US 20060010339A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2006/0010339 A1

Klein (43) Pub. Date: Jan. 12, 2006

(54) MEMORY SYSTEM AND METHOD HAVING SELECTIVE ECC DURING LOW POWER REFRESH

(76) Inventor: **Dean A. Klein**, Eagle, ID (US)

Correspondence Address:
Edward W. Bulchis, Esq.
DORSEY & WHITNEY LLP
Suite 3400
1420 Fifth Avenue
Seattle, WA 98101 (US)

(21) Appl. No.: **10/876,796**

(22) Filed: **Jun. 24, 2004**

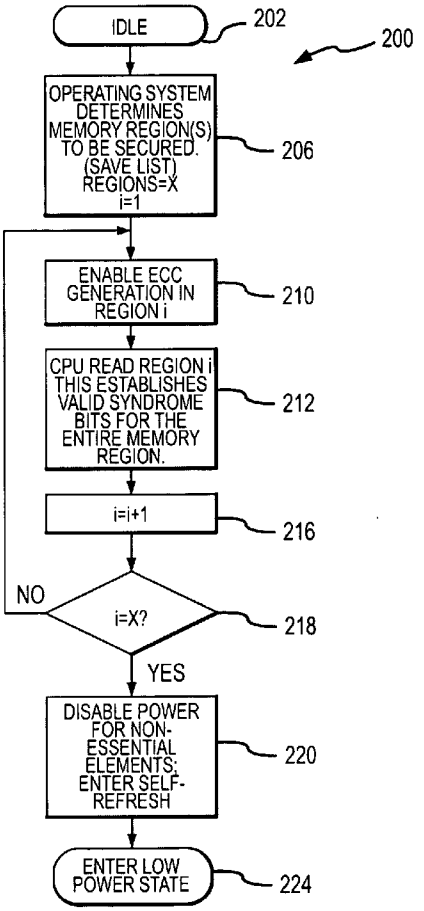
Publication Classification

(51) **Int. Cl.**
G06F 12/16 (2006.01)

(52) **U.S. Cl.** **714/5**; 711/106

(57) **ABSTRACT**

A computer system includes a processor coupled to a DRAM through a memory controller. The processor switches the DRAM to a low power refresh mode in which DRAM cells are refreshed at a sufficiently low rate that data retention errors may occur. Prior to switching the DRAM to the low power refresh mode, the processor identifies a region of an array of DRAM cells that contains essential data that needs to be protected from such data retention errors. The processor then reads data from the identified region, and either the DRAM or the memory controller generates error checking and correcting syndromes from the read data. The syndromes are stored in the DRAM, and the low power refresh mode is then entered. Upon exiting the low power refresh mode, the processor again reads the data from the identified region, and the read data is checked and corrected using the syndromes.



