be replaced by two or eight clocks each with a cycle time of t_{CC}, but offset in phase in equal increments across the entire t_{CC} interval. In such embodiments, the serial register is replaced by blocks of two or eight registers, where each register is loaded with a different clock signal so that the bit windows on the AC and DQ buses are properly sampled or driven. Another possibility is to use level-sensitive storage elements (latches) instead of edge sensitive storage elements (registers) so that the rising and falling edges of a clock signal cause different storage elements to be loaded.

Regardless of how the serialization is implemented, there will be multiple bit windows per t_{CC} interval on each wire, and many embodiments utilize multiple clock edges per t_{CC} interval in the controller in order to properly drive and sample these bit windows.

FIG. **13** also shows how the controller deals with the fact that the read and write data that is received and transmitted for each slice is in a different clock domain. Since a slice may be as narrow as a single bit, there can be 32 read clock domains and 32 write clock domains simultaneously present in the controller (this example assumes a DQ bus width of 32 bits). Remember that in this example no clocks are transferred with the read and write data, and such clocks are preferably synthesized from a frequency source. The problem of multiple clock domains would still be present even if a clock was transferred with the read and write data. This is because the memory component is the point in the system

where all local clocks are preferably in-phase. Other system clocking topologies are described later in this description.

The transmit block for address/control bus (AC) in FIG. **13** uses the ClkC2 and ClkC clocks to perform two-to-one serialization. The ClkC2 clock shifts the serial register **1302**, **1304** onto the AC wires **1328**. Note the exact phase alignment of the ClkC2 clock depends upon the circuit implementation of the driver logic; there may be a small offset to account for driver delay. There may also be small offsets to account for the exact position of the bit drive window on the AC bus relative to the ClkC clock. For example, if the output drivers have a known delay, the phase of the ClkC2 clock signal may be adjusted such that a portion of the output circuitry begins providing data to the output drivers slightly earlier than the time at which the data is to actually be driven onto an external signal line. The shifting of the phase of the ClkC2 clock signal can thus be used to account for the inherent delay in the output driver such that data is actually presented on the external data line at the desired time. Similarly, adjustments to the phase of the ClkC2 clock signal may also be used to ensure that the positioning of the valid data window for data driven based on the ClkC2 clock signal is optimally placed.

In a similar fashion, the transmit block for write data bus (D) in FIG. **13** uses a phase-delayed ClkC8 clock to perform eight-to-one serialization. The phase-delayed ClkC8 clock shifts the serial register **1314** onto the DQ wires. Note the exact alignment of the phase-delayed ClkC8 clock will depend upon the circuit implementation of the driver logic; there may be a small offset to account for driver delay. There may also be small offsets to account for the exact position of the bit drive window on the DQ bus.

The TphShA[i][n:0] control signals **1236** select the appropriate phase offset relative to the input reference vectors ClkC8[N:1]. A phase offset selector may be implemented using a simple multiplexer, a more elaborate phase interpolator, or other phase offset selection techniques. In one example of a phase interpolator, a first reference vector of less-than-desired phase offset and a second reference vector of greater-than-desired phase offset are selected. A weighting value is applied to combine a portion of the first reference vector with a portion of the second reference vector to yield the desired output phase offset of the TClkC8A clock. Thus, the desired output phase offset of the TClkC8A clock is effectively interpolated from the first and second reference vectors. In one example of a phase multiplexer, the TphShA[i][n:0] control signals **1236** are used to select one of the ClkC8[N:1] clock signals **1219** to pass through to the TClkC8A clock (note that 2^{n+1}=N). The phase that is used is, in general, different for each transmit slice on the controller. The phase for each slice on the controller is preferably selected during a calibration process during initialization. This process is described in detail later in this description.

The TClkC8A clock passes through 1/8 **1325** and 1/1 **1324** frequency dividers before clocking the parallel **1313** and serial **1314** registers. Note that the ClkC8[N:1] signals that are distributed have a small phase offset to compensate for the delay of the phase offset selection block (PhShA) **1306** and the frequency divider blocks **1324** and **1325**. This offset is generated by a phase-locked-loop circuit and will track out supply voltage and temperature variations.

Even with the transmit phase shift value set correctly (so that the bit windows on the D bus **1229** are driven properly), the phase of the TClkC1B clock used for the parallel register **1313** could be misaligned (there are eight possible combinations of phase). There are several ways of dealing with the problem. The scheme that is used in the embodiment illustrated provides an input TPhShB **1235**, such that when this input is pulsed, the phase of the TClkC1B clock will shift by $\frac{1}{8}^{th}$ of a cycle (45 degrees). The initialization software adjusts the phase of this clock until the parallel register loads the serial register at the proper time. This initialization process is described in detail later in this description.

Alternatively, it is also possible to perform the phase adjustment in the ClkC domain when preparing the TD bus **1232** for loading into the transmit block **202**. To do so, multiplexers and registers may be used to rotate the write data across ClkC cycle boundaries. A calibration process may be provided at initialization to accommodate the phase of the TClkC1B clock during which the transmit block **202** is powered up.

After the phase shift controls are properly adjusted, the write data can be transmitted onto the D bus from the parallel register **1313**. However, the write data still needs to be transferred from the TD bus **1232** in the ClkC **1215** domain into the parallel register **1313** in the TClkC1B domain. This is accomplished with the skip multiplexer **1312**. The multiplexer selects between registers that are clocked on the rising **1310** and falling **1311** edges of ClkC. The SkipT value determines which of the multiplexer paths is selected. The SkipT value is determined by sampling the TClkC1B clock by the ClkCD clock **1222**. The resulting value is loaded into a register **1309** by the LoadSkip signal **1238** during the initialization routine. This circuitry is described in detail later in this description.

The receive block **203** for the read data Q is shown at the bottom of FIG. **13**. The receive block has essentially the same elements as the transmit block that was just discussed, except that the flow of data is reversed. However, the clock domain crossing issues are fundamentally similar.

The RPhShA[i][n:0] control signals **1238** select one of the ClkC8[N:1] clock signals **1219** to pass through to the RClkC8 clock. The phase that is used is, in general, different for each receive slice on the controller. The phase is selected during a calibration process during initialization. This process is described in detail later in this description.

The RClkC8A clock passes through 1/8 **1327** and 1/1 **1326** frequency dividers before clocking the parallel **1320** and serial **1319** registers. Note that the ClkC8[N:1] signals **1219** that are distributed have a small phase offset to compensate for the delay of the phase offset selection block (PhShA) **1315** and the frequency divider blocks **1326** and **1327**. This offset is generated by a phase-locked-loop circuit and will track out supply voltage and temperature variations.

Even with the receive phase shift value set correctly (so that the bit windows on the Q bus are sampled properly), the phase of the RClkC1B clock used for the parallel register **1320** could be mismatched (there are eight possible combinations of phase). There are several ways of dealing with the problem. The scheme that is used in the embodiment illustrated provides an input RPhShB **1237**, such that when this input is pulsed, the phase of the RClkC1B clock will shift by $\frac{1}{8}^{th}$ of a cycle (45 degrees). The initialization software adjusts the phase of this clock until the parallel register **1320** loads the serial register **1319** at the proper time. This initialization process is described in detail later in this description.

A skip multiplexer similar to that described for the transmit circuit is used to move between the RClkC1B clock domain and the ClkC clock domain. After the phase shift controls are properly adjusted, the read data can be received from the Q bus **1230** and loaded into the parallel register **1320**. However, the read data still needs to be transferred

from the parallel register **1320** in the RClkC1B domain into the register **1323** in the ClkC **1215** domain. This is accomplished with the skip multiplexer **1322**. The multiplexer can insert or not insert a register **1321** that is clocked on the negative edge of ClkC in between registers that are clocked on the rising edges of RClkC1B **1320** and ClkC **1323**. The SkipR value determines which of the multiplexer paths is selected. The SkipR value is determined by sampling the RClkC1B clock by the ClkCD clock **1222**. The resulting value is loaded into a register **1318** by the LoadSkip signal **1238** during the initialization routine. This circuitry is described in detail later in this description.

FIG. **14** is a logic diagram illustrating details of the PLL used to generate the ClkC8 signal as illustrated in FIG. **12** in accordance with an embodiment of the invention. PLL **1204** comprises PLL circuit **1401**, adjustable matched delays **1402**, matched buffers **1403**, and phase comparator **1404**. PLL circuit **1401** comprises VCO **1405**, dummy phase offset selector **1406**, frequency divider **1407**, and phase comparator **1408**. ClkIn clock signal **1201** is provided to VCO **1405** and phase comparator **1408**. VCO **1405** provides an output to adjustable matched delays **1402** and matched buffers **1403**. Adjustable matched delays **1402** provide a plurality of incrementally delayed outputs to matched buffers **1403**.

The PLL circuit **1401** generates a clock signal that is 8 times the frequency of the input clock signal ClkIn **1201**, and the generated signal is also phase-shifted to account for delay expected to exist in the paths of the clock signals produced by the circuit in FIG. **14**. As such, expected delays are compensated for during the clock generation process such that the clock signals that appear at the point of actual use are correctly phase adjusted. The remaining portion of the block **1204** outside of the PLL circuit **1401** is used to generate equally phase-spaced versions of the clock produced by the PLL circuit **1401**. This is accomplished through well-known delay locked loop techniques where the delay locked loop provides the mechanism for generating the equally spaced clock signals. The clock signals produced as a result of the block **1204** in FIG. **14** are provided to the phase shifting logic described below with respect to FIG. **15**. The results of the clock generation performed by the circuits of FIGS. **14** and **15** are used to perform the serial-to-parallel or parallel-to-serial conversion as described in FIG. **13** above.

Output **1409** of matched buffers **1403**, which is not delayed by adjustable matched delays **1402**, is provided to an input of dummy phase offset selector **1406** and an input of phase comparator **1404** and provides the ClkC8 clock signal. Delayed output **1410** provides the $ClkC8_1$ clock signal. Delayed output **1411** provides the $ClkC8_2$ clock signal. Delayed output **1412** provides the $ClkC8_3$ clock signal. Delayed output **1413** provides the $ClkC8_{N-1}$ clock signal. Delayed output **1414** provides the $ClkC8_N$ clock signal, which is provided to an input of phase comparator **1404**. Phase comparator **1404** provides a feedback signal to adjustable matched delays **1402**, thereby providing a delay-locked loop (DLL). Each of the matched buffers **1403** has a substantially similar propagation delay, thereby providing a buffered output without introducing unintended timing skew among output **1409** and delayed outputs **1410–1414**.

The ClkIn reference clock **1201** is received and is frequency-multiplied by 8× by the PLL **1204**. Several delays are included with the PLL feedback loop of PLL **1204**, including a buffer delay introduced by matched buffers **1403**, a dummy phase offset selection delay introduced by dummy phase offset selector **1406**, and a frequency divider delay introduced by frequency divider **1407**. By including these delays in the feedback loop, the clock that is used for sampling and driving bits on the DQ will be matched to the ClkIn reference, and any delay variations caused by slow drift of temperature and supply voltage will be tracked out.

The output of the PLL circuit **1401** is then passed through a delay line **1402** with N taps. The delay of each element is identical, and can be adjusted over an appropriate range so that the total delay of N elements can equal one ClkC8 cycle ($t_{CC}/8$). There is a feedback loop **1404** that compares the phase of the undelayed ClkC8 to the clock with maximum delay ClkC8[N]. The delay elements are adjusted until their signals are phase aligned, meaning there is $t_{CC}/8$ of delay across the entire delay line.

The ClkC8[N:1] signals pass through identical buffers **1403** and see identical loads from the transmit and receive slices to which they connect. The ClkC8 reference signal **1409** also has a buffer and a matched dummy load to mimic the delay.

FIG. **15** is a block diagram illustrating how the ClkC8 [N:1] signals are used in the transmit and receive blocks of the memory controller component such as that illustrated in FIG. **13** in accordance with an embodiment of the invention. PhShA logic block **1501** comprises phase offset selection circuit **1502**, which comprises phase offset selector **1503**. Phase offset selector **1503** receives $ClkC8_1$ clock signal **1410**, $ClkC8_2$ clock signal **1411**, $ClkC8_3$ clock signal **1412**, $ClkC8_{N-1}$ clock signal **1413**, and $ClkC8_N$ clock signal **1414** (i.e., N variants of the ClkC8 clock signal) and selects and provides ClkC8A clock signal **1504**. This is accomplished using the N-to-1 multiplexer **1503** which selects one of the signals depending upon the setting of the control signals PhShA[i][n:0], where $N=2^{n+1}$. This allows the phase of the ClkC8A output clock for slice [i] to be varied across one ClkC8 cycle ($t_{CC}/8$) in increments of $t_{CC}/8N$.

At initialization, a calibration procedure is performed with software and/or hardware in which test bits are sampled and driven under each combination of the control signals PhShA[i][n:0]. The combination which yields the best margin is selected for each slice. This static value compensates for the flight time of the DQ and AC signals between the controller and the memory components. This flight time is mainly a factor of trace length and propagation velocity on printed wiring boards, and does not vary much during system operation. Other delay variations due to supply voltage and temperature are automatically tracked out by the feedback loops of the PLLs in the system.

FIG. **16** is a block diagram illustrating the PhShB circuit **1307** and **1316** of FIG. **13**. Clock conversion circuit **1601** of FIG. **16** preferably corresponds to 1/1 divider circuit **1324** and 1/1 divider circuit **1326** of FIG. **13**. Similarly, clock conversion circuit **1602** of FIG. **16** preferably corresponds to 1/8 divider circuit **1325** and 1/8 divider circuit **1327** of FIG. **13**. It produces a ClkC8B clock and a ClkC1B clock based on the ClkC8A clock in accordance with an embodiment of the invention. Clock conversion circuit **1601** comprises a multiplexer **1603**, which receives ClkC8A signal **1504** and provides ClkC8B signal **1604**. Clock conversion circuit **1602** comprises registers **1605**, **1606**, **1607**, and **1612**, logic gate **1608**, multiplexer **1611**, and incrementing circuits **1609** and **1610**. PhShB signal **1614** is applied to register **1605**, and ClkC8A clock signal **1504** is used to clock register **1605**. Outputs of register **1605** are applied as an input and a clock input to register **1606**. An output of register **1606** is applied as an input to register **1607** and logic gate **1608**. An output of register **1606** is used to clock register **1607**. An output of register **1607** is applied to logic gate **1608**. An output of

register **1607** is used to clock register **1612**. An output of logic gate **1608** is applied to multiplexer **1611**.

Incrementing circuit **1609** increments an incoming three-bit value by two. Incrementing circuit **1610** increments the incoming three-bit value by one in a binary manner such that it wraps from 111 to 000. Multiplexer **1611** selects among the three-bit outputs of incrementing circuits **1609** and **1610** and provides a three-bit output to register **1612**. Register **1612** provides a three-bit output to be used as the incoming three-bit value for incrementing circuits **1609** and **1610**. The most significant bit (MSB) of the three-bit output is used to provide ClkC1B clock signal **1613**.

In FIG. **16**, the ClkC8A clock that is produced by the PhShA (**1306** and **1315** of FIG. **13**) block is then used to produce a ClkC8B clock at the same frequency and to produce a ClkC1B clock at $\frac{1}{8}^{th}$ the frequency. These two clocks are phase aligned with one another (each rising edge of ClkC1B is aligned with a rising edge of ClkC8B.

ClkC1B **1613** is produced by passing it through a divide-by-eight counter **1602**. ClkC8A clocks a three bit register **1612** which increments on each clock edge. The most-significant bit will be ClkC1B, which is $\frac{1}{8}^{th}$ the frequency of ClkC8A. The ClkC8B **1604** clock is produced by a multiplexer which mimics the clock-to-output delay of the three bit register, so that ClkC1B and ClkC8B are aligned. As is apparent to one of ordinary skill in the art, other delaying means can be used in place of the multiplexer shown in block **1601** to accomplish the task of matching the delay through the divide-by-8 counter.

As described with respect to FIG. **13**, it is necessary to adjust the phase of ClkC1B **1613** so that the parallel register is loaded from/to the serial register in the transmit and receive blocks at the proper time. At initialization, a calibration procedure will transmit and receive test bits to determine the proper phasing of the ClkC1B clock. This procedure will use the PhShB control input **1614**. When this input has a rising edge, the three bit counter will increment by +2 instead of +1 on one of the following ClkC8A edges (after synchronization). The phase of the ClkC1B clock will shift $\frac{1}{8}^{th}$ of a cycle earlier. The calibration procedure will continue to advance the phase of the ClkC1B clock and check the position of test bits on the TD[i][7:0] and RQ[i][7:0] buses. When the test bits are in the proper position, the ClkC1B phase will be frozen.

FIG. **17** is a block diagram illustrating details of the PhShC block (**1240** in FIG. **12**) in accordance with an embodiment of the invention. PhShC block **1240** includes blocks **1701**–**1704**. Block **1701** comprises register **1705** and multiplexer **1706**. Write data input **1714** is provided to register **1705** and multiplexer **1706**. Register **1705** is clocked by ClkC clock signal **1215** and provides an output to multiplexer **1706**. Multiplexer **1706** receives TPhShC[0] selection input **1713** and provides write data output **1715**. Block **1702** comprises register **1707** and multiplexer **1708**. Read data input **1717** is provided to register **1707** and multiplexer **1708**. Register **1707** is clocked by ClkC clock signal **1215** and provides an output to multiplexer **1708**. Multiplexer **1708** receives RPhShC[0] selection input **1716** and provides read data output **1718**. Block **1703** comprises register **1709** and multiplexer **1710**. Write data input **1720** is provided to register **1709** and multiplexer **1710**. Register **1709** is clocked by ClkC clock signal **1215** and provides an output to multiplexer **1710**. Multiplexer **1710** receives TPhShC[31] selection input **1719** and provides write data output **1721**. Block **1704** comprises register **1711** and multiplexer **1712**. Read data input **1723** is provided to register **1711** and multiplexer **1712**. Register **1711** is clocked by ClkC clock signal **1215** and provides an output to multiplexer **1712**. Multiplexer **1712** receives RPhShC[31] selection input **1722** and provides read data output **1724**. While only two instances of the blocks for the write data and only two instances of the blocks for the read data are illustrated, it is understood that the blocks may be replicated for each bit of write data and each bit of read data.

The PhShC block **1240** is the final logic block that is used to adjust the delay of the 32x8 read data bits and the 32x8 write data bits so that all are driven or sampled from/to the same ClkC clock edge in the controller logic block. This is accomplished with an eight bit register which can be inserted into the path of the read and write data for each slice. The insertion of the delay is determined by the two control buses TPhShC[31:0] and RPhShC[31:0]. There is one control bit for each slice, since the propagation delay of the read and write data may cross a ClkC boundary at any memory slice position. Some systems with larger skews in the read and write data across the memory slices may need more than one ClkC of adjustment. The PhShC cells shown can be easily extended to provide additional delay by adding more registers and more multiplexer inputs.

The two control buses TPhShC[31:0] and RPhShC[31:0] are configured during initialization with a calibration procedure. As with the other phase-adjusting steps, test bits are read and written to each memory slice, and the control bits are set to the values that, in the example embodiment, allow all 256 read data bits to be sampled in one ClkC cycle and all 256 write data bits to be driven in one ClkC cycle by the controller logic.

FIG. **18** is a block diagram illustrating the logic details of the skip logic from the transmit block **203** (in FIG. **13**) of a memory controller component in accordance with an embodiment of the invention. The skip logic comprises registers **1801**, **1802**, **1803**, **1804**, and **1806**, and multiplexer **1805**. RClkC1B clock input **1807** is provided to register **1801** and is used to clock register **1803**. ClkCD clock input **1222** is used to clock register **1801**, which provides an output to register **1802**. Register **1802** receives LoadSkip signal **1238** and is clocked by ClkC clock signal **1215**, providing an output to multiplexer **1805** and an output used to clock registers **1804** and **1806**. Register **1803** receives data in domain RClkC1B at input **1808** and provides an output to register **1804** and multiplexer **1805**. Register **1804** provides an output to multiplexer **1805**. Multiplexer **1805** provides an output to register **1806**. Register **1806** provides data in domain ClkC at output **1809**.

The circuit transfers the data in the RClkC1B clock domain to the ClkC domain. These two clocks have the same frequency, but may have any phase alignment. The solution is to sample RClkC1B with a delayed version of ClkC called ClkCD (the limits on the amount of delay can be determined by the system, but in one embodiment, the nominal delay is ¼ of a ClkC cycle). This sampled value is called SkipR, and it determines whether the data in an RClkC1B register may be transferred directly to a ClkC register, or whether the data must first pass through a negative-edge-triggered ClkC register.

Regarding FIG. **18**, the following worst case setup constraints can be assumed:

Case B0

$$t_{D,MAX}+t_{H1,MIN}+t_{CL,MIN}+t_{V,MAX}+t_{M,MAX}+t_{S,MIN}<=t_{CYCLE}$$

or

$$t_{D,MAX}<=t_{CH,MIN}-t_{H1,MIN}-t_{V,MAX}-t_{M,MAX}-t_{S,MIN} \quad \text{**constraint S**}$$

Case D1

$$t_{D,MAX}+t_{H1,MIN}+t_{CYCLE}+t_{V,MAX}+t_{S,MIN}<=t_{CYCLE}+t_{CL,MIN}$$

or

$$t_{D,MAX}<=t_{CL,MIN}-t_{H1,MIN}-t_{V,MAX}-t_{S,MIN}$$

The following worst case hold constraints can be assumed:

Case A1

$$t_{D,MIN}-t_{S1,MIN}+t_{V,MIN}>=t_{H,MIN}$$

or

$$t_{D,MIN}>=t_{H,MIN}+t_{S1,MIN}-t_{V,MIN} \quad \text{**constraint H**}$$

Case C0

$$t_{D,MIN}-t_{S1,MIN}+t_{V,MIN}+t_{M,MIN}>=t_{H,MIN}$$

or

$$t_{D,MIN}>=t_{H,MIN}+t_{S1,MIN}-t_{V,MIN}-t_{M,MIN}$$

The timing parameters used above are defined as follow:

t_{S1}—Setup time for clock sampler
t_{H1}—Hold time for clock sampler
t_S—Setup time for data registers
t_H—Hold time for data registers
t_V—Valid delay (clock-to-output) of data registers
t_M—Propagation delay of data multiplexer
t_{CYCLE}—Clock cycle time (RClkC1B, ClkC, ClkCD)
t_{CH}—Clock high time (RClkC1B, ClkC, ClkCD)
t_{CL}—Clock low time (RClkC1B, ClkC, ClkCD)
t_D—Offset between ClkC and ClkCD (ClkCD is later)

Note:

$t_{D,NOM}=t_{CYCLE}/4$
$t_{CH,NOM}=t_{CYCLE}/2$
$t_{CL,NOM}=t_{CYCLE}/2$

FIG. **19** is a timing diagram illustrating the timing details of the skip logic of the receive block **203** (illustrated in FIG. **13**) in accordance with an embodiment of the invention. FIG. **19** illustrates waveforms of ClkCD clock signal **1901**, ClkC clock signal **1902**, RClkC1B (case A0) clock signal **1903**, RClkC1B (case A1) clock signal **1904**, RClkC1B (case B0) clock signal **1905**, RClkC1B (case B1) clock signal **1906**, RClkC1B (case C0) clock signal **1907**, RClkC1B (case C1) clock signal **1908**, RClkC1B (case D0) clock signal **1909**, and RClkC1B (case D1) clock signal **1910**. Times **1911**, **1912**, **1913**, **1914**, **1915**, **1916**, **1917**, and **1918**, at intervals of one clock cycle, are illustrated to indicate the timing differences between the clock signals.

FIG. **19** generally summarizes the possible phase alignments of RClkC1B and ClkC as eight cases labeled A0 through D1. These cases are distinguished by the position of the RClkC1B rising and falling edge relative to the set/hold window of the rising edge of ClkCD which samples RClkC1B to determine the SkipR value. Clearly, if the RClkC1B rising or falling edge is outside of this window, it will be correctly sampled. If it is at the edge of the window or inside the window, then it can be sampled as either a zero or one (i.e., the validity of the sample cannot be ensured). The skip logic has been designed such that it functions properly in either case, and this then determines the limits on the delay of the ClkCD clock t_D.

For the receive block, case B0 **1905** has the worst case setup constraint, and case A1 **1904** has the worst case hold constraint:

$$t_{D,MAX}<=t_{CH,MIN}-t_{H1,MIN}-t_{V,MAX}-t_{M,MAX}-t_{S,MIN} \quad \text{**constraint S**}$$

$$t_{D,MIN}>=t_{H,MIN}+t_{S1,MIN}-t_{V,MIN} \quad \text{**constraint H**}$$

As mentioned earlier, the nominal value of t_D (the delay of ClkCD relative to ClkC) is expected to be ¼ of a ClkC cycle. The value of t_D can vary up to the $t_{D,MAX}$ value shown above, or down to the $t_{D,MIN}$ value, also shown above. If the setup (e.g., t_{S1}, t_S), hold (e.g., t_{H1}, t_H), multiplexer propagation delay (e.g., t_M), and valid (e.g., t_V) times all went to zero, then the t_D value could vary up to $t_{CH,MIN}$ (the minimum high time of ClkC) and down to zero. However, the finite set/hold window of registers, and the finite clock-to-output (valid time) delay and multiplexer delay combine to reduce the permissible variation of the t_D value.

Note that it would be possible to change some of the elements of the skip logic without changing its basic function. For example, a sampling clock ClkCD may be used that is earlier rather than later (the constraint equations are changed, but there is a similar dependency of the timing skew range of ClkC to ClkCD upon the various set, hold, and valid timing parameters). In other embodiments, a negative-edge-triggered RClkC1B register is used instead of a ClkC register into the domain-crossing path (again, the constraint equations are changed, but a similar dependency of the timing skew range of ClkC to ClkCD upon the various set, hold, and valid timing parameters remains).

Finally, it should be noted that the skip value that is used is preferably generated once during initialization and then loaded (with the LoadSkip control signal) into a register. Such a static value is preferable to rather than one that is sampled on every ClkCD edge because if the alignment of RClkC1B is such that it has a transition in the set/hold window of the ClkCD sampling register, it may generate different skip values each time it is sampled. This would not affect the reliability of the clock domain crossing (the RClkC1B date would be correctly transferred to the ClkC register), but it would affect the apparent latency of the read data as measured in ClkC cycles in the controller. That is, sometimes the read data would take a ClkC cycle longer than at other times. Sampling the skip value and using it for all domain crossings solves this problem. Also note that during calibration, every time the RClkC1B phase is adjusted, the LoadSkip control is pulsed in case the skip value changes.

FIG. **20** is a block diagram illustrating the logic details of the skip logic of the transmit block **202** of FIG. **13** in accordance with an embodiment of the invention. The skip logic comprises registers **2001**, **2002**, **2003**, **2004**, and **2006**, and multiplexer **2005**. TClkC1B clock input **2007** is provided to register **2001** and is used to clock register **2006**. ClkCD clock input **1222** is used to clock register **2001**, which provides an output to register **2002**. Register **2002** receives LoadSkip signal **1238** and is clocked by ClkC clock signal **1215**, providing an output to multiplexer **2005** and an output used to clock registers **2003** and **2004**. Register **2003** receives data in domain ClkC at input **2008** and provides an output to register **2004** and multiplexer **2005**. Register **2004** provides an output to multiplexer **2005**. Multiplexer **2005** provides an output to register **2006**. Register **2006** provides data in domain TClkC1B at output **2009**.

The circuit of FIG. **20** is used in the transfer of data in the ClkC clock domain to the TClkC1B domain. The two clocks ClkC and TClkC1B have the same frequency, but may be phase mismatched. One technique that can be used in the

clock domain crossing is to sample TClkC1B with a delayed version of ClkC called ClkCD (the limits on the amount of delay can vary, but in one embodiment, the delay selected is ¼ of a ClkC cycle). The sampled value, SkipT, determines whether the data in a ClkC register is transferred directly to a TClkC1B register, or whether the data first passes through a negative-edge-triggered ClkC register.

Regarding FIG. **20**, the following worst case setup constraints can be assumed:

Case C0

$$t_{D,MIN} - t_{S1,MIN} >= t_{V,MAX} + t_{M,MAX} + t_{S,MIN}$$

or

$$t_{D,MIN} >= t_{S1,MIN} + t_{V,MAX} + t_{M,MAX} + t_{S,MIN} \quad \text{**constraint S**}$$

Case A1

$$t_{D,MIN} - t_{S1,MIN} >= t_{V,MAX} + t_{S,MIN}$$

or

$$t_{D,MIN} >= t_{S1,MIN} + t_{V,MAX} + t_{S,MIN}$$

The following worst case hold constraints can be assumed:

Case D1

$$t_{H,MIN} <= t_{CH,MIN} - t_{D,MAX} - t_{H1,MIN} - t_{V,MIN}$$

or

$$t_{D,MAX} <= t_{CH,MIN} - t_{H1,MIN} - t_{V,MIN} - t_{H,MIN}$$

or

$$t_{D,MAX} <= t_{CL,MIN} - t_{H1,MIN} - t_{V,MIN} - t_{M,MIN} - t_{H,MIN}$$

Case B0

$$t_{H,MIN} <= t_{CL,MIN} - t_{D,MAX} - t_{H1,MIN} - t_{V,MIN} - t_{M,MIN}$$

or

$$t_{D,MAX} <= t_{CL,MIN} - t_{H1,MIN} - t_{V,MIN} - t_{M,MIN} - t_{H,MIN} \quad \text{**constraint H**}$$

Definitions for the timing parameters used above may be found in the discussion of FIG. **18** above.

FIG. **21** is a timing diagram illustrating the timing details of the skip logic of the transmit block **202** of FIG. **13** in accordance with an embodiment of the invention. FIG. **21** illustrates waveforms of ClkCD clock signal **2101**, ClkC clock signal **2102**, TClkC1B (case A0) clock signal **2103**, TClkC1B (case A1) clock signal **2104**, TClkC1B (case B0) clock signal **2105**, TClkC1B (case B1) clock signal **2106**, TClkC1B (case C0) clock signal **2107**, TClkC1B (case C1) clock signal **2108**, TClkC1B (case D0) clock signal **2109**, and TClkC1B (case D1) clock signal **2110**. Times **2111**, **2112**, **2113**, **2114**, **2115**, **2116**, **2117**, and **2118**, at intervals of one clock cycle, are illustrated to indicate the timing differences between the clock signals.

FIG. **21** generally summarizes the possible phase alignments of TClkC1B and ClkC as eight cases labeled A0 through D1. These cases are distinguished by the position of the TClkC1B rising and falling edge relative to the set/hold window of the rising edge of ClkCD which samples TClkC1B to determine the SkipR value. Clearly, if the TClkC1B rising or falling edge is outside of this window, it will be correctly sampled. If it is at the edge of the window or inside the window, then it can be sampled as either a zero or one (i.e., the validity of the sample cannot be ensured). The skip logic has been designed such that it functions properly in either case, and this then determines the limits on the delay of the ClkCD clock t_D.

For the transmit block, case C0 **2107** has the worst case setup constraint, and case B0 **2105** has the worst case hold constraint:

$$t_{D,MIN} >= t_{S1,MIN} + t_{V,MAX} + t_{M,MAX} + t_{S,MIN} \quad \text{**constraint S**}$$

$$t_{D,MAX} <= t_{CL,MIN} - t_{H1,MIN} - t_{V,MIN} - t_{M,MIN} - t_{H,MIN} \quad \text{**constraint H**}$$

As mentioned earlier, the nominal value of t_D (the delay of ClkCD relative to Clkc) will by ¼ of a ClkC cycle. This can vary up to the $t_{D,MAX}$ value shown above, or down to the $t_{D,MIN}$ value. If the set, hold, mux (i.e., multiplexer), and valid times all went to zero, then the t_D value could vary up to $t_{CH,MIN}$ (the minimum high time of ClkC) and down to zero. However, the finite set/hold window of registers, and the finite clock-to-output (valid time) delay and multiplexer delay combine to reduce the permissible variation of the t_D value.

As described with respect to FIG. **19** above, some elements of the skip logic can be changed for different embodiments while preserving its general functionality. Similarly, as described with respect to the skip logic of FIG. **19**, the skip value that is used is preferably generated during initialization and then loaded (with the LoadSkip control signal) into a register.