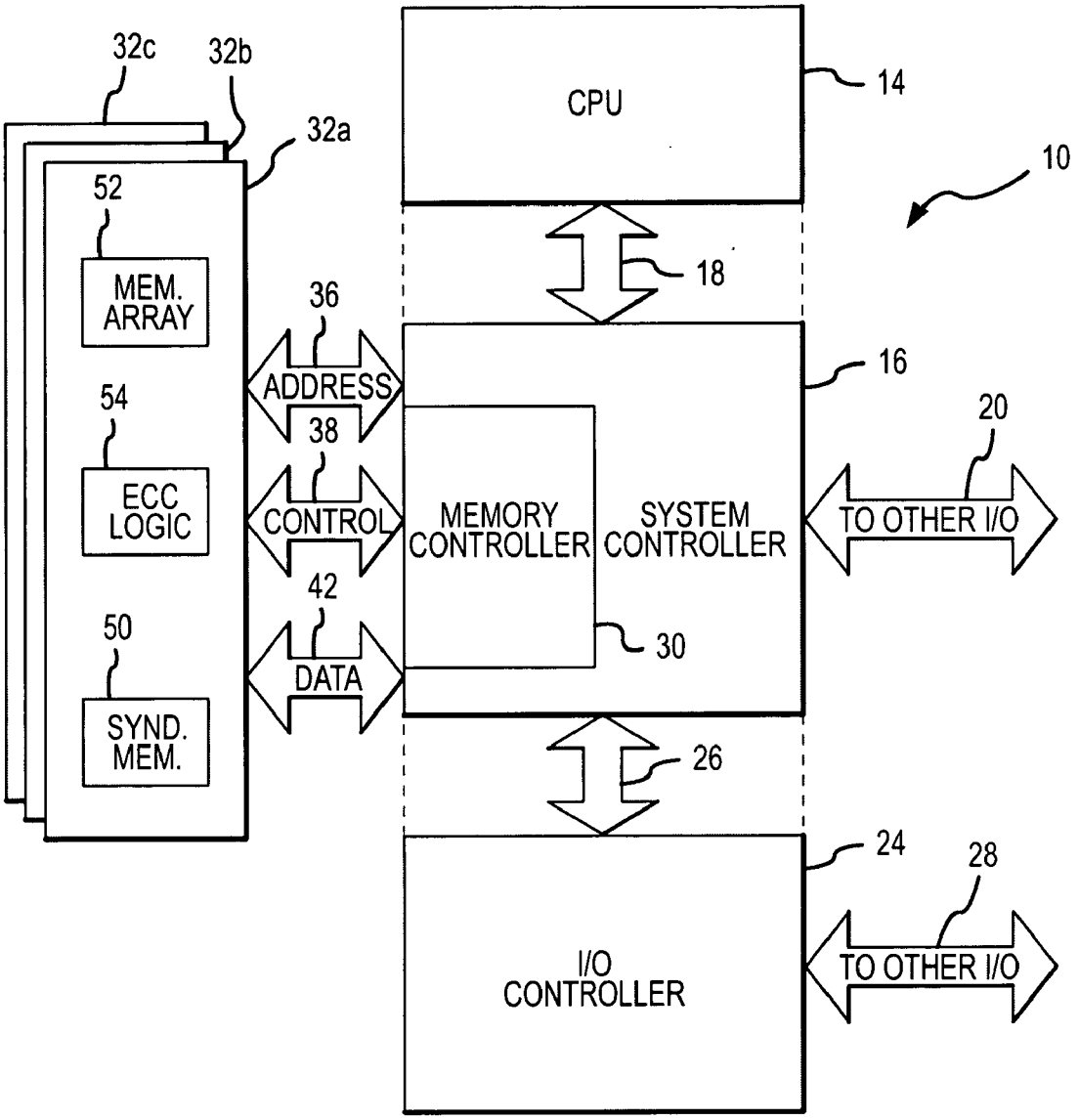

32c
32b
32a
52
MEM. ARRAY
54
ECC LOGIC
50
SYND. MEM.
36
ADDRESS
38
CONTROL
42
DATA
CPU
14
10
18
16
MEMORY CONTROLLER
SYSTEM CONTROLLER
30
20
TO OTHER I/O
26
24
28
I/O CONTROLLER
TO OTHER I/O