FIG. **22** is a timing diagram illustrating an example of a data clocking arrangement in accordance with an embodiment of the invention. However, in this example, the clock phases in the memory controller and memory components have been adjusted to a different set of values than in the example illustrated in FIGS. **5** though **21**. The waveforms of $WClk_{S1,M0}$ clock signal **2201** and $RClk_{S1,M0}$ clock signal **2202** are illustrated to show the data timing for slice **1** from the perspective of the memory controller component at slice **0**. The rising edges of sequential cycles of $WClk_{S1,M0}$ clock signal **2201** occur at times **2205**, **2206**, **2207**, **2208**, **2209**, **2210**, **2211**, and **2212**, respectively. Write datum information Da **2213** is present on the data lines at the controller at time **2205**. Read datum information Qb **2204** is present at time **2208**. Read datum information Qc **2215** is present at time **2209**. Write datum information Dd **2216** is present at time **2210**. Write datum information De **2217** is present at time **2211**.

The waveforms of $WClk_{S1,M1}$ clock signal **2203** and $RClk_{S1,M1}$ clock signal **2204** are illustrated to show the data timing for slice **1** from the perspective of the memory component at slice **1**. Write datum information Da **2218** is present on the data lines at the memory component at time **2206**. Read datum information Qb **2219** is present at time **2207**. Read datum information Qc **2220** is present at time **2208**. Write datum information Dd **2221** is present at time **2211**. Write datum information De **2222** is present at time **2212**.

The examplary system illustrated in FIGS. **5** through **21** assumed that the clock for the read and write data were in phase at each memory component. FIG. **22** assumes that for each slice the read clock at each memory component is in phase with the write clock at the controller ($RClk_{S1,M0}=WClk_{S1,M1}$), and because the propagation delay t_{PD2} is the same in each direction, the write clock at each memory component is in phase with the read clock at the controller ($WClk_{S1,M0}=RClk_{S1,M1}$). This phase relationship shifts the timing slots for the read and write data relative to FIG. **6**, but does not change the fact that two idle cycles are inserted during a write-read-read-write sequence. The phase relationship alters the positions within the system where domain

crossings occur (some domain crossing logic moves from the controller into the memory components).

FIGS. **23** through **26** are timing diagrams illustrating an example of a data clocking arrangement in accordance with an embodiment of the invention. However, in this example the clock phases in the memory controller and memory components have been adjusted to a different set of values than those in the example illustrated in FIGS. **5** though **21**. The example in FIGS. **23** through **26** also uses a different set of clock phase values than the example in FIG. **22**.

FIG. **23** is a timing diagram illustrating an example of a data clocking arrangement in accordance with an embodiment of the invention. The waveforms of $WClk_{S1,M0}$ clock signal **2301** and $RClk_{S1,M0}$ clock signal **2302** are illustrated to show the data timing for slice **1** from the perspective of the memory controller component at slice **0**. Rising edges of sequential cycles of $WClk_{S1,M0}$ clock signal **2301** occur at times **2305**, **2306**, **2307**, and **2308**, respectively. Write datum information Da **2309** is present on the data bus at the controller during a first cycle of $WClk_{S1,M0}$ clock signal **2301**. Read datum information Qb **2310** is present at a fourth cycle of $WClk_{S1,M0}$ clock signal **2301**. Read datum information Qc **2311** is present at time **2305**. Write datum information Dd **2312** is present at time **2306**. Write datum information De **2313** is present at time **2307**.

The waveforms of $WClk_{S1,M1}$ clock signal **2303** and $RClk_{S1,M1}$ clock signal **2304** are illustrated to show the data timing for slice **1** from the perspective of the memory component at slice **1**. Write datum information Da **2314** is advanced one clock cycle relative to its position from the perspective of the memory controller component at slice **0**. In other words, the write data appears on the data bus at the memory device approximately one clock cycle later than when it appears on the data bus at the controller. Read datum information Qb **2315** is delayed one clock cycle relative to its position from the perspective of the memory controller component at slice **0**. Read datum information Qc **2316** is also delayed one clock cycle relative to its position from the perspective of the memory controller component at slice **0**. Write datum information Dd **2317** is present at time **2317**. Write datum information De **2318** is present at time **2308**.

The example system assumes that the clock for the read and write data are in phase at each memory component. FIG. **23** assumes that for each slice the read clock and write clock are in phase at the controller ($RClk_{Si,M0}$=$WClk_{Si,M0}$), and also that each slice is in phase with every other slice at the controller ($WClk_{Si,M0}$=$WClk_{Sj,M0}$). This shifts the timing slots for the read and write data relative to FIG. **6** and FIG. **22**, but it does not change the fact that two idle cycles are used during a write-read-read-write sequence. The phase relationship alters the positions within the system where domain crossings occur (all the domain crossing logic moves from the controller into the memory components).

FIG. **6** represents the case in which all three clock phases (address, read data, and write data) are made the same at each memory component, FIG. **23** represents the case in which all three clock phases (address, read data, and write data) are made the same at the memory controller, and FIG. **22** represents one possible intermediate case. This range of cases is shown to emphasize that various embodiments of the invention may be implemented with various phasing. The memory controller and memory components can be readily configured to support any combination of clock phasing.

The one extreme case in which all three clock phases (address, read data, and write data) are made the same at each memory component (illustrated in FIGS. **5** through **21**) is important because there is a single clock domain within each memory component. The other extreme case in which all three clock phases (address, read data, and write data) are made the same at the memory controller (FIG. **23**) is also important because there is a single clock domain within the controller. FIGS. **24** through **26** further illustrate this case.

FIG. **24** is a timing diagram illustrating timing at the memory controller component for the example of the data clocking arrangement illustrated in FIG. **23** in accordance with an embodiment of the invention. The waveforms of $AClk_{S0,M1}$ clock signal **2401** are illustrated to show the address/control timing for memory module one from the perspective of the memory controller component at slice **0**. The rising edges of sequential cycles of $AClk_{S0,M1}$ clock signal **2401** occur at times **2406**, **2407**, **2408**, **2409**, **2410**, **2411**,**2412**, and **2413**, respectively. Address information ACa **2414** is present on the address signal lines at the controller at time **2406**. Address information ACb **2415** is present at time **2407**. Address information ACc **2416** is present at time **2408**. Address information ACd **2417** is present at time **2412**.

The waveforms of $WClk_{S1,M0}$ clock signal **2402** and $RClk_{S1,M0}$ clock signal **2403** are illustrated to show the data timing for slice **1** from the perspective of the memory controller component at module **0**. Write datum information Da **2418** is present on the data lines at the controller at time **2407**. Read datum information Qb **2419** is present at time **2411**. Read datum information Qc **2420** is present at time **2412**. Write datum information Dd **2421** is present at time **2413**.

The waveforms of $WClk_{SNs,M0}$ clock signal **2404** and $RClk_{SNs,M0}$ clock signal **2405** are illustrated to show the data timing for slice N_S from the perspective of the memory controller component at module **0**. Write datum information Da **2422** is present on the data lines at the controller at time **2407**. Read datum information Qb **2423** is present at time **2411**. Read datum information Qc **2424** is present at time **2412**. Write datum information Dd **2425** is present at time **2413**.

FIGS. **24** through **26** show the overall system timing for the case in which all clock phases are aligned at the controller. FIG. **24** is the timing at the controller, and is analogous to FIG. **7**, except for the fact that the clocks are all common at the controller instead of at each memory slice. As a result, the clocks are all aligned in FIG. **24**, and the two-cycle gap that the controller inserts into the write-read-read-write sequence is apparent between address packets ACc and ACd.

FIG. **25** is a timing diagram illustrating timing at a first slice of a rank of memory components for the example of the data clocking arrangement illustrated in FIG. **23** in accordance with an embodiment of the invention. The waveforms of $AClk_{S1,M1}$ clock signal **2501** is illustrated to show the address/control timing for memory module one from the perspective of the memory component at slice **1**. Times **2504**, **2505**, **2506**, **2507**, **2508**, **2509**, **2510**, and **2511** correspond to times **2406**, **2407**, **2408**, **2409**, **2410**, **2411**,**2412**, and **2413**, respectively, of FIG. **24**. Signal $AClk_{S1,M1}$ **2501** is delayed by a delay of t_{PD0} relative to signal $AClk_{S0,M1}$ **2401** in FIG. **24**. In other words, the AClk signal takes a time period t_{PD0} to propagate from the controller to the memory component. Address information ACa **2512** is associated with edge **2530** of signal **2501**. Address information ACb **2513** is associated with edge **2531** of signal **2501**. Address information ACc **2514** is associated with edge **2532** of signal **2501**. Address information ACd **2515** is associated with edge **2533** of signal **2501**.

The waveforms of $WClk_{S1,M1}$ clock signal **2502** and $RClk_{S1,M1}$ clock signal **2503** are illustrated to show the data timing for slice **1** from the perspective of the memory component at module **1**. FIG. **25** shows the timing at the first memory component (slice **1**), and the clocks have become misaligned because of the propagation delays t_{PD2} and t_{PD0}. Signal $WClk_{S1,M1}$ **2502** is delayed by a delay of t_{PD2} relative to signal $WClk_{S1,M0}$ **2402** of FIG. **24**. Write datum information Da **2516** is associated with edge **2534** of signal **2502**. Write datum information Dd **2519** is associated with edge **2537** of signal **2502**. Signal $RClk_{S1,M1}$ **2503** precedes by t_{PD2} signal $RClk_{S1,M0}$ **2403** of FIG. **24**. Read datum information Qb **2517** is associated with edge **2535** of signal **2503**. Read datum information Qc **2518** is associated with edge **2536** of signal **2503**.

FIG. **26** is a timing diagram illustrating timing a last slice of a rank of memory components for the example of the data clocking arrangement illustrated in FIG. **23** in accordance with an embodiment of the invention. The waveforms of $AClk_{SNs,M1}$ clock signal **2601** are illustrated to show the address/control timing for memory module one from the perspective of the memory component at slice N_S. Times **2604**, **2605**, **2606**, **2607**, **2608**, **2609**, **2610**, and **2611** correspond to times **2406**, **2407**, **2408**, **2409**, **2410**, **2411**,**2412**, and **2413**, respectively, of FIG. **24**. Signal $AClk_{SNs,M1}$ **2601** is delayed by a delay of $t_{PD0}+t_{PD1}$ relative to signal $AClk_{S0,M1}$ **2401** of FIG. **24**. In other words, address information ACa **2612** is associated with edge **2630** of signal **2601**. Address information ACb **2613** is associated with edge **2631** of signal **2601**. Address information ACc **2614** is associated with edge **2632** of signal **2601**. Address information ACd **2615** is associated with edge **2633** of signal **2601**.

The waveforms of $WClk_{SNs,M1}$ clock signal **2602** and $RClk_{SNs,M1}$ clock signal **2603** are illustrated to show the data timing for slice N_S from the perspective of the memory component at module **1**. Signal $WClk_{SNs,M1}$ **2602** is delayed by a delay of t_{PD2} relative to signal $WClk_{S1,M0}$ **2402** of FIG. **24**. Write datum information Da **2616** is associated with edge **2634** of signal **2602** (e.g., the write datum information Da **2616** is present on the data bus at the memory component when edge **2634** of signal **2602** is present on the AClk clock conductor at the memory component). Write datum information Dd **2619** is associated with edge **2637** of signal **2602**. Signal $RClk_{SNs,M1}$ **2603** precedes by t_{PD2} signal $RClk_{S1,M0}$ **2603** of FIG. **24**. Read datum information Qb **2617** is associated with edge **2635** of signal **2603**. Read datum information Qc **2618** is associated with edge **2636** of signal **2603**.

FIG. **26** shows the timing at the last memory component (slice N_S), and the clocks have become further misaligned because of the propagation delays t_{PD1}. As a result, each memory component will have domain crossing hardware similar to that which is in the controller, as described with respect to FIGS. **12–21**.

As a reminder, the example system described in FIG. **2** included single memory module, a single rank of memory components on that module, a common address and control bus (so that each controller pin connects to a pin on each of two or more memory components), and a sliced data bus (wherein each controller pin connects to a pin on exactly one memory component). These characteristics were chosen for the example embodiment in order to simplify the discussion of the details and because this configuration is an illustrative special case. However, the clocking methods that have been discussed can be extended to a wider range of system topologies. Thus, it should be understood that embodiments of the invention may be practiced with systems having features that differ from the features of the example system of FIG. **2**.

The rest of this discussion focuses on systems with multiple memory modules or multiple memory ranks per module (or both). In these systems, each data bus wire connects to one controller pin and to one pin on each of two or more memory components. Since the t_{PD2} propagation delay between the controller and each of the memory components will be different, the clock domain crossing issue in the controller becomes more complicated. If the choice is made to align all clocks in each memory component, then the controller will need a set of domain crossing hardware for each rank or module of memory components in a slice. This suffers from a drawback in that it requires a large amount of controller area and that it adversely affects critical timing paths. As such, in a multiple module or multiple rank system, it may be preferable to keep all of the clocks aligned in the controller, and to place the domain crossing logic in the memory components.

FIG. **27** is a block diagram illustrating a memory system that includes multiple ranks of memory components and multiple memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component **2702**, memory module **2703**, memory module **2730**, write clock **2705**, read clock **2706**, write clock **2726**, read clock **2727**, splitting component **2742**, splitting component **2743**, termination component **2720**, termination component **2724**, termination component **2737**, and termination component **2740**. It should be understood that there is at least one write clock per slice in the example system shown.

Within each memory module, memory components are organized in ranks. A first rank of memory module **2703** includes memory components **2716**, **2717**, and **2718**. A second rank of memory module **2703** includes memory components **2744**, **2745**, and **2746**. A first rank of memory module **2730** includes memory components **2731**, **2732**, and **2733**. A second rank of memory module **2730** includes memory components **2734**, **2735**, and **2736**.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice **2713**, slice **2714**, and slice **2715**. Each slice comprises one memory component of each rank. In this embodiment, each slice within each memory module is provided with its own data bus **2708**, write clock conductor **2710**, and read clock conductor **2711**. Data bus **2708** is coupled to memory controller component **2702**, memory component **2716**, and memory component **2744**. A termination component **2720** is coupled to data bus **2708** near memory controller component **2702**, and may, for example, be incorporated into memory controller component **2702**. A termination component **2721** is coupled near an opposite terminus of data bus **2708**, and is preferably provided within memory module **2703**. Write clock **2705** is coupled to write clock conductor **2710**, which is coupled to memory controller component **2702** and to memory components **2716** and **2744**. A termination component **2723** is coupled near a terminus of write clock conductor **2710** near memory components **2716** and **2744**, preferably within memory module **2703**. Read clock **2706** is coupled to read clock conductor **2711**, which is coupled through splitting component **2742** to memory controller component **2702** and memory components **2716** and **2744**. Splitting components are described in additional detail below. A termination component **2724** is coupled near memory controller component **2702**, and may, for example, be incorporated into

memory controller component 2702. A termination component 2725 is coupled near a terminus of read clock conductor 2711 near memory components 2716 and 2744, preferably within memory module 2703.

Slice 2713 of memory module 2730 is provided with data bus 2747, write clock conductor 2728, read clock conductor 2729. Data bus 2747 is coupled to memory controller component 2702, memory component 2731, and memory component 2734. A termination component 2737 is coupled to data bus 2747 near memory controller component 2702, and may, for example, be incorporated into memory controller component 2702. A termination component 2738 is coupled near an opposite terminus of data bus 2747, and is preferably provided within memory module 2730. Write clock 2726 is coupled to write clock conductor 2728, which is coupled to memory controller component 2702 and to memory components 2731 and 2734. A termination component 2739 is coupled near a terminus of write clock conductor 2728 near memory components 2731 and 2734, preferably within memory module 2730. Read clock 2727 is coupled to read clock conductor 2729, which is coupled through splitting component 2743 to memory controller component 2702 and memory components 2731 and 2734. A termination component 2740 is coupled near memory controller component 2702, and may, for example, be incorporated into memory controller component 2702. A termination component 2741 is coupled near a terminus of read clock conductor 2729 near memory components 2731 and 2734, preferably within memory module 2730.

The sliced data bus can be extended to multiple ranks of memory component and multiple memory components in a memory system. In this example, there is a dedicated data bus for each slice of each module. Each data bus is shared by the ranks of memory devices on each module. It is preferable to match the impedances of the wires as they transition from the main printed wiring board onto the modules so that they do not differ to an extent that impairs performance. In some embodiments, the termination components are on each module. A dedicated read and write clock that travels with the data is shown for each data bus, although these could be regarded as virtual clocks; i.e. the read and write clocks could be synthesized from the address/control clock as in the example system that has already been described.

FIG. 28 is a block diagram illustrating a memory system that includes multiple ranks of memory components and multiple memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component 2802, memory module 2803, memory module 2830, write clock 2805, read clock 2806, splitting component 2842, splitting component 2843, splitting component 2848, splitting component 2849, splitting component 2850, splitting component 2851, termination component 2820, termination component 2824, termination component 2880, and termination component 2881.

Within each memory module, memory components are organized in ranks. A first rank of memory module 2803 includes memory components 2816, 2817, and 2818. A second rank of memory module 2803 includes memory components 2844, 2845, and 2846. A first rank of memory module 2830 includes memory components 2831, 2832, and 2833. A second rank of memory module 2830 includes memory components 2834, 2835, and 2836.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice 2813, slice 2814, and slice 2815. Each slice comprises one memory component of each rank. In this embodiment, each slice across multiple memory modules is provided with a data bus 2808, write clock conductor 2810, and read clock conductor 2811. Data bus 2808 is coupled to memory controller component 2802, via splitter 2848 to memory components 2816 and 2844, and via splitter 2849 to memory components 2831 and 2834. A termination component 2820 is coupled to data bus 2808 near memory controller component 2802, and may, for example, be incorporated into memory controller component 2802. A termination component 2880 is coupled near an opposite terminus of data bus 2808, near splitter 2849. A termination component 2821 is coupled near memory components 2816 and 2844 and is preferably provided within memory module 2803. A termination component 2838 is coupled near memory components 2831 and 2834 and is preferably provided within memory module 2830.

Write clock 2805 is coupled to write clock conductor 2810, which is coupled to memory controller component 2802, via splitter 2850 to memory components 2816 and 2844, and via splitter 2851 to memory components 2831 and 2834. A termination component 2881 is coupled near a terminus of write clock conductor 2810, near splitter 2851. A termination component 2823 is coupled near memory components 2816 and 2844, preferably within memory module 2803. A termination component 2839 is coupled near memory components 2831 and 2834, preferably within memory module 2830.

Read clock 2806 is coupled to read clock conductor 2811, which is coupled through splitting component 2843 to memory components 2831 and 2834 and through splitting component 2842 to memory controller component 2802 and memory components 2816 and 2844. A termination component 2824 is coupled near memory controller component 2802, and may, for example, be incorporated into memory controller component 2802. A termination component 2825 is coupled near a terminus of read clock conductor 2811 near memory components 2816 and 2844, preferably within memory module 2803. A termination component 2841 is coupled near a terminus of read clock conductor 2811 near memory components 2831 and 2834, preferably within memory module 2830.

As illustrated, this example utilizes a single data bus per data slice that is shared by all the memory modules, as in FIG. 28. In this example, each data wire is tapped using some form of splitting component S. This splitter could be a passive impedance matcher (three resistors in a delta- or y-configuration) or some form of active buffer or switch element. In either case, the electrical impedance of each wire is maintained down its length (within manufacturing limits) so that signal integrity is kept high. As in the previous configuration, each split-off data bus is routed onto a memory module, past all the memory components in the slice, and into a termination component.

FIG. 29 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component 2902, memory module 2903, memory module 2930, write clock 2905, read clock 2906, termination component 2920, termination component 2921, termination component 2923, and termination component 2924.

Within each memory module, memory components are organized in ranks. A first rank of memory module 2903 includes memory components 2916, 2917, and 2918. A second rank of memory module 2903 includes memory components 2944, 2945, and 2946. A first rank of memory

module 2930 includes memory components 2931, 2932, and 2933. A second rank of memory module 2930 includes memory components 2934, 2935, and 2936.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice 2913, slice 2914, and slice 2915. Each slice comprises one memory component of each rank. In this embodiment, each slice across memory modules shares a common daisy-chained data bus 2908, a common daisy-chained write clock conductor 2910, and a common daisy-chained read clock conductor 2911. Data bus 2908 is coupled to memory controller component 2902, memory component 2916, memory component 2944, memory component 2931, and memory component 2934. A termination component 2920 is coupled to data bus 2908 near memory controller component 2902, and may, for example, be incorporated into memory controller component 2902. A termination component 2921 is coupled near an opposite terminus of data bus 2908.

Write clock 2905 is coupled to write clock conductor 2910, which is coupled to memory controller component 2902 and to memory components 2916, 2944, 2931, and 2934. A termination component 2923 is coupled near a terminus of write clock conductor 2910. Read clock 2906 is coupled to read clock conductor 2911, which is coupled to memory controller component 2902 and memory components 2916, 2944, 2931, and 2934. A termination component 2924 is coupled near memory controller component 2902, and may, for example, be incorporated into memory controller component 2902.

In this embodiment, there is a single data bus per data slice, but instead of using splitting components, each data wire is routed onto a memory module, past all the memory components of the slice, and back off the module and onto the main board to "chain" through another memory module or to pass into a termination component. The same three configuration alternatives described above with respect to the data bus are also applicable to a common control/address bus in a multi-module, multi-rank memory system.

FIG. 30 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a dedicated control/address bus per memory module in accordance with an embodiment of the invention. The memory system comprises memory controller component 3002, memory module 3003, memory module 3030, address/control clock 3004, address/control clock 3053, termination component 3052, and termination component 3056.

Within each memory module, memory components are organized in ranks. A first rank of memory module 3003 includes memory components 3016, 3017, and 3018. A second rank of memory module 3003 includes memory components 3044, 3045, and 3046. A first rank of memory module 3030 includes memory components 3031, 3032, and 3033. A second rank of memory module 3030 includes memory components 3034, 3035, and 3036.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice 3013, slice 3014, and slice 3015. Each slice comprises one memory component of each rank. In this embodiment, each memory module is provided with its own address bus 3007 and address/control clock conductor 3010. Address bus 3007 is coupled to memory controller component 3002 and memory components 3016, 3017, 3018, 3044, 3045, and 3046. A termination component 3052 is coupled to address bus 3007 near memory controller component 3002, and may, for example, be incorporated into memory controller component 3002. A termination component 3019 is coupled near an opposite terminus of address bus 3007, and is preferably provided within memory module 3003. Address/control clock 3004 is coupled to address/control clock conductor 3009, which is coupled to memory controller component 3002 and to memory components 3016, 3017, 3018, 3044, 3045, and 3046. A termination component 3022 is coupled near a terminus of address/control clock conductor 3009, preferably within memory module 3003.

Memory module 3030 is provided with address bus 3054 and address/control clock conductor 3055. Address bus 3054 is coupled to memory controller component 3002 and to memory components, 3031, 3032, 3033, 3034, 3035, and 3036. A termination component 3056 is coupled to address bus 3054 near memory controller component 3002, and may, for example, be incorporated into memory controller component 3002. A termination component 3057 is coupled near an opposite terminus of address bus 3054 and is preferably provided within memory module 3030. Address/control clock 3053 is coupled to address/control clock conductor 3055, which is coupled to memory controller component 3002 and to memory components 3031, 3032, 3033, 3034, 3035, and 3036. A termination component 3058 is coupled near a terminus of address/control clock conductor 3055, preferably within memory module 3030.

Each control/address wire is routed onto a memory module, past all the memory components, and into a termination component. The wire routing is shown in the direction of the ranks on the module, but it could also be routed in the direction of slices.

FIG. 31 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared among the memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component 3102, memory module 3103, memory module 3130, address/control clock 3104, splitting component 3159, splitting component 3160, splitting component 3161, splitting component 3162, termination component 3163, and termination component 3164.

Within each memory module, memory components are organized in ranks. A first rank of memory module 3103 includes memory components 3116, 3117, and 3118. A second rank of memory module 3103 includes memory components 3144, 3145, and 3146. A first rank of memory module 3130 includes memory components 3131, 3132, and 3133. A second rank of memory module 3130 includes memory components 3134, 3135, and 3136.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice 3113, slice 3114, and slice 3115. Each slice comprises one memory component of each rank. In this embodiment, an address bus 3107 and an address/control clock conductor 3109 are coupled to each memory component among multiple memory modules. Address bus 3107 is coupled to memory controller component 3102, via splitter 3159 to memory components 3116, 3117, 3118, 3144, 3145, and 3146, and via splitter 3161 to memory components 3131, 3132, 3133, 3134, 3135, and 3136. A termination component 3152 is coupled to address bus 3107 near memory controller component 3102, and may, for example, be incorporated into memory controller component 3102. A termination component 3163 is coupled near an opposite terminus of address bus 3107, near splitter 3161. A termination component 3119 is coupled to address bus 3107, preferably within memory module 3103. A termina-

tion component 3157 is coupled to address bus 3107, preferably within memory module 3130.

Address/control clock 3104 is coupled to address/control clock conductor 3109, which is coupled to memory controller component 3102, via splitter 3160 to memory components 3116, 3117, 3118, 3144, 3145, and 3146, and via splitter 3162 to memory components 3131, 3132, 3133, 3134, 3135, and 3136. A termination component 3164 is coupled near a terminus of address/control clock conductor 3109, near splitter 3162. A termination component 3122 is coupled to the address/control clock conductor 3109, preferably within memory module 3103. A termination component 3158 is coupled to the address/control clock conductor 3109, preferably within memory module 3130.

In this example, each control/address wire is tapped using some form of splitting component S. This splitter could be a passive impedance matcher (three resistors in a delta- or y-configuration) or some form of active buffer or switch element. In either case, the electrical impedance of each wire is maintained down its length (within manufacturing limits) so that signal integrity is kept high. As in the previous configuration, each split-off control/address bus is routed onto a memory module, past all the memory components, and into a termination component.

FIG. 32 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared by all the memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component 3202, memory module 3203, memory module 3230, address/control clock 3204, termination component 3219, and termination component 3222.

Within each memory module, memory components are organized in ranks. A first rank of memory module 3203 includes memory components 3216, 3217, and 3218. A second rank of memory module 3203 includes memory components 3244, 3245, and 3246. A first rank of memory module 3230 includes memory components 3231, 3232, and 3233. A second rank of memory module 3230 includes memory components 3234, 3235, and 3236.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice 3213, slice 3214, and slice 3215. Each slice comprises one memory component of each rank. In this embodiment, the memory components of the memory modules share a common daisy-chained address bus 3207 and a common daisy-chained address/control clock conductor 3209. Address bus 3207 is coupled to memory controller component 3202 and memory components 3216, 3217, 3218, 3244, 3245, 3246, 3231, 3232, 3233, 3234, 3235, and 3236. A termination component 3252 is coupled to address bus 3207 near memory controller component 3202, and may, for example, be incorporated into memory controller component 3202. A termination component 3219 is coupled near an opposite terminus of address bus 3207.

Address/control clock 3204 is coupled to address/control clock conductor 3209, which is coupled to memory controller component 3202 and to memory components 3216, 3217, 3218, 3244, 3245, 3246, 3231, 3232, 3233, 3234, 3235, and 3236. A termination component 3222 is coupled near a terminus of address/control clock conductor 3209.

Unlike the memory system of FIG. 31, instead of using some kind of splitting component, each control/address wire is routed onto a memory module, past all the memory components, and back off the module and onto the main board to chain through another memory module or to pass into a termination component.

The same three configuration alternatives are possible for a sliced control/address bus in a multi-module, multi-rank memory system. This represents a departure from the systems that have been discussed up to this point—the previous systems all had a control/address bus that was common across the memory slices. It is also possible to instead provide an address/control bus per slice. Each bus is preferably routed along with the data bus for each slice, and preferably has the same topological characteristics as a data bus which only performs write operations.

FIG. 33 is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a dedicated, sliced control/address bus per memory module in accordance with an embodiment of the invention. The memory system comprises memory controller component 3302, memory module 3303, memory module 3330, address/control clock 3304, address/control clock 3353, termination component 3352, and termination component 3356.

Within each memory module, memory components are organized in ranks. A first rank of memory module 3303 includes memory components 3316, 3317, and 3318. A second rank of memory module 3303 includes memory components 3344, 3345, and 3346. A first rank of memory module 3330 includes memory components 3331, 3332, and 3333. A second rank of memory module 3330 includes memory components 3334, 3335, and 3336.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice 3313, slice 3314, and slice 3315. Each slice comprises one memory component of each rank. In this embodiment, each slice within each memory module is provided with its own address bus 3307 and address/control clock conductor 3310. Address bus 3307 is coupled to memory controller component 3302 and memory components 3316 and 3344. A termination component 3352 is coupled to address bus 3307 near memory controller component 3302, and may, for example, be incorporated into memory controller component 3302. A termination component 3319 is coupled near an opposite terminus of address bus 3307, and is preferably provided within memory module 3303. Address/control clock 3304 is coupled to address/control clock conductor 3309, which is coupled to memory controller component 3302 and to memory components 3316 and 3344. A termination component 3322 is coupled near a terminus of address/control clock conductor 3309, preferably within memory module 3303.

Memory module 3330 is provided with address bus 3354 and address/control clock conductor 3355. Address bus 3354 is coupled to memory controller component 3302 and to memory components, 3331 and 3334. A termination component 3356 is coupled to address bus 3354 near memory controller component 3302, and may, for example, be incorporated into memory controller component 3302. A termination component 3357 is coupled near an opposite terminus of address bus 3354 and is preferably provided within memory module 3330. Address/control clock 3353 is coupled to address/control clock conductor 3355, which is coupled to memory controller component 3302 and to memory components 3331 and 3334. A termination component 3358 is coupled near a terminus of address/control clock conductor 3355, preferably within memory module 3330. Each control/address wire is routed onto a memory

module, past all the memory components in the slice, and into a termination component.

FIG. **34** is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared by all the memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component **3402**, memory module **3403**, memory module **3430**, address/control clock **3404**, splitting component **3459**, splitting component **3460**, splitting component **3461**, splitting component **3462**, termination component **3463**, and termination component **3464**.

Within each memory module, memory components are organized in ranks. A first rank of memory module **3403** includes memory components **3416**, **3417**, and **3418**. A second rank of memory module **3403** includes memory components **3444**, **3445**, and **3446**. A first rank of memory module **3130** includes memory components **3431**, **3432**, and **3433**. A second rank of memory module **3430** includes memory components **3434**, **3435**, and **3436**.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice **3413**, slice **3414**, and slice **3415**. Each slice comprises one memory component of each rank. In this embodiment, an address bus **3407** and an address/control clock conductor **3409** are coupled to each memory component in a slice among multiple memory modules. Address bus **3407** is coupled to memory controller component **3402**, via splitter **3459** to memory components **3416** and **3444**, and via splitter **3461** to memory components **3431** and **3434**. A termination component **3452** is coupled to address bus **3407** near memory controller component **3402**, and may, for example, be incorporated into memory controller component **3402**. A termination component **3463** is coupled near an opposite terminus of address bus **3407**, near splitter **3461**. A termination component **3419** is coupled to address bus **3407**, preferably within memory module **3403**. A termination component **3457** is coupled to address bus **3407**, preferably within memory module **3430**.

Address/control clock **3404** is coupled to address/control clock conductor **3409**, which is coupled to memory controller component **3402**, via splitter **3460** to memory components **3416** and **3444**, and via splitter **3462** to memory components **3431** and **3434**. A termination component **3464** is coupled near a terminus of address/control clock conductor **3409**, near splitter **3462**. A termination component **3422** is coupled to the address/control clock conductor **3409**, preferably within memory module **3403**. A termination component **3458** is coupled to the address/control clock conductor **3409**, preferably within memory module **3430**.

In this example, each control/address wire is tapped using some form of splitting component S. This splitter could be a passive impedance matcher (three resistors in a delta- or y-configuration) or some form of active buffer or switch element. In either case, the electrical impedance of each wire is maintained down its length (within manufacturing limits) so that signal integrity is kept high. As in the previous configuration, each split-off control/address bus is routed onto a memory module, past all the memory components, and into a termination component.