FIG.1

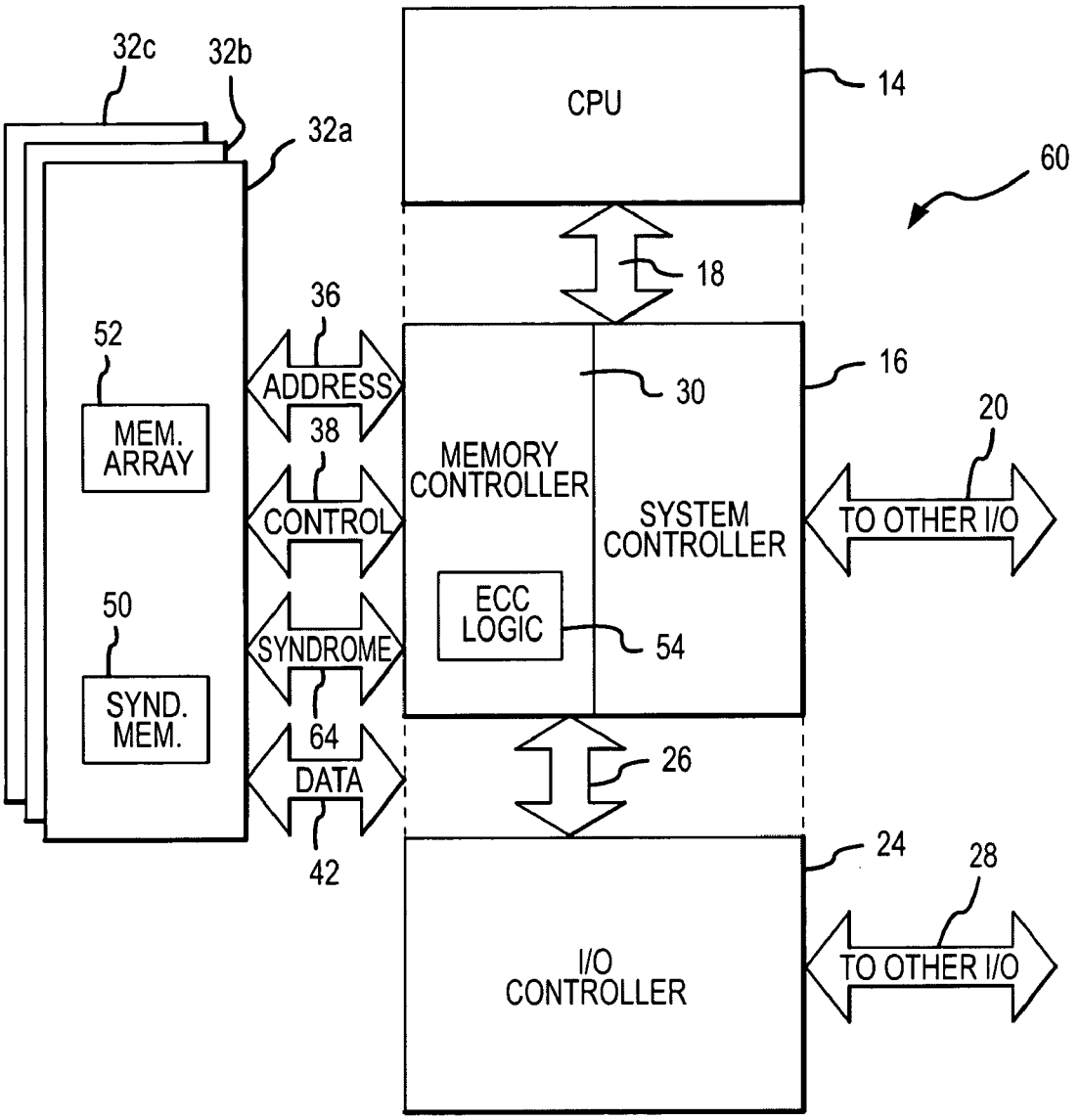

32c
32b
32a
CPU
14
60
18
36
52
ADDRESS
38
MEM. ARRAY
MEMORY CONTROLLER
30
16
20
CONTROL
SYSTEM CONTROLLER
TO OTHER I/O
50
ECC LOGIC
54
SYNDROME
SYND. MEM.
64
DATA
26
42
24
28
I/O CONTROLLER
TO OTHER I/O