FIG. **35** is a block diagram illustrating a memory system that comprises multiple ranks of memory components and multiple memory modules with a single control/address bus that is shared by all the memory modules in accordance with an embodiment of the invention. The memory system comprises memory controller component **3502**, memory module **3503**, memory module **3530**, address/control clock **3504**, termination component **3519**, and termination component **3522**.

Within each memory module, memory components are organized in ranks. A first rank of memory module **3503** includes memory components **3516**, **3517**, and **3518**. A second rank of memory module **2903** includes memory components **3544**, **3545**, and **3546**. A first rank of memory module **3530** includes memory components **3531**, **3532**, and **3533**. A second rank of memory module **3530** includes memory components **3534**, **3535**, and **3536**.

The memory system is organized into slices across the memory controller component and the memory modules. Examples of these slices include slice **3513**, slice **3514**, and slice **3515**. Each slice comprises one memory component of each rank. In this embodiment, each slice across memory modules shares a common daisy-chained address bus **3507** and a common daisy-chained address/control clock conductor **3509**. Address bus **3507** is coupled to memory controller component **3502** and memory components **3516**, **3544**, **3531**, and **3534**. A termination component **3552** is coupled to address bus **3507** near memory controller component **3502**, and may, for example, be incorporated into memory controller component **3502**. A termination component **3519** is coupled near an opposite terminus of address bus **3507**.

Address/control clock **3504** is coupled to address/control clock conductor **3509**, which is coupled to memory controller component **3502** and to memory components **3516**, **3544**, **3531**, and **3534**. A termination component **3522** is coupled near a terminus of address/control clock conductor **3509**.

Unlike the memory system of FIG. **34**, instead of using some kind of splitting component, each control/address wire is routed onto a memory module, past all the memory components, and back off the module and onto the main board to chain through another memory module or to pass into a termination component.

As can be seen with reference to the Figures described above, embodiments of the invention allow implementation of a memory system, a memory component, and/or a memory controller component. Within these embodiments skew may be measured according to bit time and/or according to a timing signal. In some embodiments, logic in the memory controller component accommodates skew, while in other embodiments, logic in a memory component accommodates skew. The skew may be greater than a bit time or greater than a cycle time.

One embodiment of the invention provides a memory module with a first wire carrying a first signal. The first wire is connected to a first module contact pin. The first wire is connected to a first pin of a first memory component. The first wire is connected to a first termination device. The first wire maintains an approximately constant first impedance value along its full length on the memory module. The termination component approximately matches this first impedance value. Optionally, there is a second memory component to which the first wire does not connect. Optionally, the first signal carries principally information selected from control information, address information, and data information during normal operation. Optionally, the termination device is a component separate from the first memory component on the memory module. Optionally, the termination device is integrated into first memory component on the memory module. Such a memory module may be connected to a memory controller component and may be used in a memory system.

One embodiment of the invention provides a memory module with a first wire carrying a first signal and a second

wire carrying a second signal. The first wire connects to a first module contact pin. The second wire connects to a second module contact pin. The first wire connects to a first pin of a first memory component. The second wire connects to a second pin of the first memory component. The first wire connects to a third pin of a second memory component. The second wire does not connect to a pin of the second memory component. The first wire connects to a first termination device. The second wire connects to a second termination device. The first wire maintains an approximately constant first impedance value along its full length on the memory module. The second wire maintains an approximately constant second impedance value along its full length on the memory module. The first termination component approximately matches the first impedance value. The second termination component approximately matches the second impedance value. Optionally, the first and/or second termination device is a component separate from the first memory component on the memory module. Optionally, the first and/or second termination device is a integrated into the first memory component on the memory module. Optionally, the first signal carries address information and the second signal carries data information. Such a memory module may be connected to a memory controller component and may be used in a memory system.

One embodiment of the invention provides a method for conducting memory operations in a memory system. The memory system includes a memory controller component and a rank of memory components. The memory components include slices. The slices include a first slice and a second slice. The memory controller component is coupled to conductors, including a common address bus connecting the memory controller component to the first slice and the second slice, a first data bus connecting the memory controller component to the first slice, and a second data bus connecting the memory controller component to the second slice. The first data bus is separate from the second data bus The method includes the step of providing a signal to one of the conductors. The signal may be an address signal, a write data signal, or a read data signal. The propagation delay of the one of the conductors is longer than an amount of time that an element of information represented by the signal is applied to that conductor. Optionally, the method may include the step of providing a first data signal to the first data bus and a second data signal to the second data bus. The first data signal relates specifically to the first slice and the second data signal relates specifically to the second slice. In one example, the first data signal carries data to or from the first slice, while the second data signal carries data to or from the second slice.

One embodiment of the invention provides a method for coordinating memory operations among a first memory component and a second memory component. The method includes the step of applying a first address signal relating to the first memory component to a common address bus over a first time interval. The common address bus is coupled to the first memory component and the second memory component. The method also includes the step of applying a second address signal relating to the second memory component to the common address bus over a second time interval. The first time interval is shorter than a propagation delay of the common address bus, and the second time interval is shorter than a common address bus propagation delay of the common address bus. The method also includes the step of controlling a first memory operation of the first memory component using a first memory component timing signal. The first memory component timing signal is dependent upon a first relationship between the common address bus propagation delay and a first data bus propagation delay of a first data bus coupled to the first memory component. The method also includes the step of controlling a second memory operation of the second memory component using a second memory component timing signal. The second memory component timing signal is dependent upon a second relationship between the common address bus propagation delay and a second data bus propagation delay of a second data bus coupled to the second memory component.

One embodiment of the invention (referred to as description B) provides a memory system with a memory controller component, a single rank of memory components on a single memory module, a common address bus connecting controller to all memory components of the rank in succession, separate data buses connecting controller to each memory component (slice) of the rank, an address bus carrying control and address signals from controller past each memory component in succession, data buses carrying read data signals from each memory component (slice) of the rank to the controller, data buses carrying write data signals from controller to each memory component (slice) of the rank, data buses carrying write mask signals from controller to each memory component (slice) of the rank, the read data and write data signals of each slice sharing the same data bus wires (bidirectional), the buses designed so that successive pieces of information transmitted on a wire do not interfere, a periodic clock signal accompanying the control and address signals and used by the controller to transmit information and by the memory components to receive information, a periodic clock signal accompanying each slice of write data signals and optional write mask signals and which is used by the controller to transmit information and by a memory component to receive information, and a periodic clock signal accompanying each slice of read data signals and which is used by a memory component to transmit information and by the controller to receive information.

One embodiment of the invention (referred to as description A) provides a memory system with features taken from the above description (description B) and also a timing signal associated with control and address signals which duplicates the propagation delay of these signals and which is used by the controller to transmit information and by the memory components to receive information, a timing signal associated with each slice of write data signals and optional write mask signals which duplicates the propagation delay of these signals and which is used by the controller to transmit information and by a memory component to receive information, a timing signal associated with each slice of read data signals which duplicates the propagation delay of these signals and which is used by a memory component to transmit information and by the controller to receive information, wherein a propagation delay of a wire carrying control and address signals from the controller to the last memory component is longer than the length of time that a piece of information is transmitted on the wire by the controller.

One embodiment of the invention provides a memory system with features taken from the above description (description A) wherein a propagation delay of a wire carrying write data signals and optional write mask signals from the controller to a memory component is longer than the length of time that a piece of information is transmitted on the wire by the controller.

One embodiment of the invention provides a memory system with features taken from the above description

(description A) wherein a propagation delay of a wire carrying read data signals from a memory component to the controller is longer than the length of time that a piece of information is transmitted on the wire by the memory component.

One embodiment of the invention provides a memory system with features taken from the above description (description A) wherein the alignments of the timing signals of the write data transmitter slices of the controller are adjusted to be approximately the same regardless of the number of slices in the rank, wherein the alignments of timing signals of the read data receiver slices of the controller are adjusted to be approximately the same regardless of the number of slices in the rank, and the alignments of timing signals of the read data receiver slices of the controller are adjusted to be approximately the same as the timing signals of the write data transmitter slices.

One embodiment of the invention provides a memory system with features taken from the above description (description A) wherein the alignments of the timing signals of the write data transmitter slices of the controller are adjusted to be mostly different from one another.

One embodiment of the invention provides a memory system with features taken from the above description (description A) wherein the alignments of timing signals of the read data receiver slices of the controller are adjusted to be mostly different from one another.

One embodiment of the invention provides a memory system with features taken from the above description (description A) wherein the alignments of the timing signals of the read data transmitter of each memory component is adjusted to be the approximately the same as the timing signals of the write data receiver in the same memory component and wherein the alignments of the timing signals will be different for each memory component slice in the rank.

One embodiment of the invention provides a memory system with features taken from the above description (description A) wherein the alignments of the timing signals of the write data transmitter of each memory component is adjusted to be different from the timing signals of the read data receiver in the same memory component.

Numerous variations to the embodiments described herein are possible without deviating from the scope of the claims set forth herein. Examples of these variations are described below. These examples may be applied to control and address signals, read data signals, write data signals, and optional write mask signals. For example, a timing signal associated with the such signals may be generated by an external clock component or by a controller component. That timing signal may travel on wires that have essentially the same topology as the wires carrying such signals. That timing signal may be generated from the information contained on the wires carrying such signals or from a timing signal associated with any of such signals. That timing signal may be asserted an integral number of times during the interval that each piece of information is present on a wire carrying such signals. As another variation, an integral number of pieces of information may be asserted on a wire carrying such signals each time the timing signal associated with such signals is asserted. As yet another variation, an integral number of pieces of information may be asserted on a wire carrying such signals each time the timing signal associated with such signals is asserted an integral number of times. The point when a timing signal associated with such signals is asserted may have an offset relative to the time interval that each piece of information is present on a wire carrying such signals.

As examples of other variations, the termination components for some of the signals may be on any of a main printed wiring board, a memory module board, a memory component, or a controller component. Also, two or more ranks of memory components may be present on the memory module and with some control and address signals connecting to all memory components and with some control and address signals connecting to some of the memory components. It is also possible for two or more modules of memory components to be present in the memory system, with some control and address signals connecting to all memory components and with some control and address signals connecting to some of the memory components.

Various aspects of the subject-matter described herein are set out non-exhaustively in the following numbered clauses:

1. A memory system comprising:
   a memory controller component;
   a rank of memory components comprising slices; and
   conductors coupling the memory controller component to the rank of memory components and coupling the memory controller component to the slices of the rank of memory components, wherein a propagation delay of one of the conductors carrying a signal selected from a group consisting of an address signal, a write data signal, and a read data signal is longer than an amount of time that an element of information represented by the signal is applied to the conductor, wherein the conductors comprise:
      a common address bus connecting the memory controller component to each of the slices of the rank in succession; and
      separate data buses connecting the memory controller component to each of the slices of the rank.
2. The memory system of clause 1 wherein the common address bus is coupled to a plurality of the slices.
3. The memory system of clause 2 wherein the separate data buses comprise:
   a first data bus connecting the memory controller component to a first slice of the slices; and
   a second data bus connecting the memory controller component to a second slice of the slices, wherein the first data bus and the second data bus carry different signals independently of each other.
4. A memory system comprising:
   a memory controller component;
   a rank of memory components comprising slices; and
   conductors coupling the memory controller component to the slices of the rank of memory components, wherein the conductors comprise a first data bus coupled to the memory controller component and the first slice and a second data bus coupled to the memory controller component and the second slice, the first data bus being separate from the second data bus, wherein first elements of information relating to the first slice are driven on a first conductor of the conductors coupled to the first slice for a first element time interval from a first time to a second time, wherein second elements of information relating to the second slice are driven on a second conductor of the conductors coupled to the second slice for a second element time interval from a third time to a fourth time, and wherein the memory controller component comprises a logic circuit adapted

to accommodate a difference between the first time and the third time that is greater than a first duration of the first element time interval.

5. A memory system comprising:
a memory controller component;
a rank of memory components comprising slices, the slices comprising a first slice and a second slice;
conductors coupling the memory controller component to the slices of the rank of memory components, wherein the conductors comprise a first data bus coupled to the memory controller component and the first slice and a second data bus coupled to the memory controller component and the second slice, the first data bus being separate from the second data bus, wherein first elements of information relating to the first slice are driven on a first conductor of the conductors coupled to the first slice for a first element time interval from a first time to a second time, wherein second elements of information relating to the second slice are driven on a second conductor of the conductors coupled to the second slice for a second element time interval from a third time to a fourth time; and
a logic circuit adapted to accommodate a difference between the first time and the third time that is greater than a cycle time of a clock circuit of the memory controller component.

6. The memory system of clause 5 wherein the logic circuit is incorporated into the memory controller component.

7. The memory system of clause 5 wherein the logic circuit is incorporated into at least one of the memory components.

8. A memory system comprising:
a memory controller component;
a rank of memory components comprising slices, the slices comprising a first slice and a second slice;
a common address bus connecting the memory controller component to the first slice and the second slice in succession;
a first data bus connecting the memory controller component to the first slice; and
a second data bus connecting the memory controller component to the second slice, the first data bus being separate from the second data bus, wherein first elements of information are driven on the first data bus for a first element time interval from a first time to a second time, wherein second elements of information are driven on the second data bus for a second element time interval from a third time to a fourth time, wherein third elements of information are driven on the common address bus for a third element time interval from a fifth time to a sixth time, wherein there is a first access time interval between the fifth time and the first time, wherein fourth elements of information are driven on the common address bus for a fourth element time interval from a seventh time to an eighth time, wherein there is a second access time interval between the seventh time and the third time, and wherein at least one of the memory components of the rank of memory components comprises a logic circuit adapted to accommodate a difference between the first access time interval and the second access time interval that is greater than a first duration of the first element time interval.

9. A memory system comprising:
a memory controller component;
a rank of memory components comprising slices, the slices comprising a first slice and a second slice;
conductors coupling the memory controller component to the slices of the rank of memory components, wherein the conductors comprise a first data bus coupled to the memory controller component and the first slice and a second data bus coupled to the memory controller component and the second slice, the first data bus being separate from the second data bus, wherein first elements of information relating to the first slice are driven on a first conductor of the conductors coupled to the first slice for a first element time interval, the first element time interval associated with a first timing signal event, wherein second elements of information relating to the second slice are driven on a second conductor of the conductors coupled to the second slice for a second element time interval, the second element time interval associated with a second timing signal event; and
a logic circuit adapted to accommodate a difference between the first timing signal event and the second timing signal event that is greater than a duration selected from a group consisting of the first element time interval and a cycle time of a clock circuit of the memory controller component.

10. The memory system of clause 9 wherein the logic circuit is incorporated into the memory controller component.

11. The memory system of clause 9 wherein the logic circuit is incorporated into at least one of the memory components.

12. A memory system comprising:
a memory controller component;
a rank of memory components comprising slices, the slices comprising a first slice and a second slice;
a common address bus connecting the memory controller component to the first slice and the second slice in succession;
a first data bus connecting the memory controller component to the first slice; and
a second data bus connecting the memory controller component to the second slice, wherein first elements of information are driven on the first data bus for a first element time interval, the first element time interval associated with a first timing signal event, wherein second elements of information are driven on the second data bus for a second element time interval, the second element time interval associated with a second timing signal event, wherein third elements of information are driven on the common address bus for a third element time interval, the third element time interval associated with a third timing signal event, wherein there is a first access time interval between the third timing signal event and the first timing signal event, wherein fourth elements of information are driven on the common address bus for a fourth element time interval, the fourth element time interval associated with a fourth timing signal event, wherein there is a second access time interval between the fourth timing signal event and the second timing signal event, and wherein at least one of the memory components of the rank of memory components comprises a logic circuit adapted to accommodate a difference between the first access time interval and the second access time interval that is greater than a first duration of the first element time interval.

13. A memory component adapted to be coupled to a memory controller component as a first slice of a rank of memory components, the rank further comprising a second slice, wherein conductors couple the memory con-

troller component to the first slice and the second slice, the conductors comprising a first data bus coupling the first slice to the memory controller component and a second data bus coupling the second slice to the memory controller component, the first data bus being separate from the second data bus, the memory component comprising:

a logic circuit adapted to cause the memory controller component to accommodate a difference between a first time and a third time that is greater than a first duration of a first element time interval, wherein first elements of information are driven on a first conductor of the conductors coupled to the first slice for the first element time interval from the first time to a second time, wherein second elements of information relating to the second slice are driven on a second conductor of the conductors coupled to the second slice for a second element time interval from the third time to a fourth time.

14. A memory component adapted to be coupled to a memory controller component as a first slice of a rank of memory components, the rank further comprising a second slice, a common address bus connecting the memory controller component to the first slice and the second slice in succession, a first data bus connecting the memory controller component to the first slice, a second data bus connecting the memory controller component to the second slice, the first data bus being separate from the second data bus, the memory component comprising:

a logic circuit adapted to accommodate a difference between a first access time interval and a second access time interval that is greater than a first duration of a first element time interval, wherein first elements of information are driven on the first data bus for the first element time interval from a first time to a second time, wherein second elements of information are driven on the second data bus for a second element time interval from a third time to a fourth time, wherein third elements of information are driven on the common address bus for a third element time interval from a fifth time to a sixth time, wherein the first access time interval occurs between the fifth time and the first time, wherein fourth elements of information are driven on the common address bus for a fourth element time interval from a seventh time to an eighth time, wherein the second access time interval occurs between the seventh time and the third time.

15. A memory component adapted to be coupled to a memory controller component as a first slice of a rank of memory components, the rank further comprising a second slice, wherein conductors couple the memory controller component to the slices of the rank of memory components, the conductors comprising a first data bus coupled to the memory controller component and the first slice and a second data bus coupled to the memory controller component and the second slice, the first data bus being separate from the second data bus, the memory component comprising:

a logic circuit adapted to cause the memory controller component to accommodate a difference between a first timing signal event and a second timing signal event that is greater than a first duration of a first element time interval, wherein first elements of information are driven on a first conductor of the conductors coupled to the first slice for the first element time interval, the first element time interval associated with the first timing signal event, wherein second elements of information relating to the second slice are driven on a second conductor of the conductors coupled to the second slice for a second element time interval, the second element time interval associated with the second timing signal event.

16. A memory component adapted to be coupled to a memory controller component as a first slice of a rank of memory components, the rank of memory components further comprising a second slice, a common address bus connecting the memory controller component to the first slice and the second slice in succession, a first data bus connecting the memory controller component to the first slice, a second data bus connecting the memory controller component to the second slice, the memory component comprising:

a logic circuit adapted to accommodate a difference between a first access time interval and a second access time interval that is greater than a first duration of a first element time interval, wherein first elements of information are driven on the first data bus for the first element time interval, the first element time interval associated with a first timing signal event, wherein second elements of information are driven on the second data bus for a second element time interval, the second element time interval associated with a second timing signal event, wherein third elements of information are driven on the common address bus for a third element time interval, the third element time interval associated with a third timing signal event, wherein the first access time interval occurs between the third timing signal event and the first timing signal event, wherein fourth elements of information are driven on the common address bus for a fourth element time interval, the fourth element time interval associated with a fourth timing signal event, wherein the second access time interval occurs between the fourth timing signal event and the second timing signal event.

17. A method for conducting memory operations in a memory system comprising a memory controller component and a rank of memory components comprising slices, the slices comprising a first slice and a second slice, the memory controller component coupled to conductors, the conductors including a common address bus connecting the memory controller component to the first slice and the second slice, a first data bus connecting the memory controller component to the first slice, and a second data bus connecting the memory controller component to the second slice, the first data bus being separate from the second data bus, the method comprising the step of:

providing a signal to one of the conductors, the signal selected from a group consisting of an address signal, a write data signal, and a read data signal, wherein the propagation delay of the one of the conductors is longer than an amount of time that an element of information represented by the signal is applied to the conductor.

18. The method of clause 17 further comprising the step of:

providing a first data signal to the first data bus and a second data signal to the second data bus, the first data signal relating specifically to the first slice and the second data signal relating specifically to the second slice.

19. A method for coordinating memory operations among a first memory component and a second memory component, the method comprising the steps of:

applying a first address signal relating to the first memory component to a common address bus over a first time

interval, the common address bus coupled to the first memory component and the second memory component;

applying a second address signal relating to the second memory component to the common address bus over a second time interval, the first time interval being shorter than a propagation delay of the common address bus and the second time interval being shorter than a common address bus propagation delay of the common address bus; and

controlling a first memory operation of the first memory component using a first memory component timing signal, the first memory component timing signal dependent upon a first relationship between the common address bus propagation delay and a first data bus propagation delay of a first data bus coupled to the first memory component; and

controlling a second memory operation of the second memory component using a second memory component timing signal, the second memory component timing signal dependent upon a second relationship between the common address bus propagation delay and a second data bus propagation delay of a second data bus coupled to the second memory component.

Accordingly, a method and apparatus for coordinating memory operations among diversely-located memory components has been described. It should be understood that the implementation of other variations and modifications of the invention in its various aspects will be apparent to those of ordinary skill in the art, and that the invention is not limited by the specific embodiments described. It is therefore contemplated to cover by the present invention, any and all modifications, variations, or equivalents that fall within the spirit and scope of the basic underlying principles disclosed and claimed herein.

What is claimed is:

1. A system comprising:

a first memory device and a second memory device;

a control signal path coupled to the first memory device and the second memory device such that a read command propagating on the control signal path propagates past the first memory device before reaching the second memory device and such that a first propagation time required for the read command to propagate on the control signal path from the memory controller to the first memory device is different than a second propagation time required for the read command to propagate on the control signal path from the memory controller to the second memory device;

a first signal line coupled to the first memory device to convey first data output from the first memory device in response to the read command;

a second signal line coupled to the second memory device to convey second data output from the second memory device in response to the read command; and

a memory controller including:

a first circuit to receive the first data from the first memory device after delaying for a first time interval that is based, at least in part, on the first propagation time; and

a second circuit to receive the second data from the second memory device after delaying for a second time interval that is based, at least in part, on the second propagation time.

2. The system of claim 1, wherein the first time interval corresponds to a sum of the first propagation time and a time required for the first data to propagate on the first signal line from the first memory device to the memory controller.

3. The system of claim 1, wherein:

the first circuit comprises a first phase offset selection circuit to select a first clock signal from a plurality of phase-distributed clock signals to be a timing reference for reception of the first data on the first signal line; and

the second circuit comprises a second phase offset selection circuit to select a second clock signal from the plurality of phase-distributed clock signals to be a timing reference for reception of the second data on the second signal line.

4. The system of claim 1, further comprising:

a first termination component coupled to the first signal line, wherein the first termination component is disposed on the first memory device;

a second termination component coupled to the second signal line, wherein the second termination component is disposed on the second memory device; and

a termination component coupled to the control signal path.

5. The system of claim 1, further comprising a timing signal path coupled to the first memory device and the second memory device, the timing signal path to carry a timing signal that indicates a first time at which the read command propagating on the control signal path is sampled by the first memory device, and wherein the timing signal indicates a second time at which the read command propagating on the control signal path is sampled by the second memory device.

6. The system of claim 1, further comprising:

a third signal line to carry a first timing signal that indicates a time at which the first data conveyed on the first signal line is sampled by the memory controller; and

a fourth signal line to carry a second timing signal that indicates a time at which the second data conveyed on the second signal line is sampled by the memory controller.

7. A memory controller comprising:

control transmit circuitry to transmit a control signal to a plurality of memory devices via a shared control signal path, the shared control signal path being coupled to each of the memory devices at a different point along its length such that respective times required for the control signal to propagate from the memory controller to the memory devices are different;

data receive circuitry to receive data signals from the memory devices via respective data signal paths; and

timing circuitry to delay reception of data signals on each of the data signal paths by a respective time interval that is based, at least in part, on the time required for the control signal to propagate on the control signal path from the memory controller to a respective memory device of the memory devices.

8. The memory controller of claim 7, wherein the respective time interval corresponds to a sum of the time required for the control signal to propagate on the control signal path from the memory controller to the respective memory device and a time required for data signals to propagate on the data signal path from the respective memory device to the memory controller.

9. The memory controller of claim 7, wherein the timing circuitry comprises a plurality of phase offset selection circuits, each phase offset selection circuit of the plurality of phase offset selection circuits to select a clock signal of a

plurality of phase-distributed clock signals to be a timing reference for receiving data conveyed on a respective one of the data signal paths.

10. The memory controller of **9**, further comprising clock generation circuitry to generate the plurality of phase-distributed clock signals, the plurality of phase-distributed clock signals having phase offsets that are substantially evenly distributed within a cycle time of a first clock signal.

11. The memory controller of claim **10**, wherein the clock generation circuitry comprises a phase-locked loop circuit to generate the first clock signal based on a reference clock signal that has a lower frequency than the first clock signal.

12. The memory controller of claim **9**, wherein each phase offset selection circuit of the plurality of phase offset selection circuits comprises a multiplexer to select the clock signal of the plurality of phase-distributed clock signals in response to a respective control signal.

13. A method of controlling memory devices in a system that includes at least a first memory device and a second memory device, the method comprising:

transmitting a control signal to the memory devices via a control signal path, wherein the control signal propagates past the first memory device and the second memory devices in sequence, and wherein a first propagation time required for the control signal to propagate on the control signal path to the first memory device is different from a second propagation time required for the control signal to propagate on the control signal path to the second memory device;

receiving a first data signal from the first memory device via a first data signal path after delaying for a first time interval, wherein the first time interval is based, at least in part on the first propagation time; and

receiving a second data signal from the second memory device via a second data signal path after delaying for a second time interval, wherein the second time interval is based, at least in part on the second propagation time.

14. The method of claim **13**, wherein receiving the first data signal from the first memory device via the first data signal path after delaying for a first time interval comprises delaying reception of the first data signal according to a sum of the first propagation time and a time required for the first data signal to propagate from the first memory device to the memory controller via the first data signal path.

15. The method of claim **13**, wherein receiving the second data signal from the second memory device via the second data signal path after delaying for the second time interval comprises delaying reception of the second data signal according to a sum of the second propagation time and a time required for the second data signal to propagate from the second memory device to the memory controller via the second data signal path.

16. The method of claim **13**, wherein receiving the first data signal from the first memory device via the first data signal path after delaying for the first time interval comprises selecting a receive clock signal from a plurality of phase-distributed clock signals such that the receive clock signal includes a phase offset that corresponds to the first time interval.

17. The method of claim **16**, further comprising generating the plurality of phase-distributed clock signals in a phase-locked loop circuit.

18. A controller device comprising:

means for transmitting, via a control signal path, a control signal to a first memory device and a second memory device such that the control signal propagates past the first memory device and the second memory device in sequence, and such that a first propagation time required for the control signal to propagate on the control signal path to the first memory device is different from a second propagation time required for the control signal to propagate on the control signal path to the second memory device;

means for receiving a first data signal from the first memory device via a first data signal path after delaying for a first time interval, wherein the first time interval is based, at least in part, on the first propagation time; and

means for receiving a second data signal from the second memory device via a second data signal path after delaying for a second time interval, wherein the second time interval is based, at least in part, on the second propagation time.

19. A memory system comprising:

a memory module having a row of memory devices, including a first memory device and a second memory device, and a first termination structure;

a first data signal path coupled to the first memory device;

a second data signal path coupled to the second memory device;

a timing signal path coupled to each of the memory devices and the first termination structure, the timing signal path extending along the row of memory devices such that a timing signal propagating on the timing signal path propagates past each of the memory devices in order before reaching the first termination structure, and such that a first propagation time required for the timing signal to propagate on the timing signal path to the first memory device is different from a second propagation time required for the timing signal to propagate on the timing signal path to the second memory device; and

a memory controller, coupled to the memory module, the memory controller having first receive circuitry to receive read data on the first data path after a first time delay based, at least in part, on the first propagation time, and second receive circuitry to receive read data on the second data path after a second time delay based, at least in part, on the second propagation time.

20. The memory system of claim **19**, wherein the first time delay is selected to enable the read data on the first data signal path to be transmitted by the first memory device synchronously with respect to arrival of the timing signal at the first memory device, and wherein the second time delay is selected to enable the read data on the second data signal path to be transmitted by the second memory device synchronously with respect to arrival of the timing signal at the second memory device.

21. A memory controller comprising:

a first data reception circuit to receive first read data from a first memory device via a first dedicated data signal path after delaying for a first period of time that is based, at least in part, on a first propagation time required for a timing signal to propagate on a clock line to the first memory device; and

a second data reception circuit to receive second read data from a second memory device via a second dedicated data signal path after delaying for a second period of time that is based, at least in part, on a second propagation time required for the timing signal to propagate on the clock line to the second memory device, the second propagation time being different from the first propagation time.

22. The memory controller of claim 21, wherein the first period of time is selected to enable the first read data to be transmitted by the first memory device synchronously with respect to arrival of the timing signal at the first memory device, and wherein the second period of time is selected to enable the second read data to be transmitted by the second memory device synchronously with respect to arrival of the timing signal at the second memory device.

23. The memory controller of claim 21, wherein the first period of time corresponds to a sum of the first propagation time and a time required for the first read data to propagate, via the first data signal path, from first memory device to the memory controller, and wherein the second period of time corresponds to a sum of the second propagation time and a time required for the second read data to propagate, via the second data signal path, from the second memory device to the memory controller.

24. A system comprising:

a first memory device including a first memory array;

a second memory device including a second memory array;

a first signal line coupled to the first memory device, the first signal line to convey first data retrieved from a first location in the first memory array and output from the first memory device, the first location specified by address information;

a second signal line coupled to the second memory device, the second signal line to convey second data retrieved from a second location in the second memory array and output from the second memory device, the second location specified by the address information;

a third signal line coupled to the first memory device and the second memory device, such that the address information, propagating on the third signal line, propagates past the first memory device before reaching the second memory device, and such that a first propagation time required for the address information to propagate on the third signal line to the first memory device is different than a second propagation time required for the address information to propagate to the second memory device; and

a controller including:

a first circuit to delay reception of the first data output from the first memory device by a first time interval that is based, at least in part, on the first propagation time; and

a second circuit to delay reception of the second data output from the second memory device by a second time interval that is based, at least in part, on the second propagation time.

* * * * *

US006292903B1

(12) United States Patent

Coteus et al.

(10) Patent No.: US 6,292,903 B1
(45) Date of Patent: Sep. 18, 2001

(54) **SMART MEMORY INTERFACE**

(75) Inventors: **Paul William Coteus**, Yorktown Heights, NY (US); **Daniel Mark Dreps**, Georgetown, TX (US); **Frank Ferraiolo**, Essex, VT (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (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.

(21) Appl. No.: **09/106,639**

(22) Filed: **Jun. 29, 1998**

Related U.S. Application Data

(60) Provisional application No. 60/052,044, filed on Jul. 9, 1997.

(51) **Int. Cl.**[7] **G06F 1/04**
(52) **U.S. Cl.** **713/401**; 713/503
(58) **Field of Search** 713/400, 401, 713/500, 503, 600; 711/167, 168, 169

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,097,489 * 3/1992 Tucci 375/374
5,577,236 * 11/1996 Johnson et al. 713/400
5,579,352 * 11/1996 Llewellyn 375/376
5,615,358 * 3/1997 Vogley 713/501
6,131,149 * 10/2000 Lu et al. 711/167

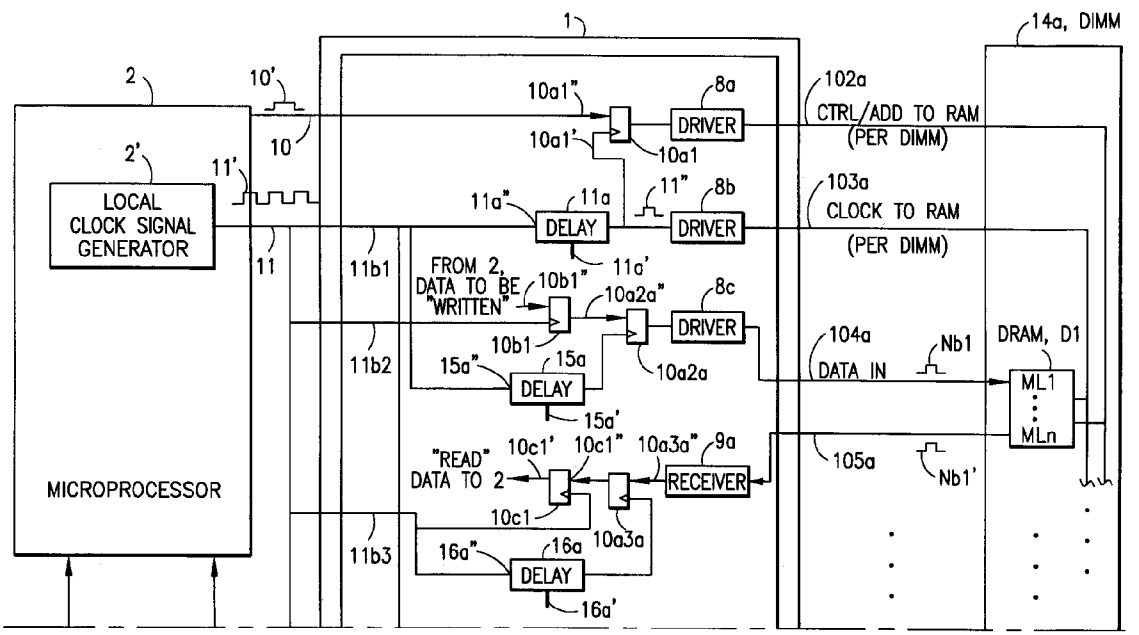
* cited by examiner

Primary Examiner—Dennis M. Butler
(74) *Attorney, Agent, or Firm*—Daniel P. Morris, Esq.; Perman & Green, LLP

(57) **ABSTRACT**

A method and apparatus are disclosed for initiating a start-up operation of a system (**1'**) having a master device (**1**) and a slave device (**14*a*–14*n***). The method comprises steps of: A) exercising the slave device (**14*a*–14*n***) using the master device (**1**) to determine a temporal range within which temporal relationships of electrical signals need to be set in order to operate the system (**1'**) without error; B) setting the temporal relationships of the electrical signals so as to be within the determined temporal range; and C) storing a record of the determined temporal range, for subsequent use in operating the system (**1'**). In one embodiment of the invention, the system (**1'**) includes a memory control system of a computer system (**1"**), and the slave device (**14*a*–14*n***) includes memory devices of the computer system (**1"**). The method of the invention substantially compensates for any differences in times of arrival for data being transferred from the master device (**1**) to the slave device (**14*a*–14*n***), and vice versa, and thus minimizes the possibility of read/write errors being encountered, while increasing the overall processing speed and efficiency of the system (**1'**).

16 Claims, 16 Drawing Sheets



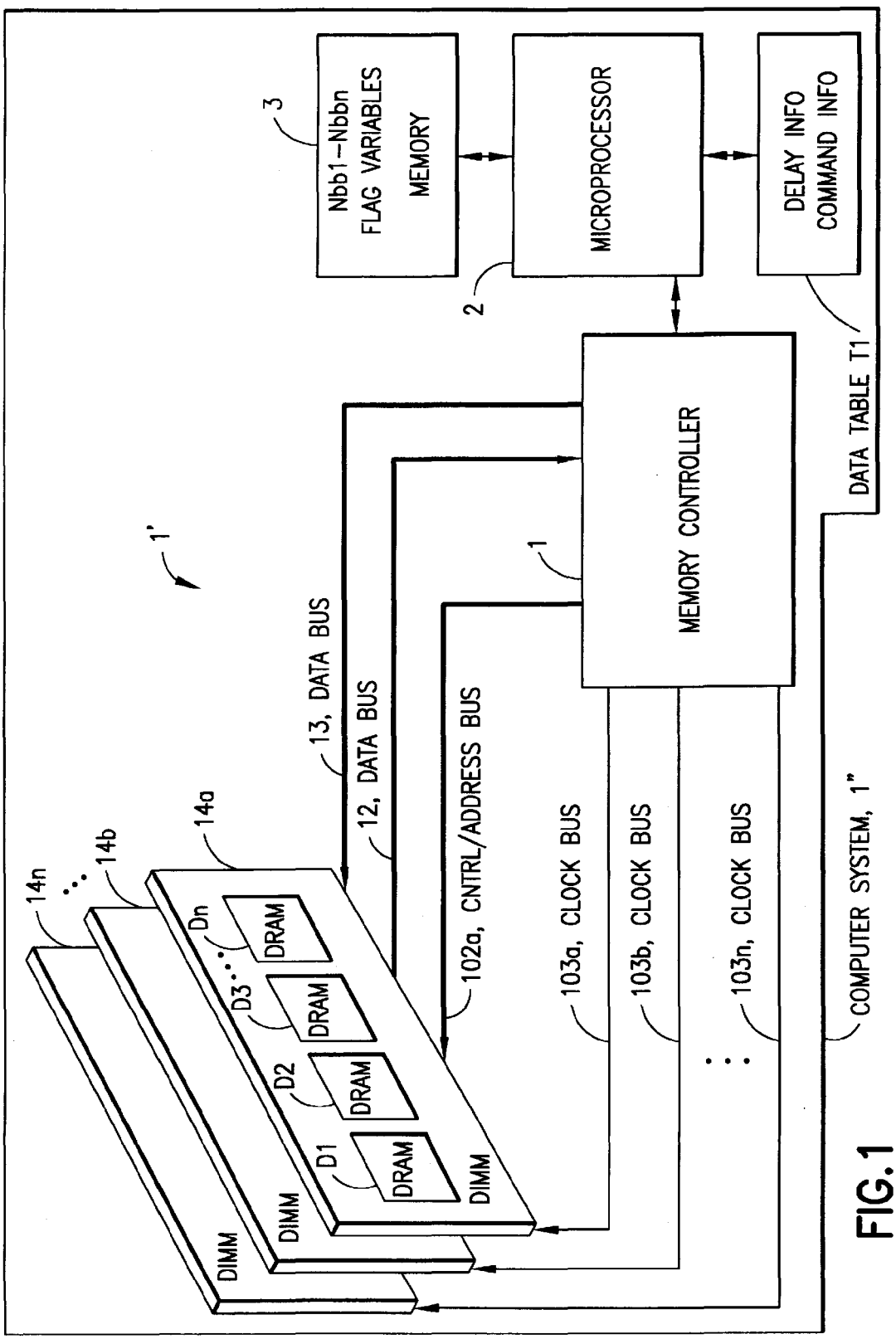


FIG. 1

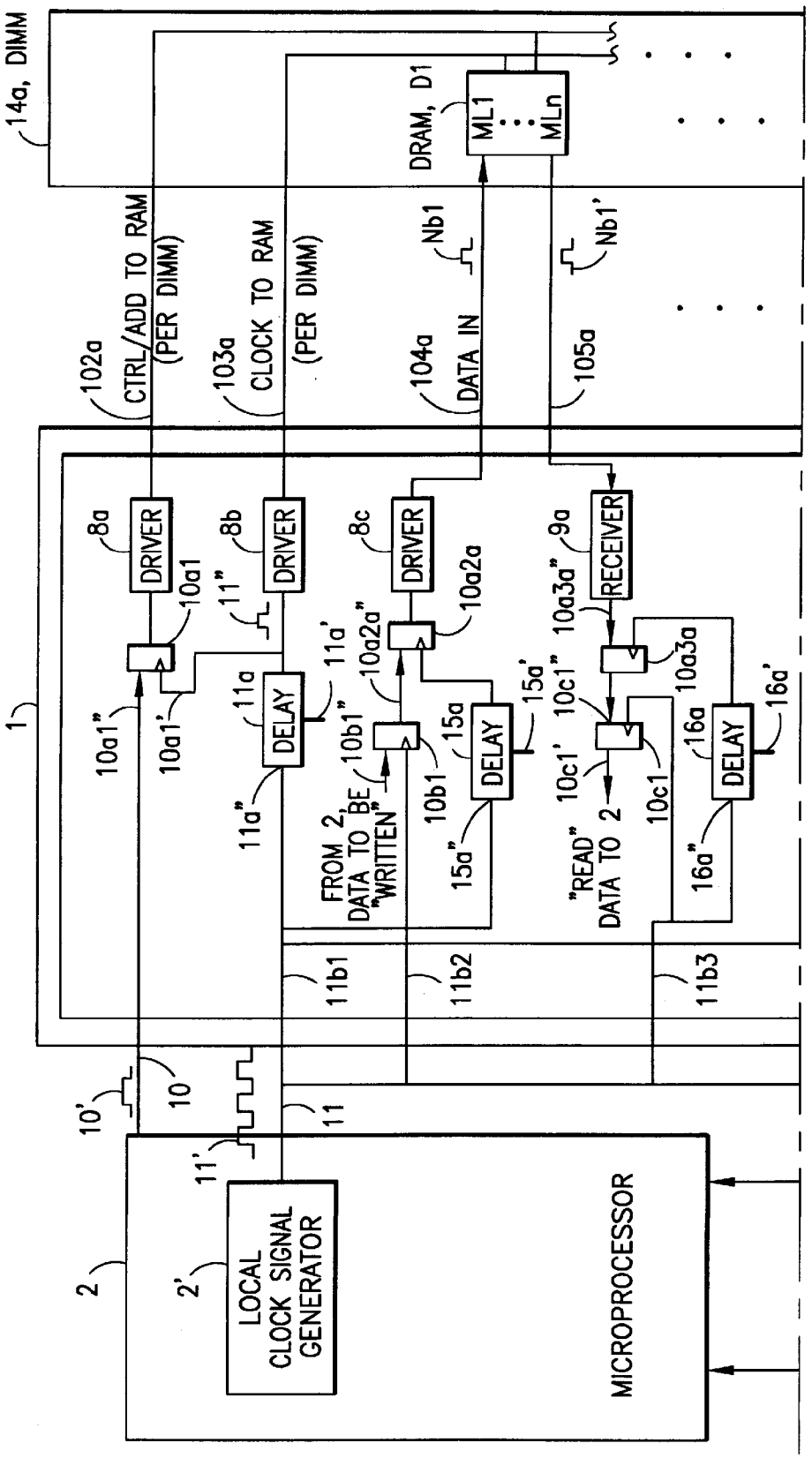


FIG.2a

FIG.2b

FIG.2

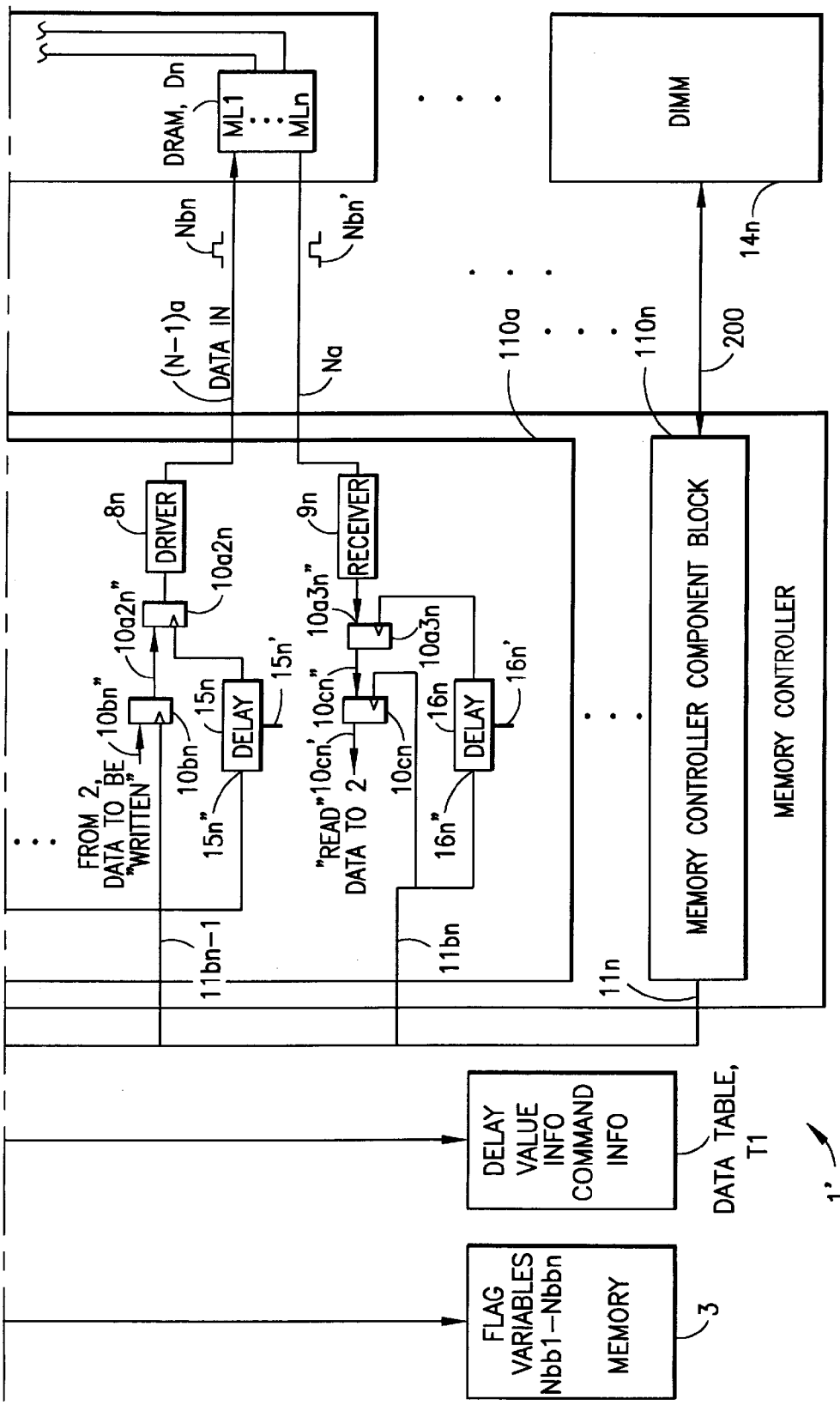


FIG.2b

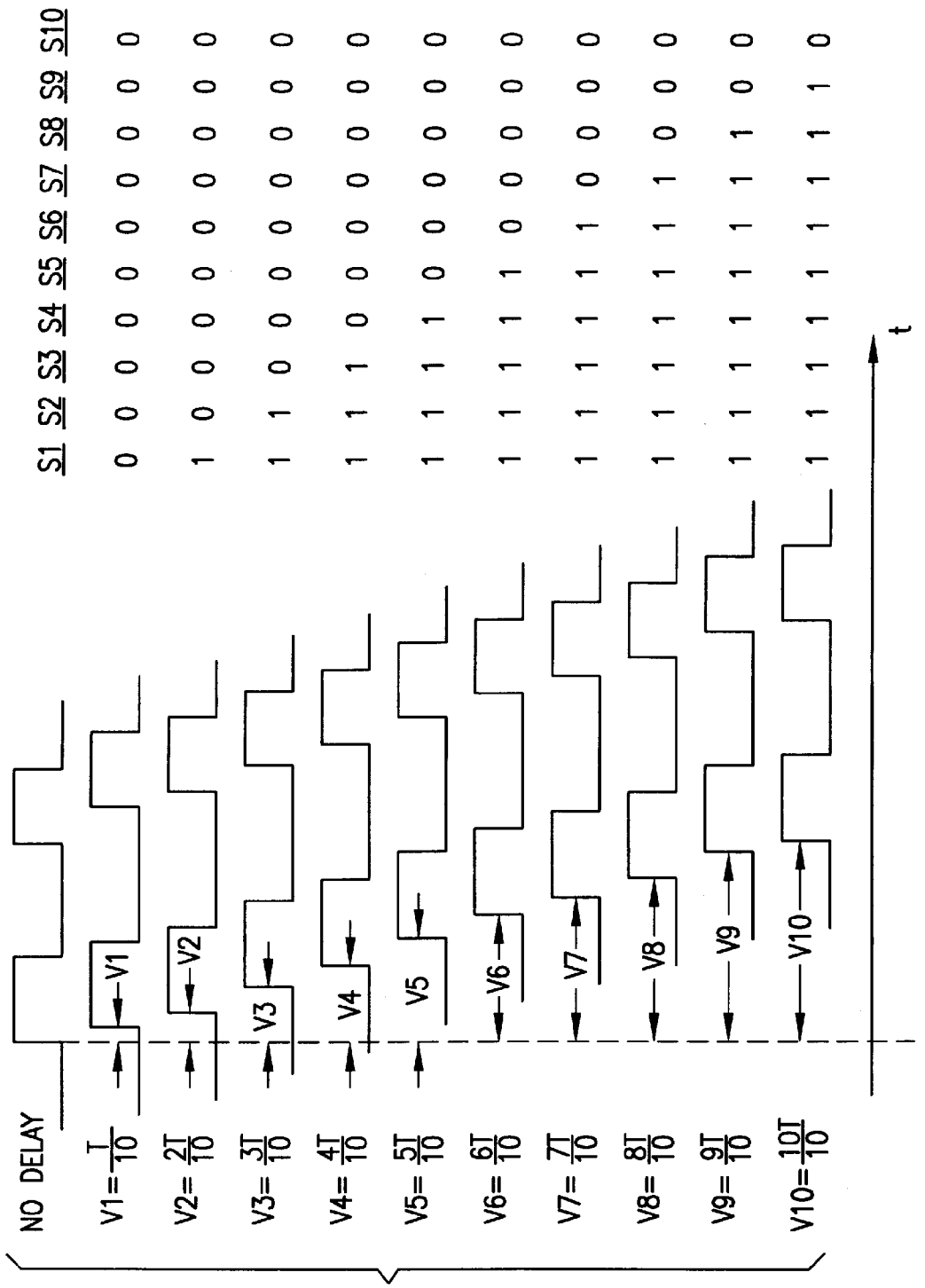


FIG.2c

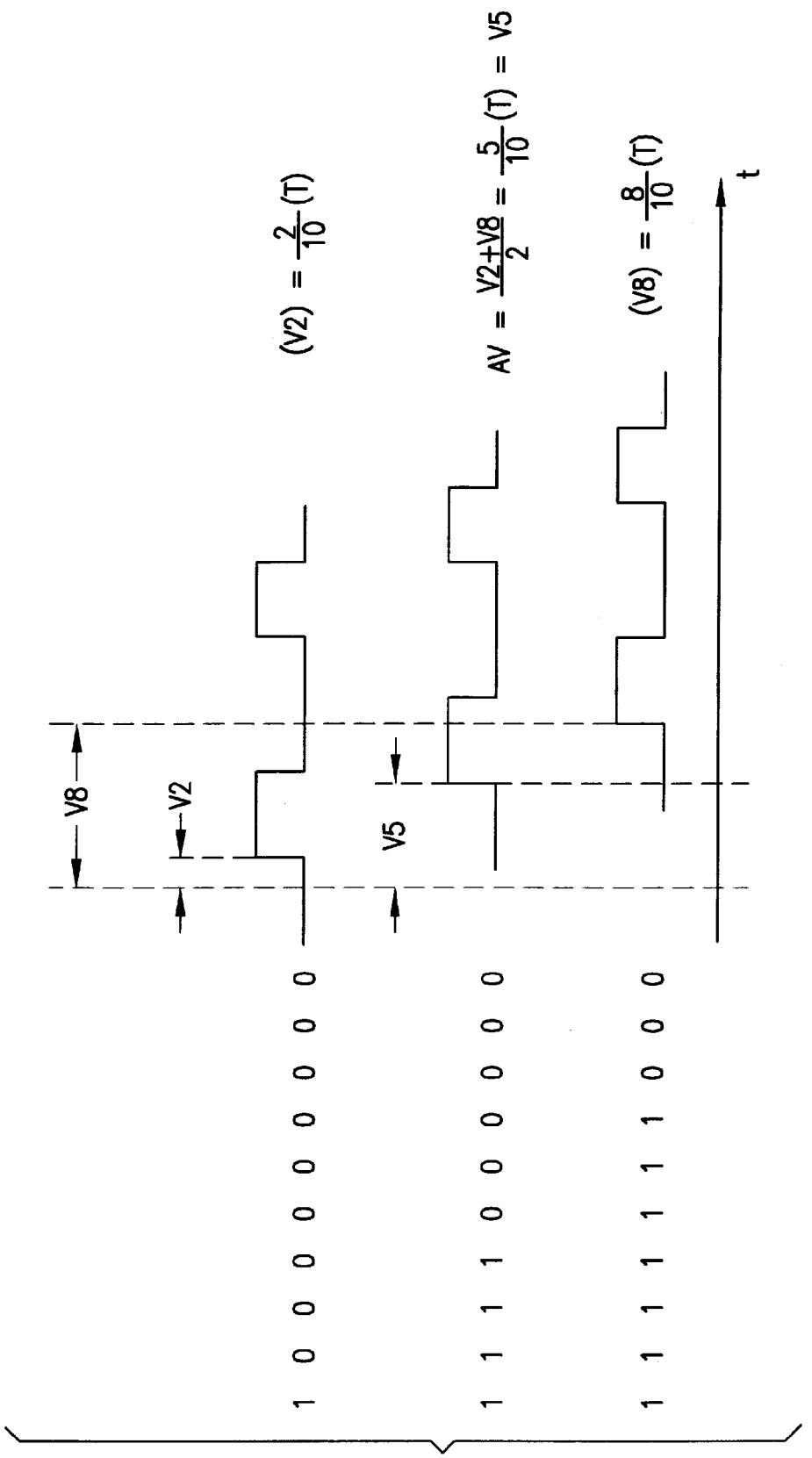


FIG.2d

A40625

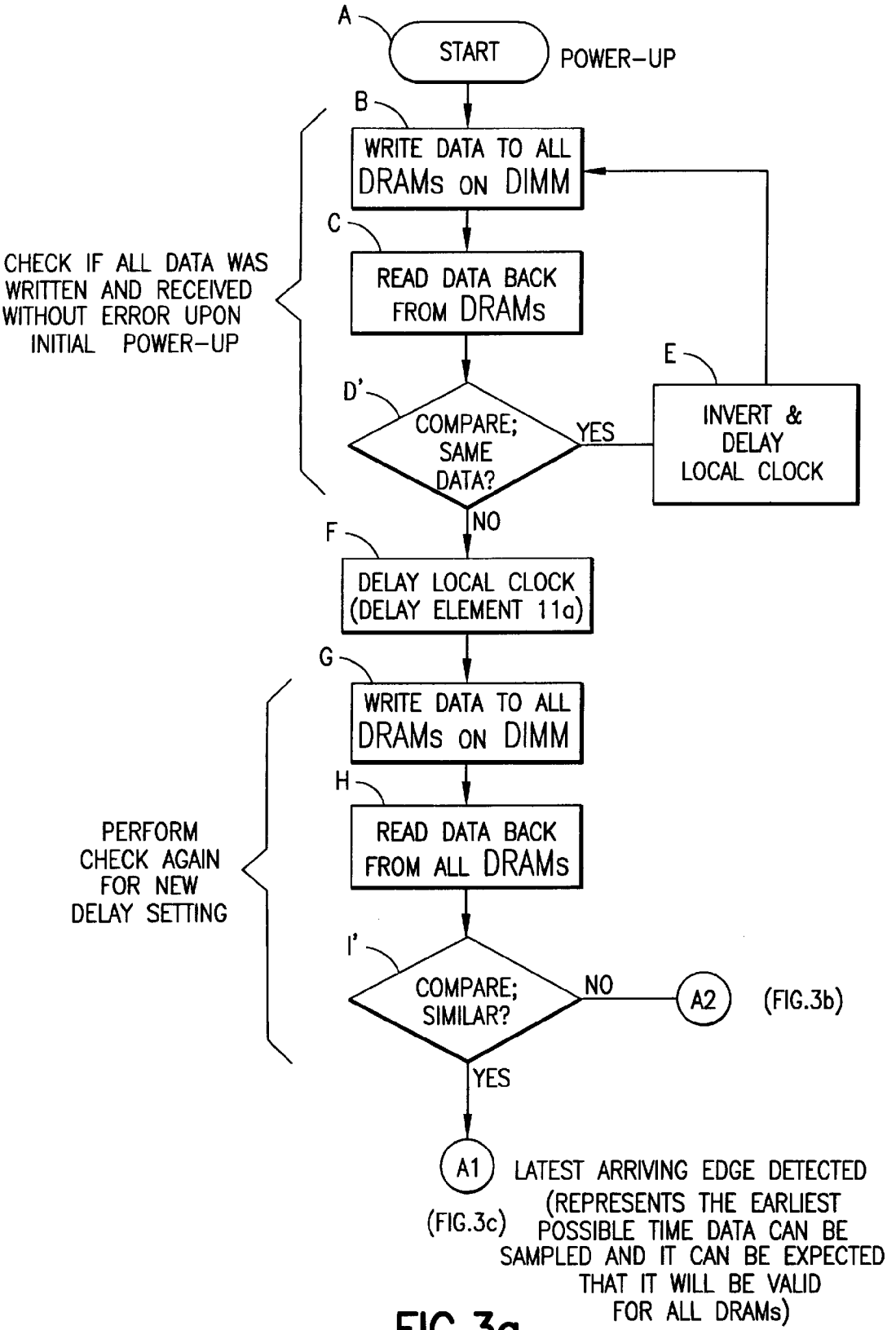


FIG.3a

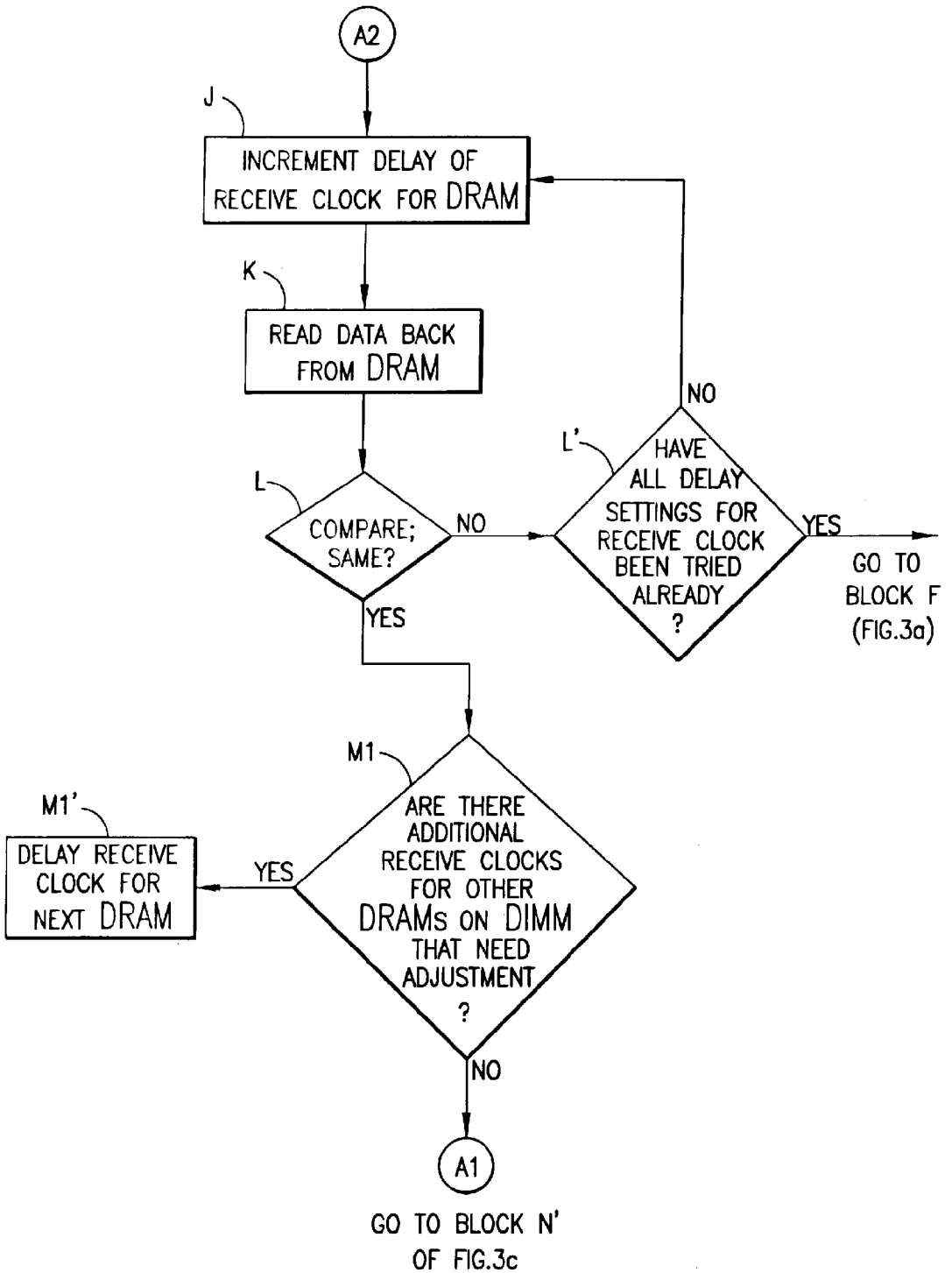

A2
J
INCREMENT DELAY OF RECEIVE CLOCK FOR DRAM
K
READ DATA BACK FROM DRAM
L
COMPARE; SAME?
NO
L'
HAVE ALL DELAY SETTINGS FOR RECEIVE CLOCK BEEN TRIED ALREADY ?
NO
YES
GO TO BLOCK F (FIG.3a)
YES
M1
ARE THERE ADDITIONAL RECEIVE CLOCKS FOR OTHER DRAMs ON DIMM THAT NEED ADJUSTMENT ?
M1'
DELAY RECEIVE CLOCK FOR NEXT DRAM
YES
NO
A1
GO TO BLOCK N' OF FIG.3c

FIG.3b

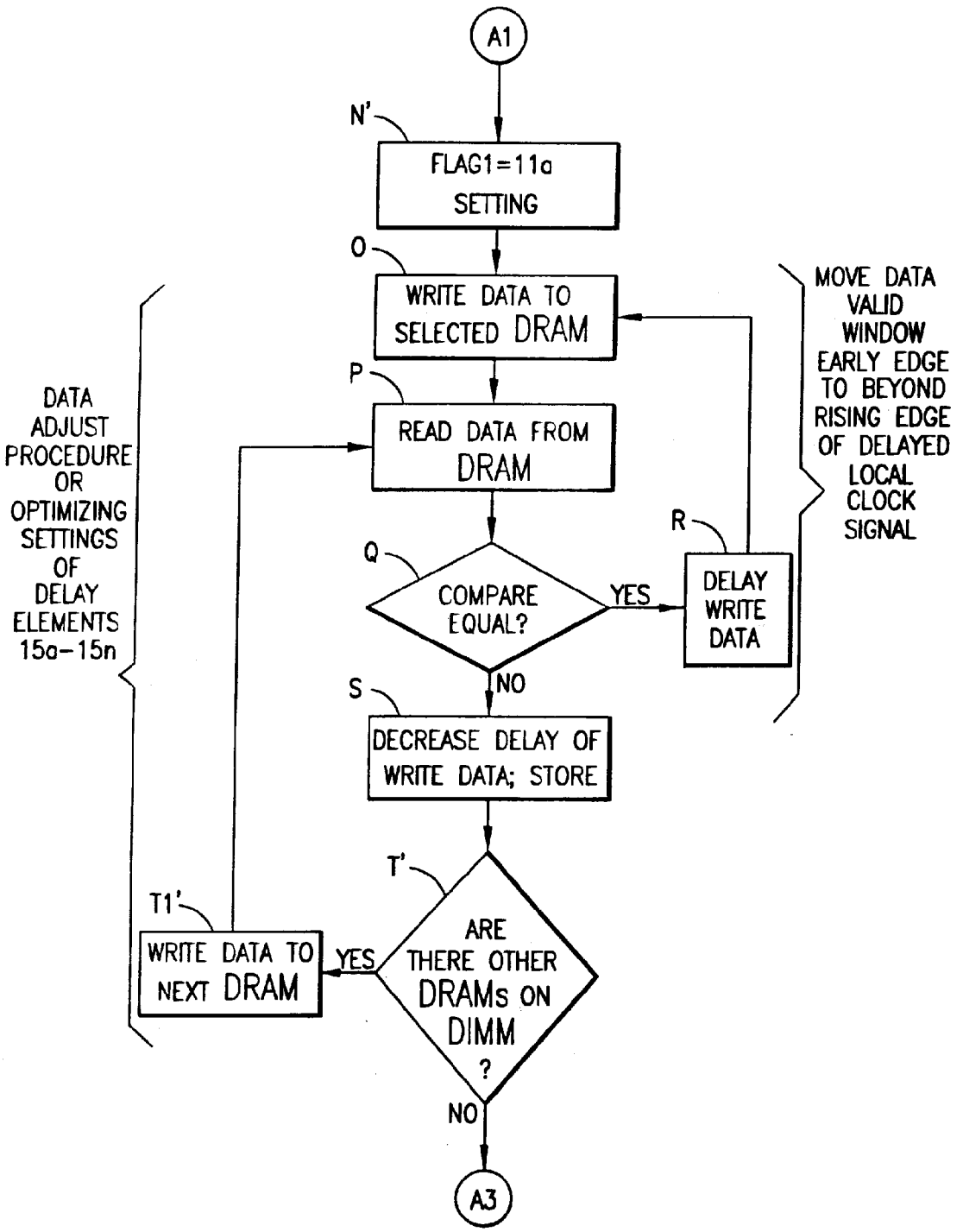


FIG.3c

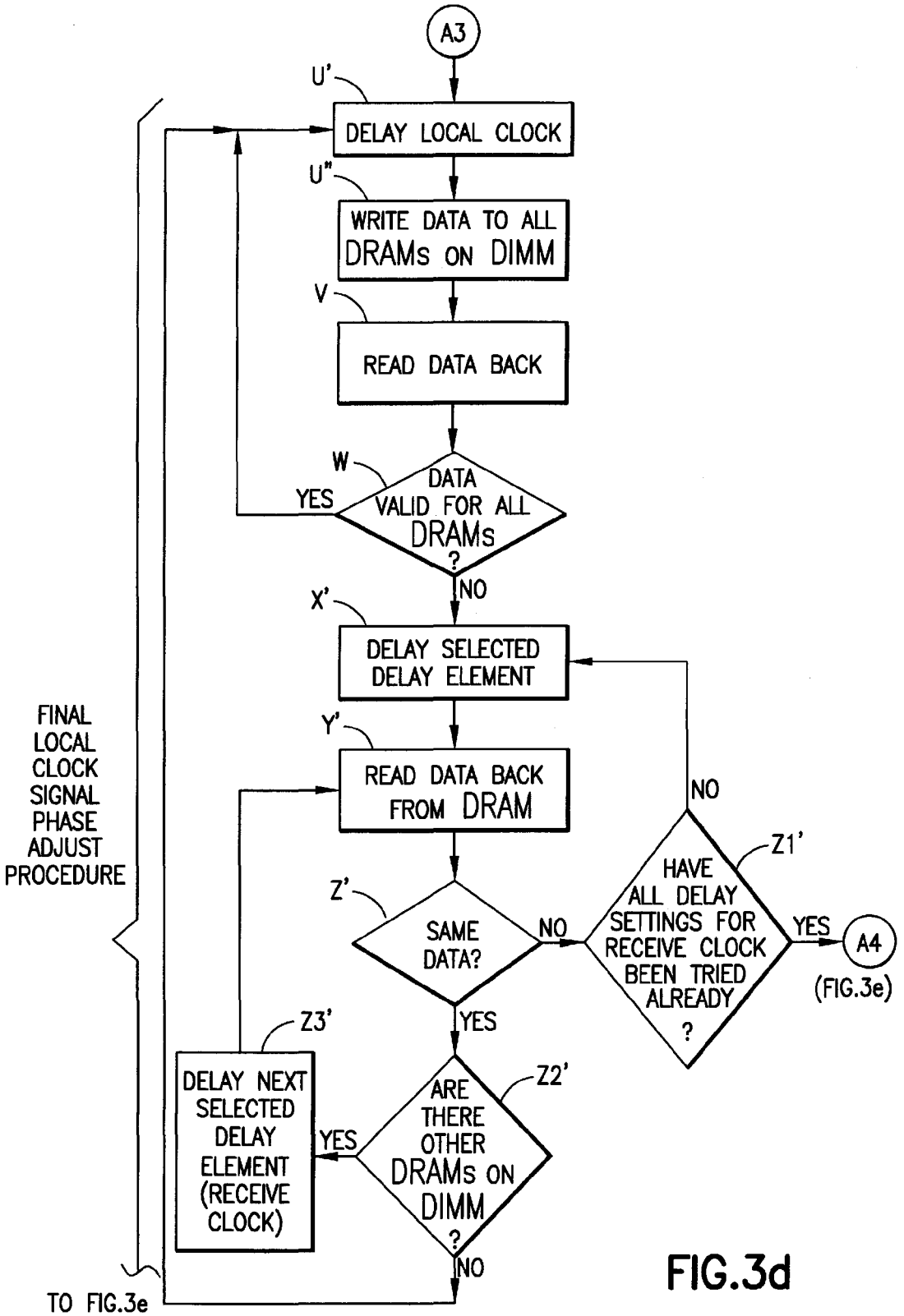


FIG.3d

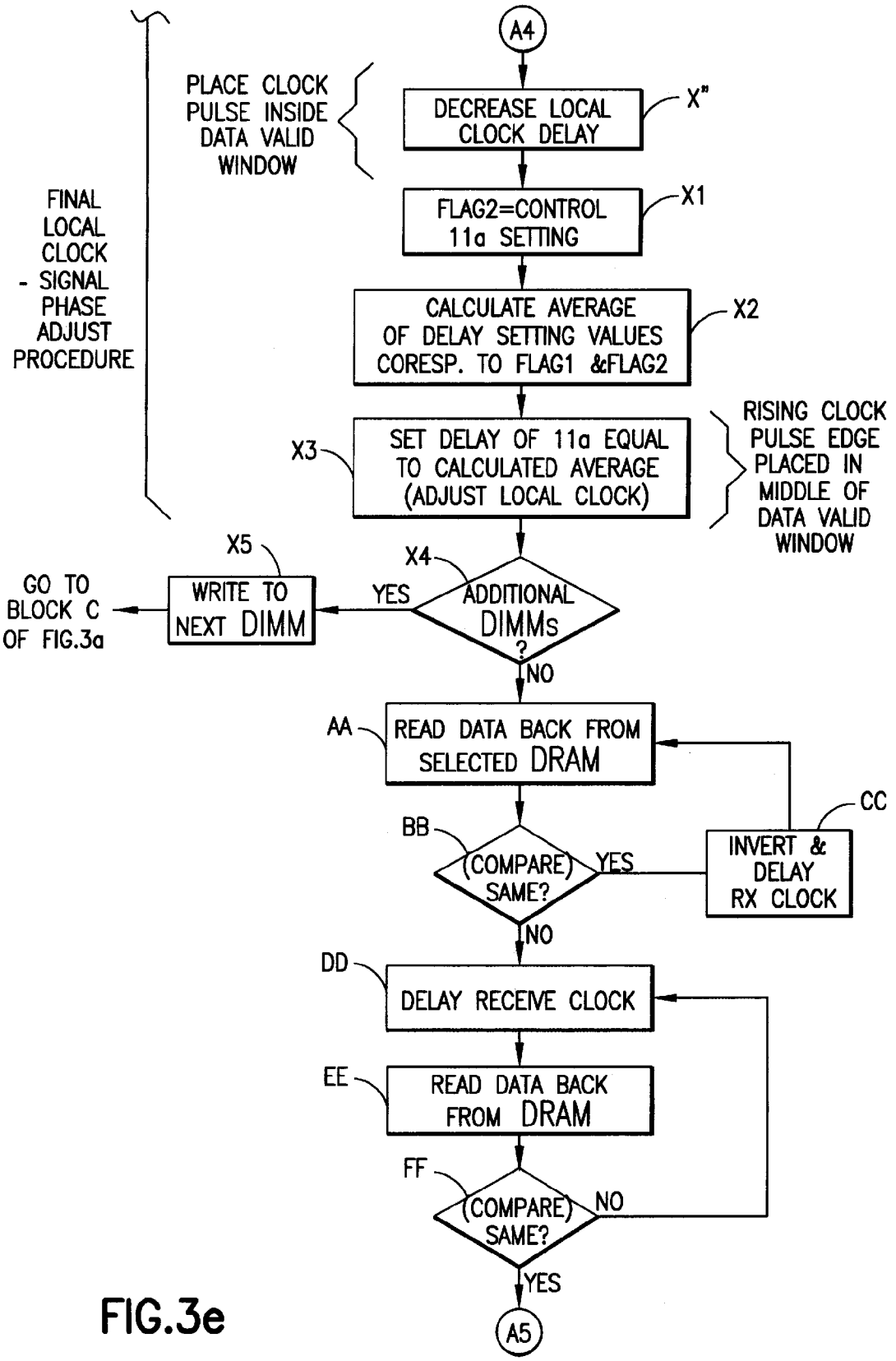


FIG.3e

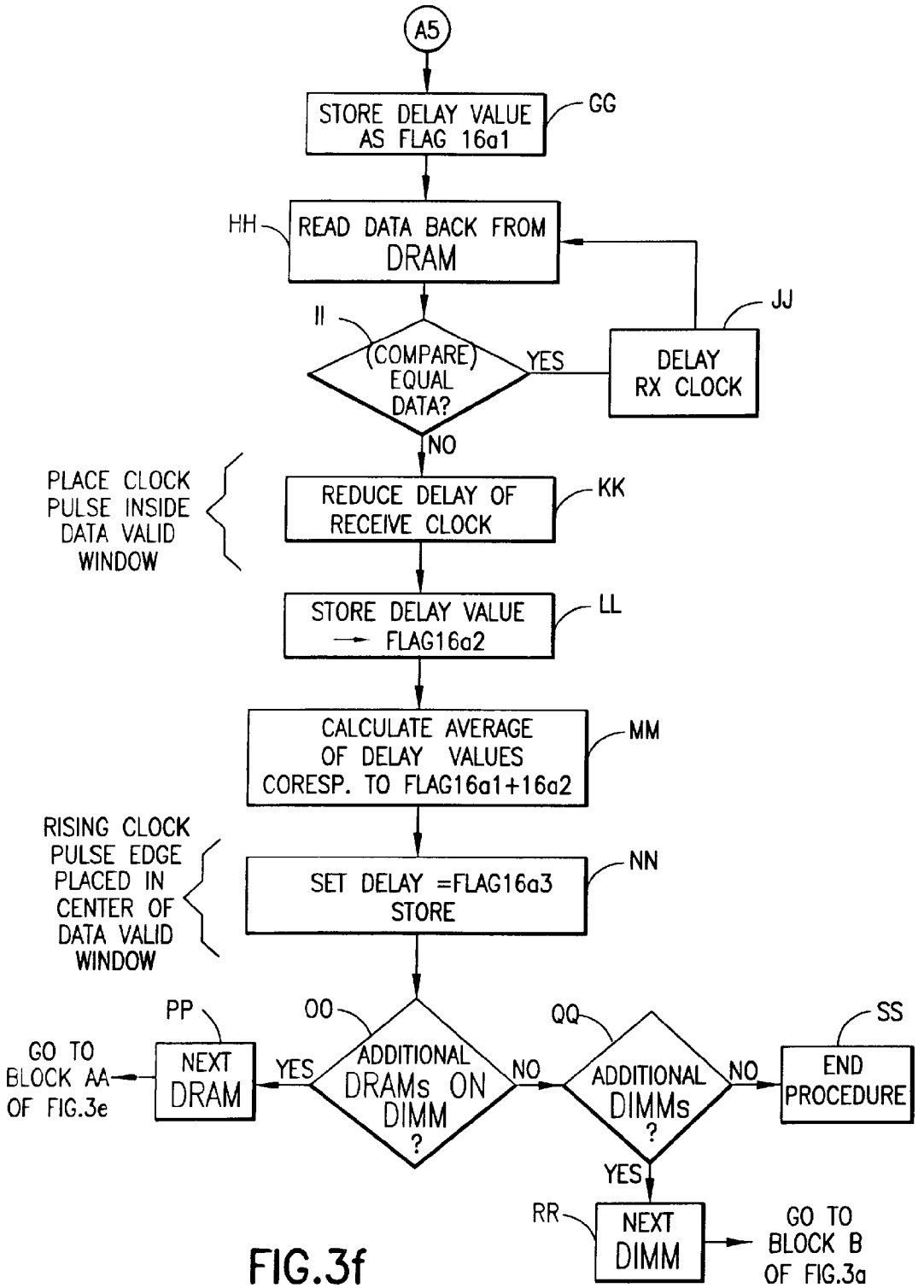

A5
STORE DELAY VALUE AS FLAG 16a1
GG
HH
READ DATA BACK FROM DRAM
II
(COMPARE) EQUAL DATA?
YES
DELAY RX CLOCK
JJ
NO
PLACE CLOCK PULSE INSIDE DATA VALID WINDOW
REDUCE DELAY OF RECEIVE CLOCK
KK
STORE DELAY VALUE → FLAG16a2
LL
CALCULATE AVERAGE OF DELAY VALUES CORESP. TO FLAG16a1+16a2
MM
RISING CLOCK PULSE EDGE PLACED IN CENTER OF DATA VALID WINDOW
SET DELAY =FLAG16a3 STORE
NN
PP
OO
QQ
SS
GO TO BLOCK AA OF FIG.3e
NEXT DRAM
YES
ADDITIONAL DRAMs ON DIMM ?
NO
ADDITIONAL DIMMs ?
NO
END PROCEDURE
YES
RR
NEXT DIMM
GO TO BLOCK B OF FIG.3a

FIG.3f

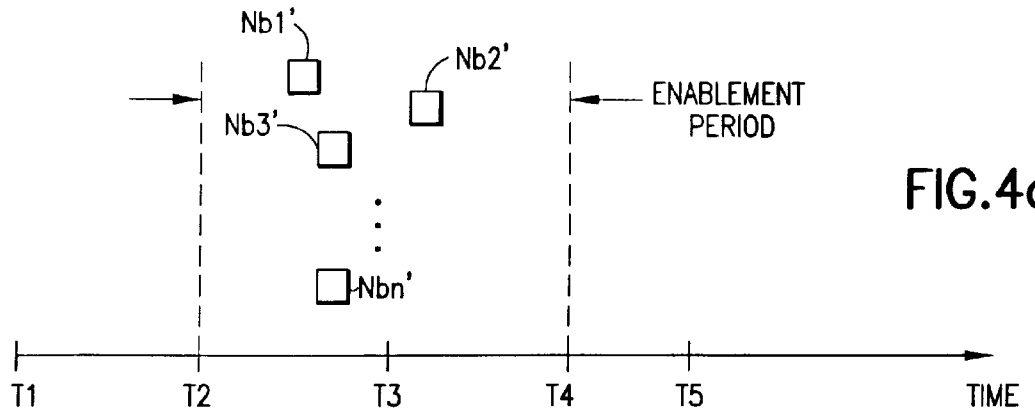


FIG.4a

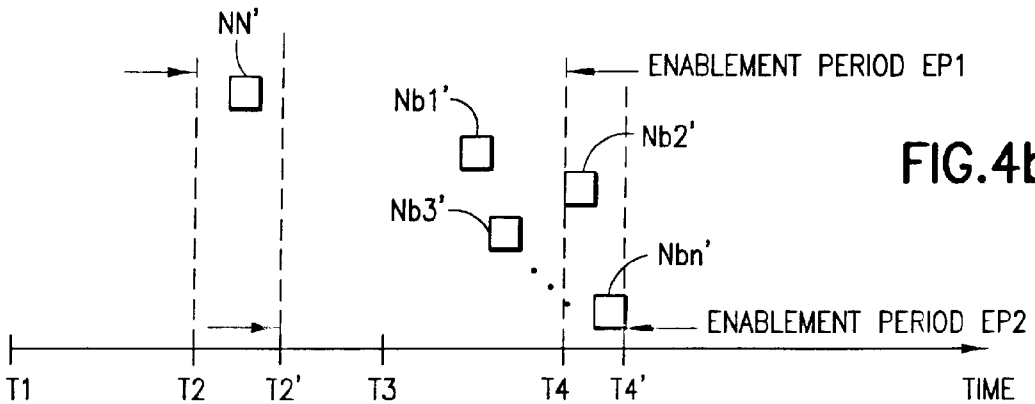


FIG.4b

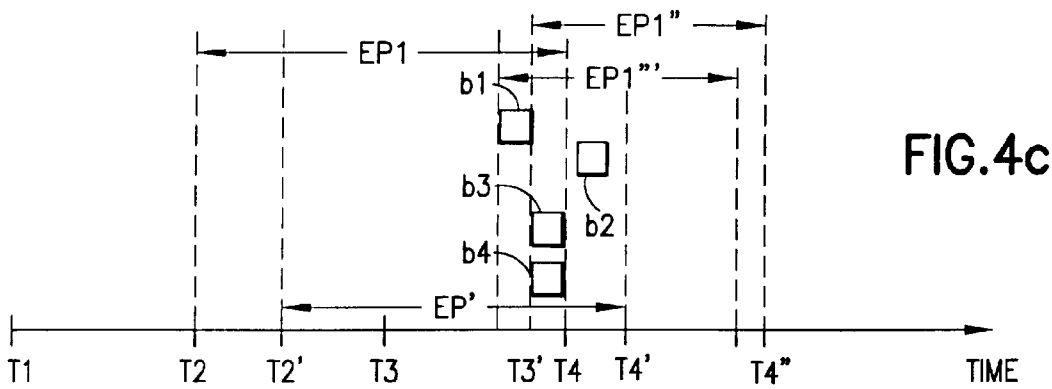


FIG.4c

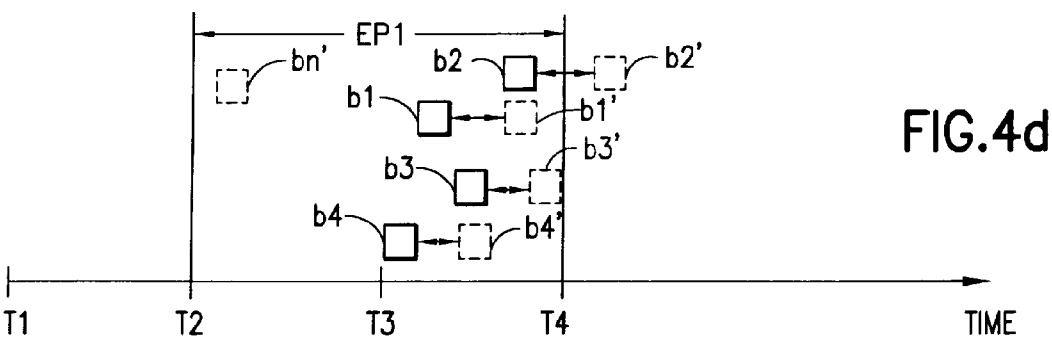


FIG.4d

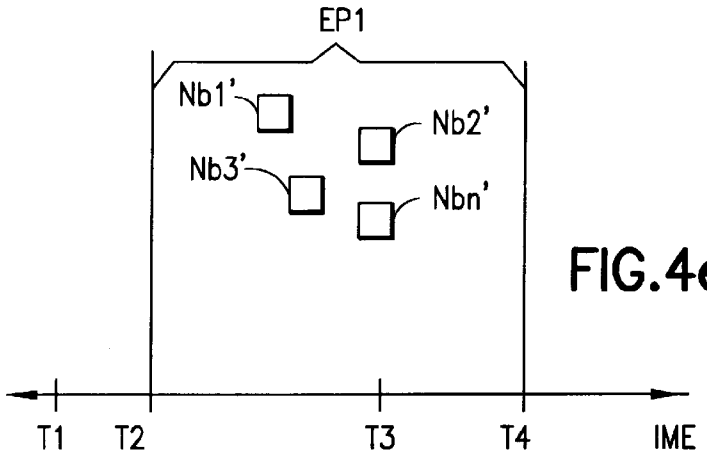


FIG.4e

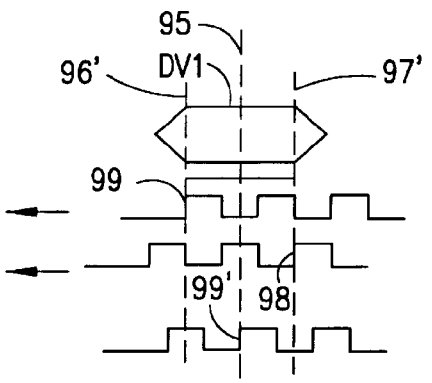


FIG.4f

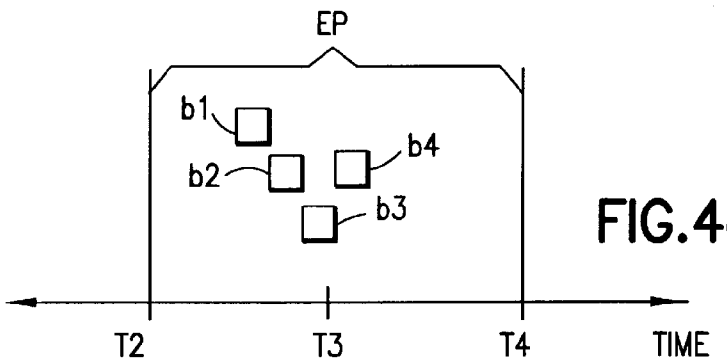


FIG.4g

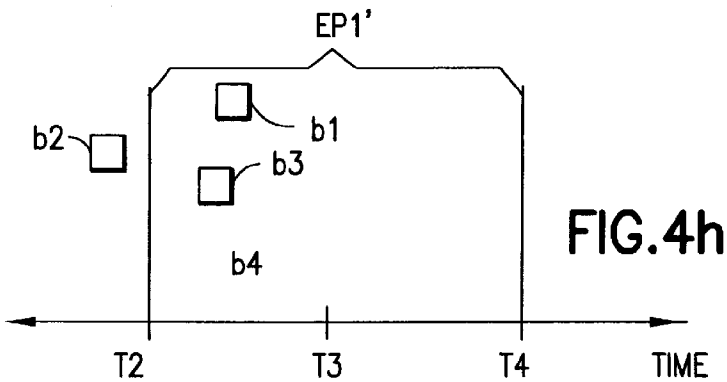


FIG.4h

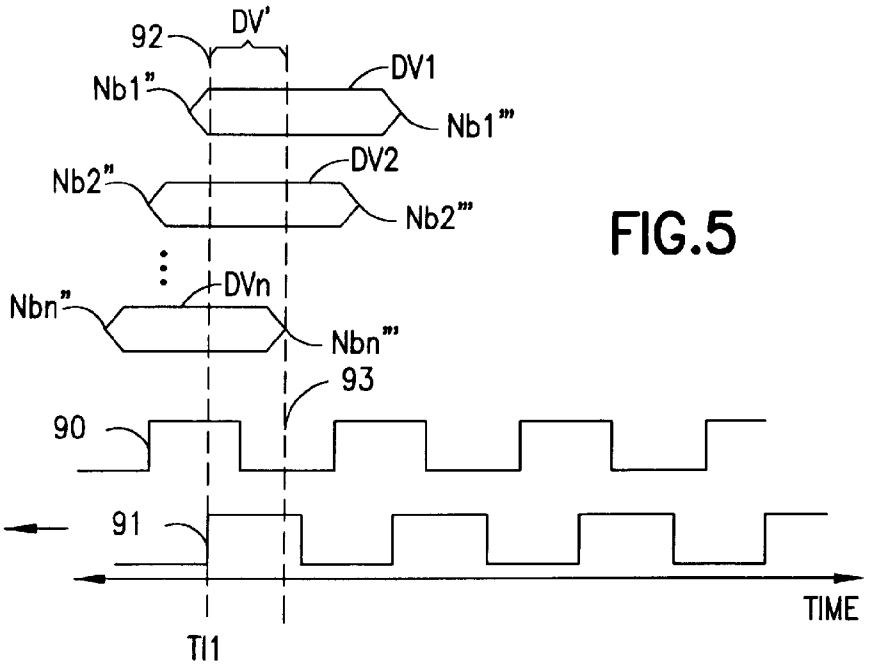


FIG.5

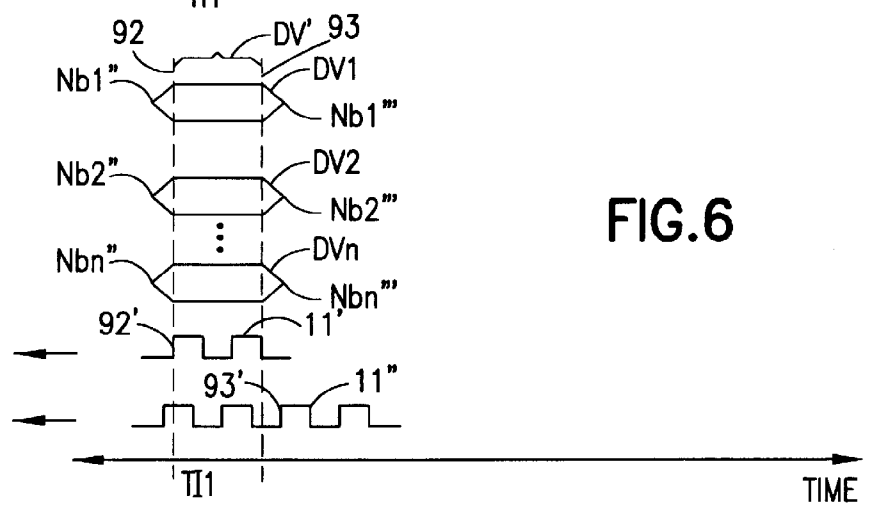


FIG.6

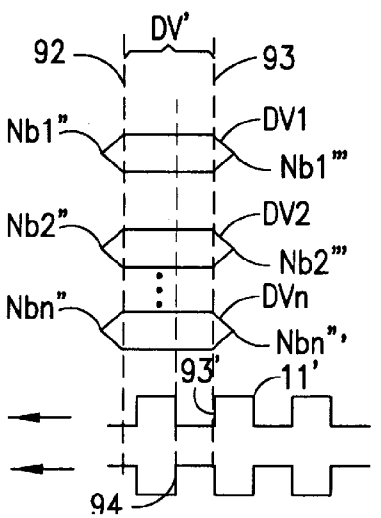


FIG.7

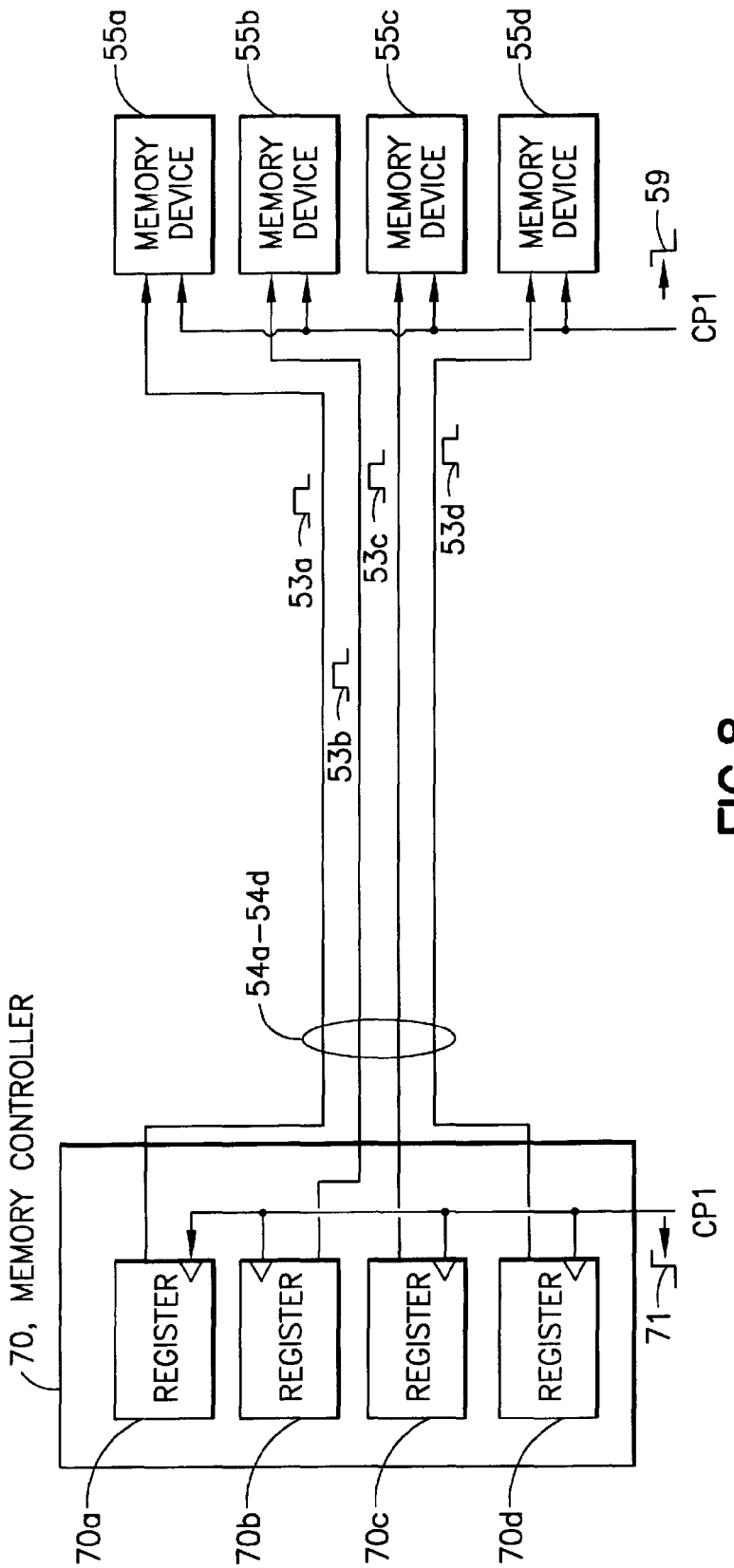


FIG.8
PRIOR ART

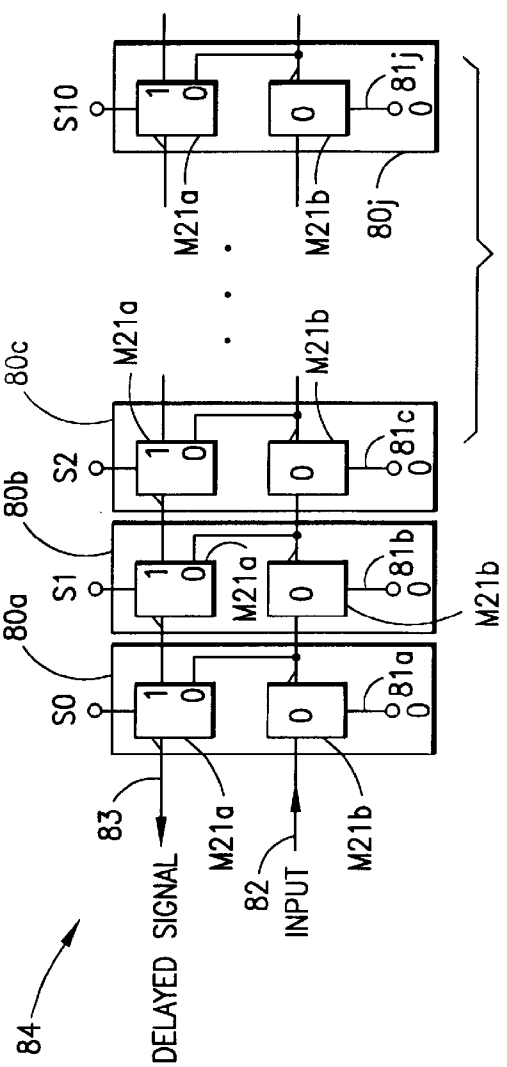


FIG.9a

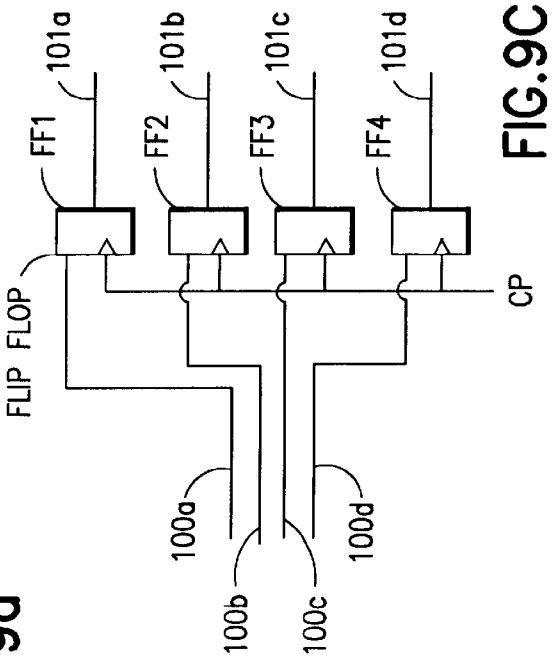


FIG.9C

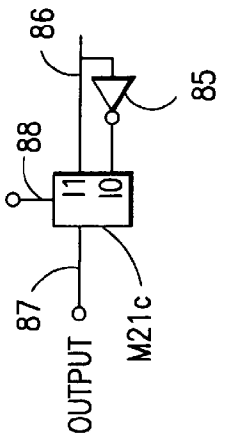


FIG.9B

SMART MEMORY INTERFACE

Priority is herewith claimed under 35 U.S.C. 119(e) from copending Provisional Patent Application Ser. No. 60/052,044, filed on Jul. 9, 1997, entitled "Smart Memory Interface", by Paul W. Coteus, Daniel M. Dreps, and Frank D. Ferraiolo. The disclosure of this Provisional Patent Application is incorporated by reference herein in its entirety.

FIELD OF THE INVENTION

This invention relates generally to computer systems, and, in particular, to a technique for optimizing the performance of a memory subsystem of a computer system.

BACKGROUND OF THE INVENTION

In modern computer systems, the need to perform data storage and retrieval operations at high speeds is often critical. Unfortunately, however, these rates can often be limited by, for example, the limited speeds at which at least some conventional memory control systems operate during the performance of such data storage and retrieval operations.

Various factors can limit the speeds at which memory control systems operate. By example, in at least some conventional memory control systems, wherein data is transferred between memory controller and memory device components during 'read' and 'write' operations, there may be differences between the times at which portions of the data arrive at the individual components. These differences may result from, for example, the use of interface buses having different lengths for coupling the memory controller to the memory device components, and/or the presence of variations in the amount of data loading on the buses. Such differences can cause errors to occur during the 'read' and/or 'write' operations, and can limit the operating speed of the memory control system since, for example, a microprocessor of the computer system may need to delay performing an operation until all of the data portions are successfully 'written' to and/or 'read' from the memory.

Reference is now made to FIG. **8**, which shows various components of a memory control system (hereinafter referred to as a "memory subsystem") of a conventional computer system. In particular, FIG. **8** shows a memory controller **70** that is coupled to memory devices **55***a*–**55***d* through buses **54***a*–**54***d*, respectively. Memory controller **70** is employed for 'writing' data to, and for 'reading' data from, the memory devices **55***a*–**55***d* through the buses **54***a*–**54***d*. The memory controller **70** includes 4-bit registers **70***a*–**70***d*, which are assumed to have a capability for being enabled for a predetermined time period (also referred to as an "enablement period"), in response to receiving individual positive edges **71** of a pulsed clock signal through input CP1. Data that is received by the registers **70***a*–**70***d* during the enablement period is accepted (i.e., loaded) by these devices **70***a*–**70***d*, for subsequent transfer to, for example, a microprocessor (not shown).

Memory devices **55***a*–**55***d* are assumed to be memory chips, such as, for example, Dynamic Random Access Memory (DRAM) chips, and are each assumed to have a capability for being enabled for a predetermined time period, for accepting (i.e., loading) data received over buses **54***a*–**54***d*, in response to receiving individual positive edges **59** of a pulsed clock

As was previously described, in at least some conventional memory subsystems, such as the one represented in FIG. **8**, there may be variations in the lengths of the buses

54*a*–**54***d* coupling the devices **70** and **55***a*–**55***d*. These variations may be a result of, for example, the use of memory devices **55***a*–**55***d* and associated buses **54***a*–**54***d* manufactured in accordance with different manufacturing tolerances/specifications. The variations in the lengths of the buses **54***a*–**54***d* can cause data **53***a*–**53***d* that is simultaneously transmitted from the registers **70***a*–**70***d* of memory controller **70** during a write operation, to eventually arrive at the respective memory devices **55***a*–**55***d* at different times, and at times that are not within a duration of a same enablement period of the respective memory devices **55***a*–**55***d*. This can result in 'write' errors. A similar problem can also arise during 'read' operations where data is provided from the memory devices **55***a*–**55***d* to the memory controller **70**, resulting in 'read' errors.

For memory subsystems that include multiple memory modules (e.g., dual in-mode memory modules), wherein one or more memory devices **54***a*–**55***d* are arranged on the memory modules, different ones of the memory modules may be manufactured in accordance with different manufacturing tolerances/criteria. As a consequence, there may be a great number of variations between the lengths of buses employed for coupling a memory controller to the different memory modules. As such, in these types of memory subsystems the above-described problems can be even more severe.

It is known to increase the speeds at which memory subsystems operate by employing techniques for reducing the overall amount of time required to successfully read and write data to individual memory chips, and by employing parallel memory chips. Extended-Data-Out (EDO) mode memory devices, Synchronous Dynamic Random Access Memory (SDRAM) devices, and Synchronous Dynamic Random Access Memory Double Data Rate (SDRAM-DDR) devices are examples of recent developments for increasing memory subsystem operating speeds. Memory subsystems that employ SDRAMs are synchronous (i.e., data is sent upon an occurrence of a positive edge of a clock signal pulse, and data is received upon an occurrence of a positive edge of a different clock signal pulse). In memory subsystems employing SDRAM-DDR devices, data can be sent upon an occurrence of a positive edge of a clock signal pulse, and received upon a negative edge of the same clock signal pulse. This capability allegedly reduces subsystem latency in half relative to the latency of subsystems that do not employ SDRAM-DDR devices. Memory subsystems that include SDRAM-DDR devices typically employ so called data strobes, which are sent along with data being transferred. Unfortunately, such data strobes require the use of extra pins and wiring, can increase system latency, and can cause an increase in the amount of time required for the memory subsystem to transition between 'read' and 'write' operations. In view of the foregoing considerations, it can be appreciated that it would be desirable to provide a technique which optimizes the performance of a memory subsystem by overcoming the above-described problems. It would also be desirable that the technique not require the use of extra signals or additional memory device circuitry, or require an increase in system latency, or an increase in the length of time needed to transition between 'read' and 'write' operations.

OBJECT OF THE INVENTION

It is an object of this invention to provide a technique which determines optimum temporal relationships of electrical (clock) signals employed for operating a memory control system of a computer system, for enabling the

determined optimum temporal relationships to be subsequently used for operating the memory control system without error.

It is another object of this invention to provide a technique which compensates for differences in times at which portions of data transmitted from one component of a memory control system arrive at another, destination component of the memory control system, for enabling data transfer errors to be minimized.

SUMMARY OF THE INVENTION

The foregoing and other problems are overcome and the objects of the invention are realized by a method, and an apparatus that operates in accordance with the method, for initiating a start-up operation of a computer system having a memory control system. The memory control system includes a memory device and a memory controller which writes data to, and reads data from, the memory, as needed during the operation of the computer system. The method comprises steps of: A) exercising the memory device using the memory controller to determine a temporal range within which temporal relationships of electrical signals (e.g., clock signals) need to be set in order to operate the memory control system without error; B) setting the temporal relationships of the electrical signals so as to be within the determined temporal range; and C) storing a record of the determined temporal range, for subsequent use in operating the memory control system.

The method of the invention compensates for any differences in times at which portions of data being transferred from the memory controller to the memory device, and vice versa, arrive at the respective destination components, and minimizes the possibility of read/write errors being encountered. The method of the invention also enables the overall processing speed of the memory control system (and the computer system in general) to be increased, as the differences in the arrival times of the data are compensated for.

BRIEF DESCRIPTION OF THE DRAWINGS

The above set forth and other features of the invention are made more apparent in the ensuing Detailed Description of the Invention when read in conjunction with the attached Drawings, wherein:

FIG. **1**, depicts a memory subsystem of a computer system that is suitable for practicing this invention.

FIG. **2** shows a relationship between FIGS. **2***a* and **2***b*.

FIGS. **2***a* and **2***b* depict the memory subsystem of FIG. **1** in greater detail.

FIG. **2***c* depicts various clock signals employed in the memory subsystem of FIG. **1**, after having been temporally displaced by delay elements of the memory subsystem, and further depicts information provided to the delay elements for placing the delay elements in settings corresponding to the temporal displacements.

FIG. **2***d* depicts another example of various clock signals employed in the memory subsystem of FIG. **1**, after having been temporally displaced by delay elements of the memory subsystem of FIG. 1.

FIGS. **3***a*–**3***f* are logical flow diagrams depicting a method in accordance with this invention.

FIGS. **4***a* and **4***b* depict an exemplary relationship between times at which enablement periods for registers of the memory subsystem of FIG. 1 occur, relative to times at which data is received by these registers.

FIGS. **4***c* and **4***d* depict an exemplary relationship between times at which enablement periods for registers and memory devices of the memory subsystem of FIG. **1** occur, relative to times at which bits of data are received by these registers and memory devices.

FIG. **4***e* depicts another exemplary relationship between times at which enablement periods for memory devices of the memory subsystem of FIG. **1** occur, relative to times at which data is received by these memory devices.

FIG. **4***f* represents an exemplary relationship between clock signals employed in the memory subsystem of FIG. **1** and a "data valid window", the data valid window representing a time period that extends between a first, earliest time at which register components within the memory subsystem may be triggered for enabling data to be read from memory devices of the memory subsystem without error, and a second, latest time at which the register components may be triggered for enabling data to be read from the memory devices without error.

FIGS. **4***g* and **4***h* depict further exemplary relationships between times at which enablement periods for register components of the memory subsystem of FIG. 1 occur, relative to times at which bits of data are received by these register components.

FIGS. **5**, **6**, and **7** represent exemplary relationships of clock signals employed the memory subsystem of FIG. **1** and various "data valid windows" for states of the memory subsystem corresponding to the beginning of the procedures of FIGS. **3***c* and **3***d*, and to an end of the procedures of FIG. **3***f*, respectively, wherein the data valid windows represent time periods extending between respective first, earliest times at which registers within the memory subsystem may be triggered for enabling data to be read from memory devices of the memory subsystem without error, and respective second, latest times at which the registers may be triggered for enabling data to be read from the memory devices without error.

FIG. **8** shows a memory controller and memory device components of a conventional memory subsystem of a typical computer system.

FIGS. **9** and **9***b* show portions of delay elements of the memory subsystem of FIG. **1***a*.

FIG. **9***c* shows flip-flops of registers of the memory subsystem of FIG. **1**.

DETAILED DESCRIPTION OF THE INVENTION

FIG. **1** depicts a memory subsystem **1**' of a computer system **1**" that is suitable for practicing this invention. The memory subsystem **1**' comprises a microprocessor **2** that is bidirectionally coupled to a memory **3**, a data table T1, and a memory controller **1**, and also comprises various memory modules which, in a preferred embodiment of the invention, include dual in-line memory modules (DIMMs) **14***a*–**14***n*. Preferably, each DIMM **14***a*–**14***n* includes a plurality of memory devices, which in the preferred embodiment include Dynamic Random Access Memories (DRAMs) D1–Dn. The memory **3** is assumed to store an operating program for the microprocessor **2**, flag variables (e.g., FLAG**1**, FLAG**2**, FLAG**3**, FLAG**15***a*–FLAG**15***n*, and FLAG**16***a***1**–FLAG**16***n***3**), information identified as (Nbb1) –(Nbbn) (hereinafter also referred to as "data Nbb1–Nbbn"), and other information that is received from the microprocessor **2** during the performance of a method in accordance with this invention. The flag variables and the information identified as (Nbb**1**)–(Nbbn) are employed in a manner as will be described below.

The data table T**1** stores information specifying various values V**1**–V**10**, which are also referred to herein as prede-

termined phase delay values V1–V10, and corresponding command information. The information specifying the predetermined phase delay values V1–V10 and the corresponding command information is represented in FIG. 2*c*, and is employed in the method of the invention in a manner as will be described below.

The microprocessor **2** is assumed to control the operations of the computer system 1" in general, and is also assumed to control the memory controller **1** and the various DIMMs **14***a*–**14***n* for writing (i.e., storing) data to, and for reading data from, the DRAMs D1–Dn of DIMMs **14***a*–**14***n*, when required during the operation of the computer system **1**". The microprocessor **2** also controls the timing of these operations, using clock signals. In other embodiments, the block **2** may represent a controller of the memory subsystem **1**' that operates in accordance with instructions provided from a microprocessor of the computer system **1**".

A control/address bus **102***a*, clock buses **103***a*–**103***n*, and data buses **12** and **13**, are also provided. The control/address bus **102***a* is preferably a multi-drop bus, and is employed for carrying control and address information from the memory controller **1** to the DRAMs D1–Dn of the DIMMs **14***a*–**14***n* during write and read operations. Clock buses **103***a*–**103***n* are employed for providing clock signals to the DRAMs D1–D*n* of the respective DIMMs **14***a*–**14***n*.

Data buses **12** and **13** are also preferably multi-drop buses, and are employed for carrying data being exchanged between the memory controller **1** and memory locations within DRAMs D1–D*n* of the DIMMs **14***a*–**14***n*, during read and write operations. It should be noted that each of the buses **102***a*, **12**, and **13** may be a single bus that is coupled to the DRAMs D1–D*n* of each DIMM **14***a*–**14***n* or may include a plurality of respective buses that are coupled to DRAMs D1–D*n* of respective ones of the DIMMs **14***a*–**14***n*. Also, it should be noted that in other embodiments of the invention, there may be more than a single clock bus **103***a*–**103***n* provided between the memory controller **1** and each respective DIMM **14***a*–**14***n*, if more than a single clock signal is provided to each DIMM **14***a*–**14***n*. For simplicity, only a single clock bus **103***a*–**103***n* is shown as being coupled to each individual DIMM **14***a*–**14***n*. Preferably, the buses which are coupled to single ones of the DIMMs **14***a*–**14***n* have similar load-carrying capabilities.

Reference is now made to FIGS. **2***a* and **2***b* which depict the memory subsystem **1**' in further detail. For convenience, only two DIMMs **14***a* and **14***n*, and only the DRAMs D1 and D*n* of DIMM **14***a*, are shown in FIGS. **2***a* and **2***b*, although it is assumed for the purposes of this description that the subsystem **1**' includes DIMMs **14***a*–**14***n*, and that each DIMM **14***a*–**14***n* includes one or more DRAMs D1–D*n*. It should be noted that any suitable number of DIMMs **14***a*–**14***n*, and any suitable number of DRAMs D1–D*n* per DIMM **14***a*–**14***n*, may be employed in the memory subsystem **1**', depending on applicable computer system performance criteria. Also, memory locations within each DRAM D1–D*n* are hereinafter referred to as memory locations ML1–ML*n*.

In accordance with a preferred embodiment of the invention, the memory controller **1** includes memory controller component blocks (also referred to as circuit blocks) **110***a*–**110***n*, which correspond to the DIMMs **14***a*–**14***n*, respectively. According to a preferred embodiment of the invention, each of the memory controller component blocks **110***a*–**110***n* comprises registers **10***a***1**, **10***a***2***a*–**10***a***2***n*, **10***a***3***a*–**10***a***3***n*, **10***b***1**–**10***bn*, and **10***c***1**–**10***cn*, delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n*, buses **11***b***1**–**11***bn*, drivers **8***a*–**8***n*, and receivers **9***a*–**9***n*. These various components are interconnected within each block **110***a*–**110***n* in the manner shown in FIGS. **2***a* and **2***b*. The delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* have respective inputs **11***a*', **15***a*'–**15***n*', and **16***a*'–**16***n*', which are each assumed to be coupled to an output (not shown) of the microprocessor **2**, although this is not shown in FIGS. **2***a* and **2***b* in order to simplify the depiction of the subsystem **1**'. Similarly, the registers **10***b***1**–**10***bn* have respective inputs **10***b***1**"–**10***bn*" that are each assumed to be coupled to an output of the microprocessor **2**, and the registers **10***c***1**–**10***cn* have respective outputs **10***c***1**'–**10***cn*' that are each assumed to be coupled to an input (not shown) of the microprocessor **2**, although this is also not shown in FIGS. **2***a* and **2***b* in order to simplify the depiction of the subsystem **1**'. It should further be noted that for convenience, only the various components of memory controller component block **110***a* are shown in FIGS. **2***a* and **2***b*.

FIG. **2***b* also shows buses (N–1)a and (N)a which couple the DRAM Dn of DIMM **14***n* to driver and receiver components **8***n* and **9***n*, respectively, of the memory controller **1**. The memory controller **1** writes data to, and receives data from, the DRAM DN through these respective buses (N–1)a and (N)a. FIG. **2***b* further shows a bus **11***n*, which couples bus **11** to memory controller component block **110***n*, and which is assumed to singularly represent a plurality of buses similar to buses **11***b***1**–**11***bn*. In addition, FIG. **2***b* shows the DIMM **14***n*, and an interface **200** which couples the DIMM **14***n* to the memory controller **1**. The interface is assumed to represent various buses required for coupling the DRAMs (not shown) of the DIMM **14***n* to memory controller component block **110***n*.

According to a preferred embodiment of the invention, each register **10***a***1**, **10***a***2***a*–**10***a***2***n*, **10***a***3***a*–**10***a***3***n*, **10***b***1**–**10***bn*, and **10***c***1**–**10***cn* is "enabled" for a predetermined time period (also referred to as an "enablement period") in response to receiving a positive edge of a clock pulse, for accepting (i.e., loading) data received at a respective input **10***a***1**", **10***a***2***a*'–**10***a***2***n*", **10***a***3***a*"–**10***a***3***n*", **10***b***1**"–**10***bn*", and **10***c***1**"–**10***cn*" of the register, for subsequent transfer to an output of the register. According to a preferred embodiment of the invention, each register **10***a***1**, **10***a***2***a*–**10***a***2***n*, **10***a***3***a*–**10***a***3***n*, **10***b***1**–**10** *bn*, and **10***c***1**–**10***cn* is a 4-bit register and includes flip-flips FF1–FF4, which are depicted in FIG. **9***c*. In FIG. **9***c*, inputs **100***a*–**100***d* and outputs **101***a*–**101***d* are shown. The inputs **100***a*–**100***d* are assumed to collectively represent to individual ones of the data inputs **10***a***1**", **10***a***2***a*"–**10***a***2***n*", **10***a***3***a*"–**10***a***3***n*", **10***b***1**"–**10***bn*", and **10***c***1**"–**10***cn*", of the respective registers **10***a***1**, **10***a***2***a*–**10***a***2***n*, **10***a***3***a*–**10***a***3***n*, **10***b*–**10***bn*, and **10***c***1**–**10***cn*, and the outputs **101***a*–**101***d* shown in FIG. **9***c* are assumed to collectively represent individual data outputs of these registers. Also, a clock signal input (CP) of FIG. **9***c* is assumed to correspond to an individual clock pulse input of the individual registers **10***a***1**, **10***a***2***a*–**10***a***2***n*, **10***a***3***a*–**10***a***3***n*, **10***b***1**–**10***bn*, and **10***c***1**–**10***cn*. Preferably, the flip-flops FF1–FF4 are positive-edge-triggered flip-flops such as, for example, D-type positive-edge-triggered flip-flops, although in other embodiments positive-edge-triggered master-slave flip-flops may also be employed. Each flip-flop FF1–FF4 is preferably responsive to a positive edge of a clock signal being applied to input (CP) for being enabled for a predetermined time period (referred to hereinafter as an "enablement period"), during which time period bits of data received at the inputs **100***a*–**100***d* of the respective flip-flops FF1–FF4 are accepted (i.e., loaded), for subsequent transfer to a respective output **101***a*–**101***d*.

As was previously described, the microprocessor **2** controls the timing of operations performed by the memory controller **1** and DIMMs **14***a*–**14***n*, using clock signals. To this end, the microprocessor **2** includes a local clock signal generator **2**' that generates a pulsed local clock signal **11**'. The generated local clock signal **11**' is output from the microprocessor **2** to each memory controller component block **110***a*–**110***n* through the bus **11** and the buses **11***b***1**–**11***bn*. Within each memory controller component block **110***a*–**110***n*, the local clock signal **11**' is provided to registers **10***b***1**–**10***bn* and **10***c***1**–**10***cn*, and to respective inputs **11***a*, **15***a*–**15***n*", and **16***a*"–**16***n*" of the respective delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n*. Also, the local clock signal **11**' is provided to the registers **10***a***1**, **10***a***2***a*–**10***a***2***n*, and **10***a***3***a*–**10***a***3***n* through the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n*, as can be appreciated in view of FIGS. **2***a* and **2***b*. The local clock signal **11**' is also provided to each DRAM D1–Dn of DIMM **14***a* through the delay element **11***a*, driver **8***b*, and bus **103***a*.

The microprocessor **2** has a capability for varying the amount of temporal displacement (hereinafter also referred to as "phase delay") provided by each individual delay element **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* to the local clock signal **11**' applied to the delay element (i.e., each delay element is programmable by the microprocessor **2**). To this end the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* have the respective inputs **11***a*', **15***a*'–**15***n*', and **16***a*'–**16***n*', which, as was previously described, are each coupled to an output (not shown) of the microprocessor **2**. Each delay element **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* preferably has multiple delay "settings", individual ones of which may be selected by the microprocessor **2** during the operation of the method of the invention. In a preferred embodiment, and assuming that local clock signal **11**' has a period of (T), each delay element **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* has a capability for temporally displacing the signal by a temporal displacement that is at least as small as (1/10)(T). This displacement is hereinafter referred to as a "predetermined phase delay amount", and is represented by "V1" in FIG. **2***c*. Reference is now made to FIG. **9***a* which shows a delay element **84** that is constructed in accordance with a preferred embodiment of the invention. The delay element **84** comprises a plurality of delay portions **80***a*–**80***j*, each of which includes first and second multiplexers M**21***a* and M**21***b*, respectively. The first and second multiplexers M**21***a* and M**21***b* are preferably 2-to-1 line multiplexers. The first and second multiplexers M**21***a* and M**21***b* of the respective delay portions **80***a*–**80***j* are connected together in the manner shown in FIG. **9***a*. The first multiplexer M**21***a* of the respective delay portions **80***a*–**80***j* have respective inputs S1–S**10** that are assumed to be connected to an output of the microprocessor **2**, and the second multiplexer M**21***b* of the respective delay portions **80***a*–**80***j* have respective inputs **81***a*–**81***j* that are assumed to be coupled to, for example, a binary '0' provided from microprocessor **2**. In the preferred embodiment of the invention, each of the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* of FIGS. **2***a* and **2***b* is similar to the delay element **84** of FIG. **9***a*, and each input **11***a*', **15***a*'–**15***n*', and **16***a*'–**16***n*' of the respective delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* is assumed to singularly represent the collective inputs S1–S**10** of delay element **84**. Throughout this description, reference labels S1–S**10** are used interchangeably with individual ones of the reference labels **11***a*', **15***a*'–**15***n*', and **16***a*'–**16***n*', for identifying delay element control inputs. Also, inputs **11***a*", **15***a*"–**15***n*", and **16***a*"–**16***n*" of respective delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* are each assumed to correspond to input **82** of FIG. **9***a*, and outputs of the respective delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* are each assumed to correspond to output **83** of FIG. **9***a*.

In a preferred embodiment, in addition to the delay portions **80***a*–**80***j*, the delay elements **11***a* and **16***a*–**16***n* also include an inverter **85** and another multiplexer M**21***c*, both of which are connected together in the manner shown in FIG. **9***b*. For these delay elements **11***a* and **16***a*–**16***n*, an input **86** (FIG. **9***b*) is assumed to be coupled to output **83** of the delay portion **80***a* of FIG. **9***a*, and the multiplexer M**21***c* has an output **87** that is assumed to represent the individual outputs of the respective delay elements **11***a* and **16***a*–**16***n*. Between the input **86** and an input (I**0**) of the multiplexer M**21***c* is coupled the inverter **85**. Input **86** is also coupled to an input (I1) of the multiplexer M**21***c*. Furthermore, the multiplexer M**21***c* includes an input **88** that is assumed to be coupled to the microprocessor **2**.

As was previously described, each delay element **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* preferably has multiple delay "settings", individual ones of which may be selected by the microprocessor **2** using information applied to the inputs **11***a*', **15***a*'–**15***n*', and **16***a*'–**16***n*' (i.e., inputs S1–S**10**) of the delay elements. The amount of temporal displacement (hereinafter referred to as "phase delay") (V1–V**10**) imparted by the individual delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* to received signals, for each of these delay settings, and an example of information provided by the microprocessor **2** to the inputs S1–S**10** of the individual delay elements, for causing these delay elements to be placed in the respective delay settings, is represented in FIG. **2***c*.

The microprocessor **2** also has a capability for controlling whether or not particular ones of the delay elements **11***a* and **16***a*–**16***n* invert signals that are applied to the input **86**, for providing either non-inverted or inverted versions of the signals through output **87**. By example, for selecting a non-inverted version of a signal applied to input **86**, the microprocessor **2** applies a binary '1' to input **88** of multiplexer M**21***c*. For selecting an inverted version of the signal, the microprocessor **2** applies binary '0' to input **88** of multiplexer M**21***c*, as can be appreciated in view of FIG. **9***b*.

The microprocessor **2** also has a capability for operating in conjunction with the memory controller **1** for selecting a memory location ML1–ML*n* of a DRAM D1–D*n* from a particular DIMM **14***a*–**14***n*, for writing data to, and for subsequently reading data from, this memory location, as needed during the operation of computer system **1**". For example, it is assumed that, during the operation of the computer system **1**", it is required that data be written from the microprocessor **2** to a particular one of the memory locations ML**1** of DRAM D**1** from DIMM **14***a*. In this case, the microprocessor **2** provides data **10**' to an input of register **10***a***1**, through a bus **10**. The data **10**' includes an address of the memory location ML**1**, and a command indicating that data is to be written to this memory location ML**1**. After register **10***a***1** receives this data **10**', and in response to the register **10***a***1** also receiving a positive edge of the local clock signal **11**' through input **10***a***1**', the register **10***a***1** outputs the data **10**' to the driver **8***a*, which then responds by buffering the data **10**' for driving the data **10**' from the memory controller **1**, through the bus **102***a*, and to the DRAMs D1–Dn of the DIMM **14***a*. After the data **10**' is received by DRAM D**1**, the DRAM D**1** is assumed to recognize that further data is to be written to memory location ML**1** of DRAM D**1**, based on the command and address information included in the received data **10**'.

Thereafter, the microprocessor **2** provides other data (i.e., data that is to be written to the DRAM D**1**) to register **10***b***1**

through input **10***b***1**". Then, after each of the registers **10***b***1** and **10***a***2***a* receives respective positive edges of the local clock signal **11**', the data is forwarded through the registers **10***b***1** and **10***a***2***a* to the driver **8***c*, which then buffers the data for driving the signal from the memory controller **1**, through the bus **104***a*, and to the DRAM D1 for subsequent storage therein in the memory location ML1. Each DRAM D1–Dn is assumed to have a capability for loading (also referred to as accepting) data received through a respective one of the buses **104***a*–(N–1), for subsequent storage therein, in response to receiving positive edges of local clock signal **11**' through bus **103***a* (the number of positive edges depends on the number of bits employed). As such, assuming that the data is received by the DRAM D1, and that the DRAM D1 also receives a positive edge of the local clock signal **11**' through bus **103***a*, then the DRAM D1 responds by loading the data into the DRAM D1. In this manner, a 'write' procedure is performed, wherein the microprocessor **2** operates in conjunction with the memory controller **1** so as to write data to memory location ML1 within DRAM D1.

As was previously described, the microprocessor **2** also operates in conjunction with the memory controller **1** for reading data from selected memory locations ML1–MLn of selected ones of the DRAMs D1–Dn. As an example, it is assumed that data was already written to the memory location ML1 of DRAM D1, in the manner described above, and that the microprocessor **2** subsequently recognizes that a computer system operation requires that the data be read back to the microprocessor **2** from this memory location ML1. To read back the data, the microprocessor **2** again provides data **10**' to the DRAMs D1–Dn via the register **10***a***1**, driver **8***a*, and bus **102***a*, in the manner described above. However, in this case the data **10**' includes a command specifying that the DRAM D1 provide the data stored in memory location ML1 back to the memory controller **1**.

After the DRAM D1 receives this command, the DRAM D1 retrieves the stored data, and then, in response to receiving a next positive edge of the local clock signal **11**' through bus **103***a*, the DRAM D1 forwards the retrieved data through the bus **105***a* and receiver **9***a* to register **10***a***3***a*. After register **10***a***3***a* receives the data and a positive edge of local clock signal **11**', the data is loaded into (i.e., accepted by) the register. The data is subsequently forwarded from the register **10***a***3***a* to the microprocessor **2** via register **10***c***1**. In this manner, a 'read' operation is performed, wherein data stored in DRAM D1 is read back from the DRAM D1 by the microprocessor **2** operating in conjunction with the memory controller **1**.

Before describing the method of the invention, a brief reference will first be made to some of the problems that are overcome by the invention. As was previously described, in at least some memory subsystems, there may be variations between the lengths of the buses **102***a*, **103***a*, **104***a*–(n–1)a, and **105***a*–((N)a) employed for coupling a memory controller to memory devices. These differences may be a result of, for example, the use of different types of memory devices and associated buses manufactured by different manufacturers. The variations in the bus lengths can cause data that is transferred through different ones of the buses to arrive at destination components at different times, and may ultimately result in portions of the data not being simultaneously loaded into the destination components. As a consequence, read and/or write errors may arise. Also, the variations in the bus lengths may have an affect of limiting the overall processing speed of the computer system, since a component may need to wait to receive data being forwarded to the component over a longest one of the buses, before performing a particular operation, and/or a microprocessor of the computer system may need to wait for a time interval until the read and/or write operations are completed successfully for all memory devices of the computer system before executing a next instruction.

In view of these considerations, the inventors have developed a novel technique which optimizes the performance of a memory subsystem of a computer system, and which overcomes the problems described above.

As was previously described, the memory **3** stores an operating program for controlling the operations of the microprocessor **2**, and for controlling the operations of the memory subsystem **1**' in general. In accordance with this invention, the operating program includes routines for implementing the method of the invention, which is described below in relation to FIGS. **3***a*–**3***f*.

Reference is now made to FIG. **3***a* which illustrates a first portion of the method of the invention. At block A the method is started. It is assumed at block A that the computer system **1**" is powered-on, and that this is recognized by the microprocessor **2**. It is also assumed that the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* are initially programmed so as to provide the delay amount represented by V1 in FIG. **2***c*, and that the local clock signal generator **2**' begins to continuously generate the local clock signal **11**'.

At block B it is assumed that the microprocessor **2** controls the memory controller **1** in the manner described above so as to write data to a particular memory location ML1–MLn of each DRAM D1–Dn of a particular one of the DIMMS **14***a*–**14***n*. Which one of the DIMMs **14***a*–**14***n*, and which one of the memory locations ML1–MLn of the DRAMs D1–Dn within this DIMM, the microprocessor **2** writes data to at block B is not considered to be germane to this invention, and may be determined in accordance with applicable system operating performance criteria. For example, at block B the data may be written to any selected one of the memory locations ML1–MLn of the DRAMs D1–Dn of any selected one of the DIMMs **14***a*–**14***n* coupled to the memory controller **1**. For the purposes of this description, it is assumed that at block B the microprocessor **2** writes data to memory location ML1 of each DRAM D1–Dn of DIMM **14***a*, in the above-described manner.

In a preferred embodiment of the invention, the data written to each DRAM D1–Dn at block B includes predetermined information, such as a data nibble that includes a unique, predetermined bit pattern (also hereinafter referred to as a timing pattern). The predetermined bit pattern is preferably complex enough to include data dependent jitter (i.e., a wide bandwidth of frequencies), and is preferably one that is not likely to be randomly received by the memory controller **1** from the DIMMs **14***a*–**14***n* during the operation of the memory subsystem **1**'. For the purposes of this description, the data written to the respective DRAMs D1–Dn at block B is hereinafter referred to as data (Nb1)–(Nbn), respectively, and it is assumed that the data (Nb1)–(Nbn) is similar to the respective data (Nbb1)–(Nbbn) stored in memory **3**.

At block C the microprocessor **2** operates in conjunction with the memory controller **1** in the above-described manner so as to attempt to retrieve (i.e., 'read') data from the memory location ML1 of the respective DRAMs D1–Dn of DIMM **14***a*. This 'reading' step is assumed to result in data being provided by the DRAMs D1–Dn to the microprocessor **2**. For the purposes of this description, data that is provided to the microprocessor **2** from the respective DRAMs D1–Dn during a 'read' operation is hereinafter

referred to as data (Nb1')–(Nbn'), respectively. For cases in which the data (Nb1)–(Nbn) was previously written successfully (without error) to the respective DRAMs D1–Dn, and was then successfully read back from these DRAMs D1–Dn to the microprocessor **2**, the data (Nb1')–(Nbn') is assumed to be similar (i.e., includes a same bit pattern) to the respective, "written" data (Nb1)–(Nbn), and to the respective data (Nbb1)–(Nbbn) from memory **3**.

After the data (Nb1')–(Nbn') is read by the microprocessor **2** at block B, the microprocessor **2** compares the data (Nb1')–(Nbn') to the data (Nbb1)–(Nbbn) stored in the memory **3** to determine whether or not the retrieved data (Nb1')–(Nbn') is similar to the respective data (Nbb1)–(Nbbn) (block D'). A determination of 'yes' at block D' is assumed to indicate that data was correctly written to, and subsequently correctly read from, the DRAMs D1–Dn at the respective blocks B and C. As an example of a correct or successful 'read' operation, it is assumed that the performance of the step of block B resulted in the data (Nb1)–(Nbn) being correctly written to the memory location ML1 of the DRAMs D1–Dn, and that at block C the microprocessor **2** commands the DRAMs D1–Dn to provide the data from memory location ML1 of the DRAMs D1–Dn to the memory controller **1** in the above-described manner. It is also assumed that the DRAMs D1–Dn respond to receiving the commands by forwarding respective data (Nb1')–(Nbn') to the respective registers **10***a***3***a*–**10***a***3***n*, and that the data (Nb1')–(Nbn') is received at the respective registers **10***a***3***a*–**10***a***3***n* at times which enable the data to be loaded into these registers **10***a***3***a*–**10***a***3***n* within an enablement period (occurring between times T2 and T4) of the registers **10***a***3***a*–**10***a***3***n* (even though different portions of data (Nb1')–(Nbn') may arrive at the registers at different times due to, e.g., variations in lengths of the buses **105***a*–(Na)), as is represented in FIG. **4***a*. As can be appreciated, in this case, the successful loading of the data (Nb1')–(Nbn') into the registers **10***a***3***a*–**10***a***3***n* within the enablement period of these registers enables the data to be successfully forwarded to the microprocessor **2** through respective registers **10***c***1**–**10***cn*.

If 'yes' at block D', then control passes to block E where the microprocessor **2** controls the delay element **11***a* in the manner described above so as to cause the amount of delay provided by the delay element **11***a* to be incremented by the predetermined phase delay amount, thereby causing the local clock signal **11**' applied to delay element **11***a* to be phase delayed accordingly. Also at block E, the microprocessor **2** controls the delay element **11***a* in the manner described above so as to invert the phase delayed local clock signal **11**', and, as a result, the local clock signal **11**' provided to the DRAMs D1–Dn is phase delayed and inverted accordingly. By example, the microprocessor **2** controls the delay element **11***a* at block E by providing binary information, such as '100000000', to the delay element **11***a* through delay element input **11***a*', for causing the delay element **11***a* to phase delay local clock signal **11**' by delay amount V2 (see, e.g., FIG. **2***c*), and causes the phase delayed signal to be inverted by providing a binary '0' to input **88** of multiplexer **88**.

Thereafter, control passes back to block B, where the data (Nb1)–(Nbn) is again written to one of the memory locations ML1–MLn of the respective DRAMs D1–Dn of DIMM **14***a*, in the manner described above, and the steps identified by blocks C and D' are again performed in the above-described manner until it is determined at block D' that data (Nb1')–(Nbn') read back from at least one of the DRAMs D1–Dn of DIMM **14***a* differs from the respective stored data (Nbb1)–(Nbbn) ('no' at block D'). It should be understood that in a case wherein data (Nb1')–(Nbn') read from a DRAM D1–Dn is determined to differ from the respective data (Nb1)–(Nbn) (and the respective stored data (Nbb1)–(Nbbn)) at block D', the technique of the invention assumes that an error must have occurred during the read operation of block C. By example, owing to the performance of the step of block E and, e.g., variations in the lengths of the buses **105***a*–((N)a) through which the data (Nb1')–(Nbn') travels, not all of the data (Nb1')–(Nbn') may arrive at the registers **10***a***3***a*–**10***a***3***n* at times which enable the complete data (Nb1')–(Nbn') to be accepted by the registers **10***a***3***a*–**10***a***3***n* during a same enablement period (EP1) (occurring between times T2–T4) of the registers **10***a***3***a*–**10***a***3***n*. As a result, and some other, incorrect data NN' received by the registers **10***a***3***a*–**10***a***3***n* may be accepted during this enablement period (EP1) instead, as is represented in FIG. **4***b*. It should be further noted that the steps of blocks B, C, D', and E are performed to deliberately cause the phase of the local clock signal **11**' output from the delay element **11***a* to be such that an error occurs during the performance of step D'. This enables a subsequent determination to be made of a first setting of delay element **11***a* which causes data to be transferred between devices **1** and D1–Dn without error, as will be described below.

After it is determined at block D' that the data (Nb1')–(Nbn') read back from one or more of the DRAMs D1–Dn differs from respective data (Nbb1)–(Nbbn) stored in memory **3** ('no' at block D'), control passes to block F where the microprocessor **2** again controls the delay element **11***a* through input **11***a*' so as to cause the delay element **11***a* to be placed in a next delay setting. By example, the microprocessor **2** controls the delay element **11***a* at block E by providing binary information, such as '1100000000', to the delay element **11***a* through delay element input **11***a*', for causing the delay element **11***a* to phase delay the local clock signal **11**' by a total phase delay amount V3 (see, e.g., FIG. **2***c*).

Thereafter, control passes to block G where data (Nb1)–(Nbn) (corresponding to the data (Nbb1)–(Nbbn) from memory **3**) is again written to one of the memory locations ML1–MLn of the respective DRAMs D1–Dn of DIMM **14***a*, in a similar manner as was described above. Then, at block H the microprocessor **2** operates in conjunction with the memory controller **1** in the above-described manner to read data from this memory location of the DRAMs D1–Dn. Assuming that this step results in data (Nb1')–(Nbn') being retrieved from the DRAMs D1–Dn, then at block I' the microprocessor **2** compares the retrieved data (Nb1')–(Nbn') to the data (Nbb1)–(Nbbn) stored in the memory **3** to determine whether or not the data (Nb1')–(Nbn') is similar to the respective data (Nbb1)–(Nbbn). If 'yes' at block I', indicating that the data (Nb1')–(Nbn') was successfully accepted by the respective registers **10***a***3***a*–**10***a***3***n* during a single enablement period of respective registers **10***a***3***a*–**10***a***3***n* (i.e., no error occurred during the 'read' operation performed at block H), then control passes through connector A1 to block N' of FIG. **3***c*, where a further step is performed in a manner which will be described below.

If 'no' at block I', indicating that an error occurred during the 'read' operation performed at block H, then control passes through connector A2 to block J of FIG. **3***b*, where the microprocessor **2** controls one the delay elements **16***a*–**16***n* so as to cause the amount of delay provided by this delay element to be incremented by the predetermined phase delay amount (i.e., so as to increment the delay setting of this delay

element). For the purposes of this description, it is assumed that the delay element controlled by the microprocessor **2** at block J is delay element **16***a*. By example, the microprocessor **2** may control delay element **16***a* at block J by providing binary information, such as '1000000000', to this delay element **16***a* through delay element input **16***a*', for causing the delay element **16***a* to phase delay local clock signal **11**' applied to the register **10***a***3***a* by delay amount **V2**.

Thereafter, at block K the microprocessor **2** operates in conjunction with the memory controller **1** in the above-described manner so as to read data from a memory location, such as memory location ML**1**, of the DRAM D**1**. This step is assumed to result in data (Nb**1**') being provided by the DRAM Dl to the microprocessor **2** through the bus **105***a*, receiver **9***a*, and registers **10***a***3***a* (including flip-flops FF1–FFn) and **10***c***1**.

After the step of block K is performed, and the microprocessor **2** receives the data (Nb**1**') from the register **10***c***1**, the microprocessor **2** compares the received data (Nb**1**') to the data (Nbb**1**) stored in the memory **3** to determine whether or not the compared data is similar. If 'no' at block L, then control passes to block L' where the microprocessor **2** determines whether or not the delay element **16***a* has been incremented through each of its delay settings. This determination may be made in accordance with any suitable technique. By example, each time the microprocessor **2** increments the delay setting of the delay element **16***a* at block J, the microprocessor **2** may increase a value of a counter variable (not shown). In this case, the performance of block L' may include steps of the microprocessor **2** comparing a present value of the counter variable to a predetermined value, such as '10', which indicates the total number of delay settings for the delay element **16***a*. If the value of the counter variable is less than the predetermined value, then the step of block L' results in a determination of 'no'. Otherwise, the step of block L' results in a determination of 'yes'.

A case wherein a determination of 'yes' is made at block L' will now be described. If 'yes' at block L', indicating that data (Nb**1**') has been read from the DRAM D**1** for each delay setting of the delay element **16***a*, then control passes back to block F (FIG. **3***a*) where the microprocessor **2** controls the delay element **11***a* so as to place the delay element **11***a* in a next delay setting, and the method then proceeds in the manner described above.

If the performance of the step of block L' results in a determination of 'no', then control passes back to block J where the microprocessor **2** again controls the delay element **16***a* so as to place the delay element **16***a* in its next delay setting. By example, and assuming that the delay element **16***a* was previously placed in the delay setting corresponding to phase delay amount V**2** represented in FIG. **2***c*, then at block J the microprocessor **2** controls the delay element **16***a* so as to place the delay element **16***a* in the delay setting corresponding to phase delay amount V**3** represented in FIG. **2***c*. Thereafter, the steps of blocks K and L are performed in a similar manner as was previously described. Also, as long as the performance of the steps of block L and L' result in a determination of 'no', then the steps of block J and K are continuously performed in a similar manner as was described above.

As can be appreciated, each time the delay element **16***a* is placed in a next delay setting at block J, for delaying the local clock signal **11**' applied to the register **10***a***3***a*, the time at which the register **10***a***3***a*, and hence, the time at which the flip-flops FF**1**–FF**4** of the register **10***a***3***a*, are triggered is delayed accordingly, until it is eventually determined at block L that data (Nb**1**') read at block K is similar to the data (Nbb**1**) stored in the memory **3** ('yes' at block L). This indicates that the read operation of block K was performed successfully. By example, FIG. **4***c* shows an example of how the time at which the individual flip-flops FF**1**–FFn are triggered is delayed from time T**2** to a time T**2**', owing to the performance of the steps of blocks J, K, L, and L'. In this example, it is assumed that the triggering of the flip-flops FF**1**–FFn at time T**2**' results in bits b**1**–b**4** of the data (Nb**1**') being loaded within a single enablement period EP**1** of the flip-flops F**1**–F**4**, even though not all of the bits may arrive at the flip-flops F**1**–F**4** simultaneously. As a result, the data (Nb**1**') is able to be correctly read from the DRAM D**1**, and the comparing step of block L results in a determination of 'yes'.

A case where the comparing step of block L results in a determination of 'yes' will now be described. If the performance of the step of block L results in a determination of 'yes', it is assumed that the 'write' and 'read' operations previously performed at respective blocks G and K were performed without error. Control then passes to block M**1** where the microprocessor **2** determines whether or not there are additional DRAMs D**1**–Dn within the DIMM **14***a*. This step may be performed in any suitable manner known in the art. By example, between the performance of the steps of blocks L and M**1**, the microprocessor **2** may perform a step of increasing a value of another counter variable (not shown) stored in memory **3**, and may then perform the step of block M**1** by comparing this value to a predetermined value (not shown) stored in memory **3** indicating the total number of DRAMs D**1**–Dn incorporated in the DIMM **14***a*. If the value of the counter variable is determined to be greater than the predetermined value, indicating that there are no additional DRAMs D**1**–Dn included in the DIMM **14***a* ('no' at block M**1**), then control passes through connector A**1** to block N' of FIG. **3***c*, where a further step is performed in a manner as will be described below. If it is determined that there are additional DRAMs D**1**–Dn included in the DIMM **14***a* ('yes' at block M**1**), then control passes to block M**1**', where the microprocessor **2** controls another one the delay elements **16***a*–**16***n* so as place this delay element in its next delay setting. Which one of the delay elements **16***a*–**16***n* is controlled by the microprocessor **2** at block M1' is not of particular importance, as long as it is not the same delay element as the one previously adjusted by the microprocessor **2** at block J. Thereafter, control passes back to block K and a similar procedure is performed for another one of the DRAMs D**1**–Dn in the manner described above.

The performance of the step of block N' of FIG. **3***c* will now be described. As was previously described, for a case in which the performance of the step identified by block I' of FIG. **3***a* results in a determination of 'yes', and for a case in which the performance of the step identified by block M**1** of FIG. **3***b* results in a determination of 'no', control passes to block N' of FIG. **3***c*. Before describing the step performed at block N' in detail, brief reference will first be made to FIG. **5**, which represents a state of the memory subsystem **1**' prior to the step of block N' being entered. In FIG. **5**, blocks DV**1**–DVn (hereinafter referred to as data "valid windows DV**1**–DVn" or "temporal ranges DV**1**–DVn") are shown. Edges Nb**1**"–Nbn" of the respective data valid windows DV**1**–DVn represent earliest times at which the respective registers **10***a***3***a*–**10***a***3***n* may be triggered (by applying a positive clock pulse edge thereto), after the initiation of a 'read' operation by the microprocessor **2**, and it can be expected that data (Nb1')–(Nbn') being read from the respec-

tive DRAMs D1–Dn will be successfully loaded into the respective registers **10***a***3***a*–**10***a***3***n* during the enablement periods of these registers **10***a***3***a*–**10***a***3***n*, for enabling the 'read' operation to be performed without error. Edges Nb**1**'"–Nbn'" of the respective data valid windows DV**1**–DVn represent latest times at which the respective registers **10***a***3***a*–**10***a***3***n* may be triggered, after the initiation of a 'read' operation by the microprocessor **2**, and it can be expected that data (Nb**1**')–(Nbn') being read from the respective DRAMs D1–Dn will be successfully loaded into the respective registers **10***a***3***a*–**10***a***3***n* during the enablement periods of these registers **10***a***3***a*–**10***a***3***n*, for enabling the 'read' operation to be performed without error.

Time (TI**1**) shown in FIG. **5** represents an earliest time, after the initiation of a 'read' operation by the microprocessor **2**, at which data being read from all of the DRAMs D1–Dn can be accurately 'sampled'. That is, time (TI**1**) represents an earliest time, after the initiation of a 'read' operation by the microprocessor **2**, at which all of the registers **10***a***3***a*–**10***a***3***n* may be simultaneously triggered, and it can be expected that data (Nb**1**')–(Nbn') being read from the DRAMs D1–Dn of DIM **14***a* will be successfully loaded into the respective registers **10***a***3***a*–**10***a***3***n* during enablement periods of these registers **10***a***3***a*–**10***a***3***n* (i.e., without error), even though not all portions of the data (Nb**1**')(Nbn') may arrive at the respective registers simultaneously. The time (TI**1**) is also represented by T**2**' of FIG. **4***b*, which represents data (Nb1')–(Nbn') received by the registers **10***a***3***a*–**10***a***3***n*, and being loaded into these registers **10***a***3***a*–**10***a***3***n* within enablement period EP**2** of the registers **10***a***3***a*–**10***a***3***n*. Referring again to FIG. **5**, the line **92** intersecting time (TI**1**) on the time axis is referred to as a "first side", or "early side", of a data valid window (also referred to as a temporal range) DV' for this case. In accordance with the method of this invention, the performance of the procedures appearing prior to block N' results in a detection of the first side **92** of the data valid window DV'. Before block N' is entered, it is assumed that the delay elements **11***a* and **16***a*–**16***n* have delay settings (and the signals output from these delay elements **11***a* and **16***a*–**16***n* have temporal relationships) which enable data to be read from the DRAMs D1–Dn without error. FIG. **5** also shows an exemplary depiction of a signal **90**, which represents the local clock signal **11**' prior to being phase delayed by a respective delay element **16***a*–**16***n*, and a signal **91**, which represents a delayed version of the local clock signal **11**' output from the delay element **16***a*–**16***n* after the procedures appearing prior to block N' are performed. Referring now to FIG. **3***c*, the step performed at block N' will now be described in detail. At block N' the microprocessor **3** stores information similar to that previously provided by the microprocessor **2** to the delay element **11***a* at block F, in the memory **3** as variable FLAG**1**. Also, for each of the following 'read' and 'write' steps, it is assumed that the delay elements **16***a*–**16***n* are maintained in the delay setting in which they were last placed, until they are further adjusted in accordance with this invention as described below.

After block N', control passes to block O where a procedure referred to as a 'data adjust procedure' is commenced. In this procedure, a "second side", or "late side", of the data valid window DV' is "detected", in a manner as will be described below.

At block O the microprocessor **2** controls the memory controller **1** in the manner described above for writing data (Nb**1**)–(Nbn) to a memory location ML1–MLn of a selected DRAM D1–Dn incorporated in DIMM **14***a*. For the purposes of this description, it is assumed that the microprocessor **2** controls the memory controller **1** at block O for writing data (Nb**1**) to a memory location ML1 of DRAM D**1**, through elements **10***b***1**, **10***a***2***a*, and **8***c* of the memory controller **1**, and bus **104***a*. Thereafter, at block P the microprocessor **2** operates in conjunction with the memory controller **1** in the above-described manner so as to retrieve data from the memory location ML**1** of the DRAM D**1**. This step is assumed to result in data (Nb**1**') being provided by the DRAM D**1** to the microprocessor **2** through the bus **105***a*, receiver **9***a*, and registers **10***a***3***a* and **10***c***1**.

After the step of block P is performed, and the microprocessor **2** receives the data (Nb**1**') from the register **10***c***1**, the microprocessor **2** compares the received data (Nb**1**') to the data (Nbb**1**) stored in the memory **3** to determine whether or not the compared data is similar (block Q). A determination of 'yes' at block Q indicates that the 'write' and 'read' operations performed at blocks O and P were successfully performed without error. An example of a successful 'write' operation is represented in FIG. **4***d*, where bits b**1**–b**4** of data (Nb**1**) are received by the DRAM D**1** at times which enable the bits b1–b**4** to be accepted by the DRAM D**1** within an enablement period EP**1** of the DRAM D**1**.

After a determination of 'yes' is made at block Q, control passes to block R where the microprocessor **2** controls the delay element **15***a* in the manner described above, so as to increment the delay setting of the delay element **15***a*. Then, the steps of blocks O, P, Q, and R are performed in a similar manner as described above. As long as the performance of the step of block Q results in a determination of 'yes', then the steps of block R, O, and P are continuously performed in the above-described manner. As can be appreciated, each time the delay element **15***a* is placed in a next delay setting at block R for delaying the local clock signal **11**' applied to the register **10***a***2***a*, the time at which the register **10***a***2***a*, and hence, the times at which the flip-flops FF**1**–FF**4** of register **10***a***2***a*, are triggered, are delayed accordingly (i.e., are temporally displaced). As such, the times at which data received at input **10***a***2***a*" of the register **10***a***2***a* is loaded into, and subsequently forwarded to DRAM D**1**, by the register **10***a***2***a*, are also delayed accordingly. The continuous performance of the steps of blocks R, O, P, and Q eventually results in at least some portion of the data (Nb**1**) not being accepted by DRAM D**1** within a same enablement period EP**1** of DRAM D1 as other portions of the data (Nb1). An example of times at which bits b1'–b**4**' of the data (Nb**1**) are received by DRAM D**1** relative to the time T**2** at which DRAM D**1** is triggered in response to receiving a positive edge of a clock pulse over bus **103***a*, and an example of the enablement period EP**1** of the DRAM D**1** for this case, is shown in FIG. **4***d*. These bits b**1**'–b**4**' are shown as being temporally displaced relative to bits b**1**–b**4**, owing to the delayed triggering of register **10***a***2***a*. As can be seen in FIG. **4***d*, bit b**2**' of bits b**1**'–b**4**' is received by DRAM D**1** after the occurrence of the enablement period EP**1** of the DRAM D**1**, and another arbitrary bit bn' is received within the enablement period EP1. As a result, the performance of the 'read' step of block P results in the bits b**1**', b**3**', b**4**', as well as arbitrary bit bn', being collectively loaded in DRAM D**1** (rather than bits b**1**'–b**4**'), and being subsequently provided to the microprocessor **2** during a 'read' operation of block P. In this case, the subsequent performance of the step of block Q results in a determination of 'no'.

It should be noted that the steps of blocks R, O, P, and Q are performed so as to deliberately cause the delay element **15***a* to temporally displace the local clock signal applied to the delay element **15***a* by an amount of temporal displacement which results in a determination of 'no' at block Q (i.e.,

which results in an occurrence of a write error). In this manner, a subsequent determination can be made of a delay setting of the delay element **15***a* which causes data to be written to the DRAM D1 without error, as will be described below.

After a determination of 'no' is made at block Q, control passes to block S where the microprocessor **2** provides information to the delay element **15***a* in the above described manner so as to cause the amount of phase delay provided by delay element **15***a* to be decremented by the predetermined phase delay amount, thereby reducing the delay setting of the delay element **15***a*. Also at block S, the microprocessor **2** stores the information in the memory **3** as variable FLAG**15***a*, and it is assumed that the delay element **15***a* is maintained in the new delay setting until sometime later when it is further adjusted (as will be described below). As a result of the step of block S, if a further operation were to be performed to write data to DRAM D1 via register **10***a***2***a*, the register **10***a***2***a* would be triggered at a time which would enable bits b1–b**4** of the data (Nb**1**) to be received by the DRAM D1 at times which would enable the bits b**1**–b**4** to be successfully loaded into DRAM D1 within the enablement period EP1 of DRAM D1. This "trigger" time is considered to be a latest time at which register **10***a***2***a* can be triggered, and it can be assured that the complete data (Nb**1**) will be successfully accepted by the DRAM D1 during an occurrence of enablement period EP1 of DRAM D1 (i.e., without error), for storage therein.

After block S, control passes to block T' where it is determined whether or not there are other DRAMs D1–Dn on the DIMM **14***a* (besides DRAM D1) for which the data adjust procedure needs to be performed. This step may be performed in accordance with any suitable technique (such as one employing a counter variable, as described above).

If 'yes' at block T', the control passes to block T1' where the microprocessor **2** controls the memory controller **1** in the manner described above so as to write data (Nb**1**)–(Nbn) to a memory location ML1–MLn of another selected one of the DRAMs D1–Dn incorporated in the DIMM **14***a*. By example, the microprocessor **2** may control the memory controller **1** at block T1' for writing data (Nbn) to a memory location ML**1** of DRAM Dn, through the various elements **10***bn*, **10***a***2***n*, **8***n*, and (N−1)a. Control then passes back to block P, where data (Nbn) is read from this DRAM Dn, and the method proceeds in the manner described above.

The data adjust procedure steps identified by blocks P, Q, R, S, T', and T1' are performed in the above-described manner so as to adjust the setting of each of the remaining delay elements **15***b*–**15***n*, and to store information corresponding to these delay settings in the memory **3** as the respective variables FLAG**15***b*–FLAG**15***n*. As a result of the performance of the data adjust procedure described above, optimum delay settings for all of the delay elements **15***a*–**15***n* are determined, so that in subsequently performed 'write' operations, these delay settings may be employed to enable the registers **15***a*–**15***n* to be triggered at times which enable the data (Nb**1**)–(Nbn) to be written to the DRAMs D1–Dn without error (i.e., to enable the registers **15***a*–**15***n* to be triggered at times which cause the data (Nb**1**)–(Nbn) to be received by the DRAMs D1–Dn at times which enable the data to be successfully accepted by the DRAMs D1–Dn within enablement periods of the DRAMs D1–Dn). At this point in the procedure, the settings of the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* are such that, if a further read operation were to be performed, the registers **10***a***3***a*–**10***a***3***n* would be triggered at an earliest time which would enable data (Nb**1**')–(Nbn') being read from the DRAMs D1–Dn of DIM **14***a* to be accepted at the respective registers **10***a***3***a*–**10***a***3***n* during enablement periods of these registers **10***a***3***a*–**10***a***3***n* (i.e., which would enable the data to be read without error). This time is represented by time (TI**1**) shown in FIG. **6**. That is, time (TI**1**) represents an earliest time at which all of the registers **10***a***3***a*–**10***a***3***n* may be triggered, and it can be expected that data being read from the DRAMs D1–Dn will be read without error. An exemplary relationship between a positive edge **92**' of clock signal **11**' applied to the respective registers **10***a***3***a*–**10***a***3***n* for such an operation is also represented in FIG. **6**.

After the data adjust procedure has been performed for all DRAMs D1–Dn of the DIMM **14***a* ('no' at block T'), control passes through connector A**3** to block U' of FIG. **3***d*, where a 'final local clock phase adjust' procedure is initiated. This procedure is represented by blocks U'-X**3**, collectively, of FIGS. **3***d* and **3***e*. At block U' the microprocessor **2** controls the delay element **11***a* in the above-described manner so as to increment the delay setting of the delay element **11***a*, thereby causing a resultant phase delay of the local clock signal **11**' forwarded to the DRAMs D1–Dn of DIMM **14***a*. Then, at blocks U" and V the microprocessor **2** and memory controller **1** operate in conjunction with one another to write data (Nb**1**)–(Nbn) to the respective DRAMs D1–Dn of DIMM **14***a*, and to read data (Nb**1**')–(Nbn') back from these DRAMs D1–Dn. Then, at block W the data (Nb**1**')–(Nbn') is compared to the data (Nbb**1**)–(Nbbn) from the memory **3** to determine whether or not the respective, compared data is similar. If 'yes' at block W, then control passes back to block U' where the microprocessor **2** again increments the delay setting of the delay element **11***a*. Thereafter, the steps indicated by blocks U", V, and W are again performed until the comparing step of block W results in a determination that the data (Nb**1**')–(Nbn') read back from at least one of the respective DRAMs D1–Dn at block V is not similar to the respective data (Nbb**1**)–(Nbbn) from the memory **3** ('no' at block W). It should be noted that the steps of blocks U', U", V, and W, are performed to deliberately cause the phase of the local clock signal **11**' output from the delay element **11***a* to be such that an error occurs during the performance of step W. This enables a subsequent determination to be made of a second setting of delay element **11***a* which causes data to be transferred between devices 1 and D1–Dn without error, as will be described below.

A determination of 'no' at block W indicates that the phase of local clock signal **11**' is such that, after the DRAMs D1–Dn are commanded by the microprocessor **2** to provide data to the microprocessor **2** during a 'read' operation, and the DRAMs D1–Dn subsequently respond to receiving this command and respective positive edges of local clock signal **11**' by forwarding respective data (Nb**1**')–(Nbn') to the memory controller **1**, at least some portion of the forwarded data (Nb**1**')–(Nbn') does not reach a respective register **10***a***3***a*–**10***a***3***n* prior to an elapse of an enablement period EP1 of this register, while other portions of the data (Nb**1**')–(Nbn') are received by the respective register prior to the elapse of enablement period EP1. This is depicted in FIG. **4***b*, for example, wherein the delaying of the local clock signal **11**' at block U' ultimately results in data Nb**2**' and Nbn' being received at respective registers **10***a***3***b* and **10***a***3***n* after an occurrence of an enablement period EP1 of these registers **10***a***3***b* and **10***a***3***n*.

If 'no' at block W, then control passes to block X' where the microprocessor **2** again controls one the delay elements **16***a*–**16***n* so as to increment the delay setting of this delay element. For the purposes of this description, it is assumed that the delay element controlled by the microprocessor **2** at block X' is delay element **16***a*.

Thereafter, at block Y' the microprocessor **2** operates in conjunction with memory controller **1** in the above-described manner so as to read data (Nb**1**') from memory location ML**1** of the DRAM D**1**. Assuming that the step of block Y' results in data (Nb**1**') being retrieved from the DRAM D**1**, then at block Z' the microprocessor **2** compares the retrieved data (Nb**1**') to the data (Nbb**1**) stored in the memory **3** to determine whether or not the compared data is similar.

A determination of 'yes' at block Z' indicates that, as a result of the step of block X', which caused the triggering of an enablement period of register **10***a***3***a* to be delayed, all of the bits b**1**–b**4** of the data (Nb**1**') forwarded to the register **10***a***3***a* were received at register **10***a***3***a* at times which enabled the bits b**1**–b**4** to be loaded into the register **10***a***3***a* during delayed enablement period EP**1**' of the register **10***a***3***a*, as is represented in FIG. **4***c*. If 'yes' at block Z', then control passes to block Z**2**' where the microprocessor **2** determines whether or not there are additional DRAMs D**1**–Dn on the DIMM **14***a* for which the steps of blocks X'–Z**1**' need to be performed. This step may be performed in accordance with any suitable technique known in the art, such as one employing a counter variable, as described above.

If 'yes' at block Z**2**', then control passes to block Z**3**' where the microprocessor **2** increments the delay setting of another, selected one of the delay elements **16***a*–**16***n* (besides the one adjusted previously at block X'). Thereafter, control passes back to block Y' where the method proceeds in the above-described manner. If 'no' at block Z**2**', then control passes back up to block U' (FIG. **3***d*) where the method continues in the above-described manner.

A case where a determination of 'no' is made at block Z' will now be described. A determination of 'no' at block Z' indicates that, after the DRAM D**1** forwarded data (Nb**1**') along the bus **105***a* in response to receiving a positive edge **93**' (FIG. **6**) of local clock signal **11**' during the step of block Y', at least some portion of the data (Nb**1**'), such as a bit b**2** of the data (Nb**1**'), was not received at the register **10***a***3***a* at a time which enabled the bit b**2** to be loaded into the register **10***a***3***a* during an enablement period (EP**1**') of the register **10***a***3***a*, as is represented in FIG. **4***h*. In this case, it can be said that the phase of the local clock signal **11**" output from delay element **11***a* is such that positive edge **93**' of the signal **11**" is no longer "within" an extent of the data valid windows DV**1**–DVn, as is represented in FIG. **6**. That is, the phase of the local clock signal **11**" output from delay element **11***a* is such that data is not able to be read from the DRAM D**1** without error.

After a determination of 'no' is made at block Z', control passes to block Z**1**' where the microprocessor **2** determines whether or not the delay element **16***a* has been adjusted through each of its delay settings since the step of block X' was first entered. If 'no' at block Z**1**', then control passes back to block X' where the microprocessor **2** increments the delay setting of delay element **16***a*, and the method then proceeds in the above-described manner. If 'yes' at block Z**1**', then control passes through connector A**4** to block X" of FIG. **3***e*, where a further step is performed.

The step performed at block X' of FIG. **3***e* will now be described. At block X" the microprocessor **2** provides information to the delay element **11***a* (through delay element input **11***a*') for causing the delay setting of the delay element **11***a* to be reduced to a next, lesser delay setting. Then, the microprocessor **2** stores this information in the memory **3** as variable FLAG**2** (block X**1**). The performance of the step of block X" results in the delayed local clock signal **11**' output from delay element **11***a* having a phase such that, if a further read operation were to be performed in the above-described manner, data (Nb**1**')–(Nbn') forwarded from DRAMs D**1**–Dn would be successfully accepted by the respective registers **10***a***3***a*–**10***a***3***a* during the enablement periods of the registers **10***a***3***a*–**10***a***3***n*, thereby enabling the 'read' operation to be performed without error. As such, it can be said that a second side, or "late side" **93**, of the data valid window DV', is detected. FIG. **7** shows an exemplary relationship between a positive edge **93**' of the signal **11**' output from delay element **11***a*, relative to the second side **93** of data valid window DV'. The second side **93** of the data valid window DV' represents a latest time (during a read operation) at which all of the registers **10***a***3***a*–**10***a***3***n* may be triggered, and it can be expected that data being read from the DRAMs D**1**–Dn will be read without error.

After the step of block X**1** is performed, control passes to block X**2** where the microprocessor **2** retrieves the information stored as the variables FLAG**1** and FLAG**2**. After retrieving the information from these variables, the microprocessor **2** correlates this information to the corresponding information stored in data table T**1**, and then retrieves the phase delay setting values associated with this information from data table T**1**. Thereafter, microprocessor **2** performs an algorithm for determining an average of the retrieved phase delay setting values. By example, it is assumed that the information stored as variable FLAG**1** indicates '1000000000' and that the information stored as variable FLAG**2** indicates '1111111000'. In this case, after retrieving the information '1000000000' and '1111111000' from the respective FLAG**1** and FLAG**2** variables, the microprocessor **2** correlates the retrieved information to the corresponding command information in data table T**1** specifying '1000000000' and '1111111000'. The microprocessor **2** then retrieves corresponding phase delay setting values V**2** (e.g., V**2**=(2T/10)) and V**8** (e.g., V**8**=(8T/10)) from the data table T**1**, and performs an algorithm for determining an average value (AV) of the retrieved delay setting values V**2** and V**8**. In this example, the algorithm may be in accordance with the following equation (EQ1):

$$(AV)=((V2)+(V8))/2 \quad \text{(EQ1)}$$

Referring to FIG. **2***d*, it can be appreciated that the performance of the algorithm defined by equation (EQ1) results in a determination that the average value (AV) equals phase delay value V**5** which, in this example, corresponds to command information specifying '1111000000' stored in the data table T**1**.

After the step of block X**2** is performed, control passes to block X**3** where the microprocessor **2** retrieves from the data table T**1** the command information corresponding to the average delay setting value determined in block X**2**, and loads this information into variable FLAG**3**. By example, and assuming that the average value (AV) calculated in block X**2** equals phase delay value V**5**, then at block X**2** the microprocessor **2** retrieves the command information specifying '1111000000' from data table T**1**, and then loads this information into variable FLAG**3**. Also by example, and assuming that the average value (AV) calculated in block X**2** equals some value that is between V**5** and V**6** (e.g., a value which is an average of phase delay values V**1** and V**11**), then the microprocessor **2** may retrieve the command information specifying '1111000000', which immediately precedes the information '1111000000' in data table T**1**, or, alternatively, the next command information (specifying '1111100000') appearing after the information '1111000000' in the data table T**1**, for storing this information as variable FLAG**3**.

Also at block X3, the microprocessor 2 provides the retrieved command information to the input 11*a*' of delay element 11*a* so as to place the delay element 11*a* in a delay setting corresponding to the information. As a result, the signal (identified by 11") output from delay element 11*a* is phase delayed relative to the signal 11' originally applied to delay element 11*a*. By example, assuming that the information provided to the delay element 11*a* specifies '1111000000', then the performance of the step of block X3 results in the delay element 11*a* phase delaying the signal 11" by an amount of delay equivalent to value V5.

The performance of the step of block X3 results in the signal 11" output from delay element 11*a*, and subsequently applied to the DRAMs D1–Dn, having a phase such that, if a further 'write' operation were to be performed in the above-described manner, data (Nb1)–(Nbn) written from the memory controller 1 would eventually be received by the respective DRAMs D1–Dn at times that would enable the data (Nb1)–(Nbn) to be successfully loaded into the DRAMs D1–Dn within an enablement period EP1 of the collective DRAMs D1–Dn (the enablement period of the DRAMs D1–Dn occurring in response to respective positive edges of the signal 11' being applied to the respective DRAMs D1-Dn). This is depicted in FIG. 4*e*, and the phase delay setting of the delay element 11*a* after block X3 is considered to be an optimum phase setting for the delay element 11*a*. With this phase setting of the delay element 11*a*, it can be said that the phase of signal 11" output therefrom is such that a positive edge 93' of the signal is aligned with a "center" 95 of the data valid window DV', as represented in FIG. 7. In this stage of the procedure, as long as the signal 11" is not temporally displaced by an amount which extends beyond "edges" of a temporal range defined by lines 92 and 93 in FIG. 7, 'read' and 'write' operations can be performed within the memory subsystem 1' without error.

Referring again to FIG. 3*e*, after the step of block X3 is performed, control passes to block X4. At block X4, the microprocessor 2 determines whether or not there are additional DIMMs coupled to the memory controller 1. This step may be performed in accordance with any suitable technique known in the art, such as one employing a counter variable, as described above.

If 'yes' at block X4, then control passes to block X5 where the microprocessor 2 controls the memory controller 1 in the manner described above so as to write data to a memory location ML1–MLn of all DRAMs D1–Dn incorporated in a next, selected one of the DIMMs 14*n*—14*n*, besides DIMM 14*a*. Thereafter, control passes back to block C of FIG. 3*a* where the method proceeds in the above described manner for the next, selected DIMM 14*a*–14*n*.

If 'no' at block X4, indicating that the procedures described above have been performed for all of the DIMMs 14*a*–14*n* within the subsystem 1', then control passes to block AA where a procedure is initiated for optimizing the performance of 'read' operations performed within the subsystem 1'.

At block AA the microprocessor 2 operates in conjunction with the memory controller 1 in the above-described manner so as to read data (Nb1')–(Nbn') from a memory location ML1–MLn of a DRAM D1–Dn from a particular DIMM 14*a*–14*n*. For the purposes of this description, it is assumed that the 'reading' step of block AA is performed so that data (Nb1') from memory location ML1 of DRAM D1 from DIMM 14*a*, is read by the microprocessor 2. Thereafter, the microprocessor 2 compares the data (Nb1') to the data (Nbb1) stored in the memory 3 to determine whether or not the compared data is similar (block BB). If 'yes' at block BB, then control passes to block CC where the microprocessor 2 controls the delay element 16*a* in the manner described above so as to place the delay element 16*a* in its next delay setting, for phase delaying the signal 11' applied to the delay element 16*a*, and for causing this signal 11 to be inverted. Then, control passes back to block AA, where the method proceeds in the above-described manner. The steps of blocks AA, BB, and CC are performed continuously until it is determined at block BB that the data (Nb1') read back from the DRAM D1 differs from the stored data (Nbb1) ('no' at block BB), indicating that the data (Nb1') was not successfully read from DRAM D1. By example, an indication of 'no' at block BB may indicate that, as a result of the step of block CC, at least some bits b1–b4 of the data (Nb1') were not completely accepted by the register 10*a*3*a* at block AA, during an enablement EP1 period of the register, as represented in FIG. 4c. It should be noted that the steps of blocks AA, BB, and CC are performed to deliberately cause the phase of the signal output from the delay element 16*a* to be such that the step of block BB results in a determination of 'no' (i.e., such that a read error occurs). This enables a subsequent determination to be made of a first setting of delay element 16*a* which causes data to be read from DRAM D1 without error, as will be described below.

After a determination of 'no' is made at block BB, control passes to block DD where the microprocessor 2 provides information to the delay element 16*a* in the manner described above so as to cause the delay element 16*a* to be placed in its next delay setting. Then, at block EE the microprocessor 2 again operates in conjunction with the memory controller 1 so as to 'read' data (Nb1') from memory location ML1 of DRAM D1. The microprocessor 2 then compares the data (Nb1') to the data (Nbb1) stored in the memory 3 to determine whether or not the compared data (Nb1') and (Nbb1) is similar (block FF).

If 'no' at block FF, then control passes back to block DD, and the steps of blocks DD, EE, and FF are again performed until it is determined at block FF that the data (Nb1') read back from the DRAM D1 is similar to the stored data (Nbb1) ('yes' at block FF). By example, owing to step DD, which delays the time at which the register 10*a*3*a* is triggered (i.e., which temporally displaces an occurrence of the enablement period of the register 10*a*3*a*), it eventually occurs that all of the bits b1–b4 of data (Nb1') read at block EE are accepted by the register 10*a*3*a* during a "delayed" enablement EP1' period of the register 10*a*3*a*, as represented in FIG. 4c. A determination of 'yes' at block FF indicates that a determination has been made of an earliest time at which the register 10*a*3*a* may be triggered for being enabled (during a read operation initiated at block EE), and it can be expected that data (Nb1') being read from DRAM D1 will be received at register 10*a*3*a* at a time which would enable the data (Nb1') to be successfully loaded into the register 10*a*3*a* during the period of enablement of the register 10*a*3*a* (i.e., it can be expected that the data (Nb1') will be read without error). The "earliest time" at which the register 10*a*3*a* may be triggered in this case is represented by a first side, or "early side", 96' of a data valid window DV1 in FIG. 4*f*, the data valid window DV1 of FIG. 4*f* representing a temporal "window" defining an extent of time within which the register 10*a*3*a* may be triggered, during a read operation initiated at block EE, and it can be expected that data (Nb1') being read from DRAM D1 will be read without error. FIG. 4*f* also shows an exemplary relationship between a positive edge 99 of the signal output from delay element 16*a* to register 10*a*3*a*, relative to first side 96' of data valid window DV1.

Referring again to FIG. 3*e*, a case where a determination of 'yes' is made at block FF will now be described. If 'yes'

at block FF, then control passes through connector A5 to block GG of FIG. 3*f*, where the microprocessor **2** stores the information previously provided to the delay element **16***a* at block DD in the memory **3** as variable FLAG**16***a***1**. Control then passes to blocks HH and II where reading and comparing steps similar to those of blocks EE and FF described above are performed.

If the performance of block II results in a determination that data (Nb**1**') read from DRAM D**1** is similar to stored data (Nbb**1**) ('yes' at block II), then control passes to block JJ where the microprocessor **2** places the delay element **16***a* in its next phase delay setting. Thereafter, control passes to block HH. The steps of block HH, II, and JJ are performed until it is determined that the data (Nb**1**') read from the DRAM D**1** differs from the data (Nbb**1**) stored in memory **3** ('no' at block II). By example, owing to step JJ, which delays the time at which the register **10***a***3***a* is triggered (i.e., which delays an occurrence of the enablement period of the register **10***a***3***a*), it eventually occurs that at least some of the bits b**1**–b**4** of data (Nb**1**') read at block HH, such as bit b**1**, are not accepted by the register during a "delayed" enablement period EP**1**" of the register, since the register **10***a***3***a* is triggered (in response to receiving a positive edge of a clock pulse) too "late" for enabling bit b**1** to be accepted by register **10***a***3***a* during period EP**1**", as represented in FIG. **4***c*.

After a determination has been made that the data (Nb**1**') read from the DRAM D**1** differs from the data (Nbb**1**) stored in memory **3** ('no' at block II), control passes to block KK where the microprocessor **2** provides information to the delay element **16***a* to cause the delay element **16***a* to be placed in a next, lesser delay setting. This step results in the delayed local clock signal **11**' output from delay element **16***a* having a phase such that, if a further read operation were to be performed so as to retrieve data (Nb**1**'), all bits b**1**–b**4** of the data (Nb**1**') would be received by register **10***a***3***a* at times which would allow the bits b**1**–b**4** to be successfully loaded into the register **10***a***3***a* within an enablement period of the register **10***a***3***a*, such as enablement period EP**1**" (FIG. **4***c*). Also, the phase setting of delay element **16***a* is such that, if a further read operation were to be performed, the signal output from the delay element **16***a* to register **10***a***3***a* would cause the delay element **16***a* to be triggered at a latest possible time that allows for the successful performance of a read operation (i.e., at a latest time that allows the read operation to be performed without error). The "latest time" at which the register **10***a***3***a* may be triggered in this case is represented by a "late side" **97**' of the data valid window DV**1** in FIG. **4***f*. FIG. **4***f* also shows an exemplary relationship between a positive edge **98** of the signal output from delay element **16***a* to register **10***a***3***a*, relative to late side **97**' of data valid window DV**1**.

The microprocessor **2** then stores the information provided to the delay element **16***a* at block KK in the memory **3** as variable FLAG**16***a***2** (block LL). Control then passes to block MM where the microprocessor **2** retrieves the information stored as the variables FLAG**16***a***1** and FLAG**16***a***2**. After retrieving the information from these variables, the microprocessor **2** correlates this information to the corresponding information stored in data table T**1**, and then retrieves the phase delay setting values associated with this information from data table T1. Thereafter, microprocessor **2** performs an algorithm to determine an average of the retrieved phase delay setting values (block MM) in the above described manner. By example, assuming that the information retrieved from variables FLAG**16***a***1** and FLAG**16***a***2** indicate '1000000000' and '1111111000', respectively, and that the microprocessor **2** correlates the retrieved information to the corresponding command information and associated delay setting values from the data table T**1**, then the performance of the algorithm results in a determination that the average of the delay setting values equals delay value V**5**, which corresponds to command information specifying '1111000000' stored in the data table T**1**.

After the step of block MM is performed, control passes to block NN where the microprocessor **2** retrieves from the data table T1 the command information (e.g., '1111000000') corresponding to the average delay setting value determined in block MM. Also at block NN, the microprocessor **2** stores the retrieved information in the memory **3** as variable FLAG**16***a***3**, and also places delay element **16***a* in a setting corresponding to the retrieved information (for subsequently performed 'read' operations). By example, the microprocessor **2** may provide information such as '1111000000' to the delay element **16***a* for placing the delay element in a delay setting corresponding to phase delay value V**5**.

The delay setting in which the delay element **16***a* is placed at block NN is considered to be an optimum delay setting for the delay element **16***a*. In subsequently performed 'read' operations, this delay setting causes the signal output from delay element **16***a* to trigger (enable) register **10***a***3***a* at a time which results in all bits of the data (Nb**1**') being 'read' from DRAM D**1** being successfully loaded into the register **10***a***3***a* within a duration of a same enablement period of the register **10***a***3***a*. This is represented in FIG. **4***g*. With this phase setting of the delay element **16***a*, it can be said that the phase of the signal output from delay element **16***a* is such that a positive edge **99**' of the signal is aligned with a "center" **95**' of the data valid window DV**1**, as represented in FIG. **4***f*. As long as the phase of this signal is not temporally displaced by an amount that extends beyond "edges" of a temporal range defined by lines **96**' and **97**' in FIG. **4***f*, 'read' operations can be performed within the memory subsystem **1**' without error. After block NN is performed, control passes to block OO where the microprocessor **2** determines whether or not there are additional DRAMs D1–Dn on the DIMM **14***a* for which the steps of blocks AA–NN need to be performed, in a manner as was described above. If 'yes' at block OO, then a next DRAM D1–Dn of DIMM **14***a* is selected (block PP), and control passes back to block AA where the above-described steps AANN are performed for this selected DRAM D1–Dn.

If 'no' at block **00**, it is assumed that all of the delay elements **16***a*–**16***n* have been placed in their optimum settings, in the manner described above. Then control passes to block QQ where the microprocessor **2** determines whether or not there are additional DIMMs **14***a*–**14***n* for which the steps of blocks AA-NN need to be performed for the DRAMs D1–Dn of these DIMMs **14***a*–**14***n*, in a similar manner as was described above. If 'yes' at block QQ, then it is assumed that microprocessor **2** selects a next one of the DIMMs **14***a*–**14***n* (block RR), and control passes back to block B of FIG. **3***a* where the procedures of steps B–NN are performed in the above-described manner for DRAMs D1–Dn of the selected DIMM **14***a*–**14***n*.

Assuming that a determination of 'no' is made by the microprocessor **2** at block QQ, then the procedure of the invention is terminated (block SS), and it is assumed that the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* are all placed in settings which optimize the performance of the memory control system 1'.

In accordance with this invention, the method set forth in FIGS. **3***a*–**3***f* is performed during the initial 'start-up' of the

computer system **1**", and lasts no more than 50 ns. After the system **1**" is 'powered-up', and during 'write' and 'read' operations required to be performed during subsequent computer system **1**" operations, the microprocessor **2** controls the various delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* using the information stored as the various variables FLAG**3**, FLAG**15***a*–FLAG**15***n*, and FLAG**16***a***3**–FLAG**16***n***3**, for causing these delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* to be placed in their 'optimum' settings. By example, it is assumed that an operation being performed by the microprocessor **2** of the computer system **1**" requires that data be written to a memory location ML**1**–MLn of the DRAMs D**1**–Dn of each DIMM **14***a*–**14***n*. In this case, the microprocessor **2** retrieves the information stored as the various variables FLAG**3** and FLAG**15***a*–FLAG**15***n*, and provides this information to the various, corresponding delay elements **11***a*, **15***a*–**15***n* of each block **110***a*–**110***n*, so as to cause these respective delay elements **11***a* and **15***a*–**15***n* to be placed in the delay settings corresponding to the provided information. The microprocessor **2** also operates in conjunction with the memory controller **1** in the above-described manner so as to write the data (Nb**1**)–(Nbn) to the memory location ML**1**–MLn of the DRAMs D**1**–Dn of each DIMM **14***a*–**14***n*. As can be appreciated, because the delay elements **11***a* and **15***a*–**15***n* of each block **110***a*–**110***n* are placed in their 'optimum' settings, data (Nb**1**)–(Nbn) forwarded from the memory controller to DRAMs D**1**–Dn of each DIMM **14***a*–**14***n* is received by the DRAMs D**1**–Dn at times which enable the data (Nb**1**)–(Nbn) to be successfully loaded (i.e., loaded without error) into the respective DRAMs D**1**–Dn within an enablement period of the DRAM D**1**–Dn.

A similar operation is performed for 'read' operations, except that in these cases, the microprocessor **2** provides the information from variables FLAG**3** and FLAG**16***a***3**–FLAG**16***an* to the various delay elements **11***a*, **16***a*–**16***n* of each block **110***a*–**110***n*, so as to cause these respective delay elements **11***a* and **16***a*–**16***n* to be placed in their corresponding, 'optimum' settings. During such 'read' operations, the microprocessor **2** operates in conjunction with the memory controller **1** in the above-described manner so as to read data from DRAMs D**1**–Dn of one or more DIMMs **14***a*–**14***n*. As can be appreciated, because the delay elements **11***a* and **16***a*–**16***n* of each block **110***a*–**110***n* are placed in their 'optimum' settings, data (Nb**1**)–(Nbn) forwarded to the memory controller **1** from the DRAMs D**1**–Dn of each DIMM **14***a*–**14***n* is received by the registers **10***a***3***a*–**10***a***3***n* of each block **110***a*–**110***n* at times which enable the data (Nb**1**)–(Nbn) to be successfully loaded into the registers **10***a***3***a* within an enablement period of the registers **10***a***3***a*–**10***a***3***n*, for being subsequently forwarded to the microprocessor **2** without error.

Being that the method of the invention places the delay elements **11***a*, **15***a*–**15***n*, and **16***a*–**16***n* in optimum settings in the above-described manner, the method compensates for any differences in times at which portions of data being transferred between the memory controller **1** and the DRAMs D**1**–Dn of the various DIMMs **14***a*–**14***n*, arrive at the respective components **1** and D**1**–Dn. As a result, 'read' and 'write' operations are performed within the subsystem **1**' without error, and the overall processing speed of the memory control system **1**' (and of the computer system **1**" in general) is increased relative to that of the prior art memory control systems described above.

It should be noted that, after the method of the invention is performed for a first time the system **1**" is powered up, the information stored as the variables FLAG**3**, FLAG**15***a*–FLAG**15***n*, and FLAG**16***a***3**–FLAG**16***an* in accordance with the invention may be stored permanently in the memory **3** (or registers within device **1**). Also, for this case it is within the scope of the invention to subsequently employ this information to optimize the temporal relationships between the clock signals employed for reading and writing data, for subsequent cases in which the system **1**" is powered up.

It should also be noted that in one embodiment of the invention, the phase of the clock signal applied to register **10***a***1** through input **10***a***1**' is the same as the phase of the clock signal provided to the DIMM **14***a* over bus **103***a*, owing to the fact that the input **10***a***1**' and bus **103***a* are both connected to the output of delay element **11***a*. This feature ensures that data **10**' output by register **10***a***1** through bus **102***a* is eventually received at the DRAMs D**1**–Dn at a time which enables the data **10**' to be loaded into the DRAMs D**1**–Dn during an enablement period of the respective DRAMs D**1**–Dn (i.e., the data **10**' is received at the DRAMs D**1**–Dn simultaneously with a positive edge of the clock signal **11**"). It should also be noted that, in view of the above description, one skilled in the art would appreciate that a technique similar to that described above for optimizing the times at which data is written from the registers **10***a***2***a*–**10***a***2***n* to the DRAMs D**1**–Dn (i.e., the technique for optimizing the settings of the delay elements **15***a*–**15***n*), may also be employed to optimize the times at which data **10**' is provided to the DRAMs D**1**–Dn.

It should further be noted that in one embodiment of the invention, the memory **3** and data table T**1** may be incorporated within the memory controller **1**, such as within registers of the memory controller **1**. This embodiment enables information stored in the devices **3** and T**1** to be readily accessed when needed, and minimizes data retrieval latency.

Moreover, it should be noted that, although the invention is described in the context of employing DIMMs and DRAMs, other types of memory modules and memory storage devices may also be employed, such as Synchronous DRAM Double Data Rate (SDRAM-DDR) memory modules and memory storage devices, or Synchronous DRAM (SDRAM) memory modules and memory storage devices.

Also, although the invention is herein described in the context of employing data nibbles, other suitable types of information (e.g., binary information other than nibbles) may also be employed. Moreover, the data nibbles (Nb**1**)–(Nbn) provided to the separate DRAMs D**1**–Dn of DIMM **14***a* during write operations may be similar to one another, or different from one another, depending on applicable performance criteria.

It is also noted that in cases wherein SDRAM-DDR memory modules and memory storage devices are employed, typical DDR mode operations are such that the data **10**' is provided to the DRAMs D**1**–Dn over bus **102***a* at half the frequency (i.e., the local clock signal frequency) as that of data provided over, for example, the buses **104***a*–(N–1)a, and **105***a*–Na. For the purposes of this invention, however, this frequency differential is not critical, and the data provided over the respective buses **102***a*, **104***a*–(N–1)a, and **105***a*–Na, may be provided using any clock signal frequency. It is further emphasized that the above-described procedures can be performed regardless of the particular memory configuration (e.g., board layout) employed in the system **1**", and regardless of whether or not the DRAMs D**1**–Dn and/or DIMMs **14***a*–**14***n* are manufactured by different manufacturers. In addition to overcoming latency resulting from bus length variations, the technique of the

invention may also be performed to overcome latency resulting from other factors that may be present within the system **1'**, such as latency which results from data loading variations on the buses.

Furthermore, although the invention is described in the context of employing registers, flip-flops, and memory devices that are enabled in response to receiving positive edges of clock signal pulses, it should be appreciated that other types of logic devices may also be employed, such as, for example, those which are enabled in response to receiving negative edges of clock signal pulses.

Moreover, it should be note that the method of the invention is not limited to being used only in a memory subsystem of a computer system, as described above. That is, the method of the invention may also be employed to optimize the exchange of information between any suitable types of communicating devices (e.g., master and slave devices), such as devices employed in synchronous communication systems wherein one device controls one or more subservient devices.

Having described the various aspects of the invention, it can be appreciated that the invention provides a method wherein steps are performed of providing at least one clock pulse having a leading pulse edge and a trailing pulse edge. In the method, first data is transmitted from a first location (e.g., memory controller **1**) to a second location (e.g., DRAMs D1–Dn) in accordance with the at least one clock pulse. The first data has a leading edge and a trailing edge, and there is a leading phase between the leading edge of the first data and the leading pulse edge. Also, there is a trailing phase between the trailing edge of the first data and the trailing pulse edge. After the first data is received at the second location, further steps are performed of transmitting second data from the second location to the first location, and comparing the first and second data to determine if there are any errors in the second data. If there are errors in the second data, a further step is performed of varying the leading phase and the trailing phase to determine values thereof defining a bounded relationship between the at least one clock pulse and the first data within which the first data can be transmitted substantially without error. After the varying step is performed, a further step is performed of transferring further first data between the first and second locations using the bounded relationship.

While the invention has been particularly shown and described with respect to preferred embodiments thereof, it will be understood by those skilled in the art that changes in form and details may be made therein without departing from the scope and spirit of the invention.

What is claimed is:

1. A method for initiating a start-up operation of a system, the system having a master device and a slave device, the method comprising steps of:

exercising the slave device using the master device to determine a range within which temporal relationships of electrical signals need to be set in order to operate the system without error; and

setting the temporal relationships of the electrical signals so as to be within the determined range;

wherein the electrical signals include first and second electrical signals, and wherein the exercising step includes steps of:

applying the first electrical signals to the master device for generating second electrical signals;

applying the second electrical signals to the slave device;

transferring information between the master device and the slave device in accordance with temporal relationships between the second electrical signals;

varying the temporal relationships between first predetermined ones of the second electrical signals to determine upper and lower bounds of the range within which the temporal relationships need to be set in order for the information to be transferred from the slave device to the master device without error; and

wherein the setting step includes setting the temporal relationships between the second electrical signals so as to be between the upper and lower bounds.

2. A method as set forth in claim **1**, and further comprising a step of storing a record of the determined range for subsequent use in operating said system.

3. A method as set forth in claim **1**, wherein the step of varying includes a step of programming a plurality of programmable delay elements so as to vary the temporal relationships between the second electrical signals.

4. A method as set forth in claim **1**, further comprising a step of varying temporal relationships between second predetermined ones of the second electrical signals to determine a relationship that causes the information to be transferred from the master device to the slave device without error.

5. A method as set forth in claim **4**, and further comprising steps of:

setting the temporal relationships between the second predetermined ones of the electrical signals to the relationship determined to cause the information to be transferred from the master device to the slave device without error; and

further varying the temporal relationships between the first predetermined ones of the second electrical signals to determine revised upper and lower bounds of the range within which the temporal relationships need to be set in order for the information to be transferred from the slave device to the master device without error.

6. A method as set forth in claim **1**, wherein the master device includes a memory controller and wherein the slave device includes at least one memory device.

7. A data processing system, said data processing system comprising:

at least one memory device; and

memory control means, said memory control means being responsive to a start-up condition of said system for generating electrical signals to exercise the at least one memory device for setting a timing of said electrical signals so as to enable data to be stored in and read from the at least one memory device without error;

wherein the electrical signals include first and second electrical signals and wherein the system further comprises:

circuitry for applying the first electrical signals to the memory control means for generating second electrical signals;

circuitry for applying the second electrical signals to the memory device;

circuitry for transferring information between the memory control means and the memory device in accordance with temporal relationships between the second electrical signals;

circuitry for varying the temporal relationships between first Predetermined ones of the second electrical signals to determine upper and lower bounds of the range within which the temporal relationships need to be set in order for the information to be transferred from the memory device to the memory control means without

error; and circuitry for setting the temporal relationships between the second electrical signals so as to be between the upper and lower bounds.

8. A data processing system as set forth in claim **7**, wherein said memory control means is also for storing a record of the timing of said electrical signals.

9. A data processing system as set forth in claim **8**, wherein said memory control means further comprises:

means for generating the first electrical signals in response to said start-up condition;

an output port; and

at least one circuit block interposed between an output of said generating means and said output port, said at least one circuit block being responsive to said first electrical signals for generating said second electrical signals for applying said second electrical signals to said at least one memory device through said output port, wherein said at least one circuit block comprises at least one programmable delay element.

10. A data processing system as set forth in claim **7**, wherein the memory control means comprises a memory controller and wherein the at least one memory device comprises a plurality of memory devices included in at least one memory module.

11. A data processing system as set forth in claim **7**, wherein the at least one memory device comprises at least one Dynamic Random Access Memory (DRAM) device.

12. A data processing system as set forth in claim **7**, wherein the at least one memory device comprises at least one Synchronous Dynamic Random Access Memory (SDRAM) device.

13. A method for transmitting data from a first location to a second location in an electronic system, the method comprising steps of:

providing at least one clock pulse having a leading pulse edge and a trailing pulse edge from a clock which controls signals at said first location and at said second location;

transmitting first data from said first location to said second location in accordance with said at least one clock pulse, said first data having a leading edge and a trailing edge, there being a leading phase between said leading edge of said first data and said leading pulse edge, there also being a trailing phase between said trailing edge of said first data and said trailing pulse edge;

after the first data is received at the second location, transmitting second data from said second location to said first location;

comparing said first and second data to determine if there are any errors in said second data; and

if there are any errors in said second data, varying said leading phase and said trailing phase to determine values thereof defining a bounded relationship between said at least one clock pulse and said first data within which the first data can be transferred from said first location to said second location substantially without error.

14. A method according to claim **13**, wherein after the varying step is performed, a further step is performed of transferring further first data between the first and second locations using said bounded relationship.

15. An apparatus for transmitting data from a first location to a second location in an electronic system, said apparatus comprising:

means for providing at least one clock pulse having a leading pulse edge and a trailing pulse edge from a clock which controls signals at said first location and at said second location;

means for transmitting first data from said first location to said second location in accordance with said at least one clock pulse, said first data having a leading edge and a trailing edge, there being a leading phase between said leading edge of said first data and said leading pulse edge, there being a trailing phase between said trailing pulse edge and said trailing edge of said first data; means for transmitting second data from said second location to said first location;

means for comparing said first and second data to determine if there are any errors in said second data; and

means for varying said leading phase and said trailing phase to determine values thereof defining a bounded relationship between said at least one clock pulse and said first data within which said first data can be transmitted substantially without error.

16. An apparatus according to claim **15**, wherein said means for transmitting first data subsequently transfers further first data from said first location to said second location using said bounded relationship.

* * * * *

CERTIFICATE OF FILING AND SERVICE

I hereby certify that on this 9th day of May, 2011, two copies of the Confidential Brief of Intervenors NVIDIA Corporation et al. and two copies of the Non-Confidential Brief of Intervenors NVIDIA Corporation et al. were served by Federal Express to the following:

PAUL M. BARTKOWSKI
U.S. International Trade Commission
500 E Street, SW, Suite 614
Washington, DC 20436
Tel: (202) 708-5432
Fax: (202) 205-3111

MICHAEL J. JAKES
KATHLEEN DALEY
Finnegan, Henderson, Farabow
901 New York Avenue, NW
Washington, DC 20001-4413
Tel: (202) 408-4000
Fax: (202) 408-4400

RUFFIN B. CORDELL
Fish & Richardson, P.C.
1425 K Street, NW, 11th Floor
Washington, DC 20005-3500
Tel: (202) 783-5070
Fax: (202) 783-2331

I also certify pursuant to Fed. R. App. P. 25(a)(2)(B)(ii) and Fed. Cir. R. 31(b) that on this 9th day of May, 2011, 12 copies of the Brief of Intervenors NVIDIA Corporation et. al., including the original, and 5 copies of the Non-Confidential Brief of Intervenors NVIDIA Corporation et al., including the original, were filed by hand delivery to the Clerk of

the Court for the United States Court of Appeals for the Federal Circuit.

Respectfully submitted,

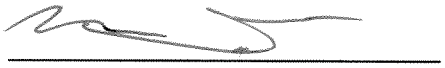
Mark S. Davies

CERTIFICATE OF COMPLIANCE
(No. 2010-1483)

Counsel for Intervenors NVIDIA Corporation certifies that the brief contained herein has a proportionally spaced 14-point typeface and contains 13,965 words, based on the "Word Count" feature of Microsoft Word, including footnotes and endnotes. Pursuant to Federal Rule of Appellate Procedure 32(a)(7)(B)(iii) and Federal Circuit Rule 32(b), this word count does not include the words contained in the Certificate of Interest, Certificate of Filing and Service, Table of Contents, Table of Authorities, Addendum, and Statement of Related Cases.

Dated: May 9, 2011

Respectfully submitted,



Mark S. Davies