FIG.2

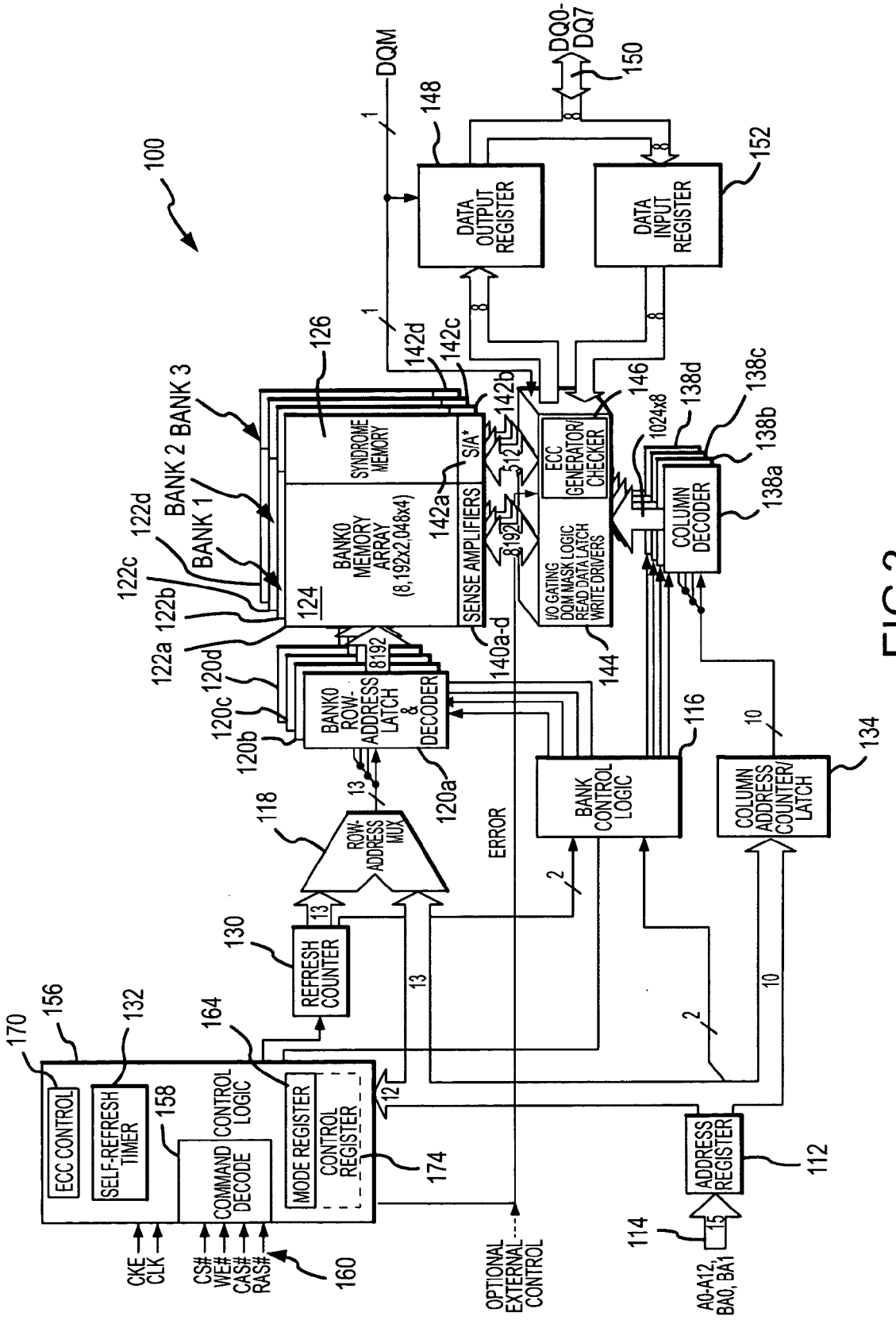

100
DQM
DQ0-DQ7
150
148
152
DATA OUTPUT REGISTER
DATA INPUT REGISTER
126
142d
142c
142b
146
138d
138c
138b
138a
1024x8
BANK 3
BANK 2
BANK 1
122d
122c
122b
122a
120d
120c
120b
120a
124
SYNDROME MEMORY
BANK0 MEMORY ARRAY (8,192x2,048x4)
S/A'
SENSE AMPLIFIERS
512
8192
ECC GENERATOR/ CHECKER
I/O GATING DQM MASK LOGIC READ DATA LATCH WRITE DRIVERS
COLUMN DECODER
140a-d
144
BANK0 ROW-ADDRESS LATCH & DECODER
8192
116
134
BANK CONTROL LOGIC
COLUMN ADDRESS COUNTER/ LATCH
118
ROW-ADDRESS MUX
ERROR
130
REFRESH COUNTER
156
132
164
170
158
ECC CONTROL
SELF-REFRESH TIMER
COMMAND DECODE
CONTROL LOGIC
MODE REGISTER
CONTROL REGISTER
174
160
CKE
CLK
CS#
WE#
CAS#
RAS#
OPTIONAL EXTERNAL CONTROL
ADDRESS REGISTER
112
114
A0-A12, BA0, BA1

FIG.3

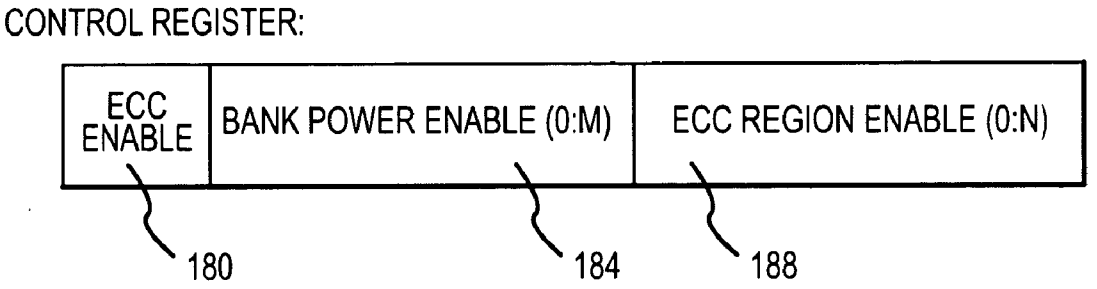

CONTROL REGISTER:
ECC ENABLE
BANK POWER ENABLE (0:M)
ECC REGION ENABLE (0:N)
180
184
188

FIG.4

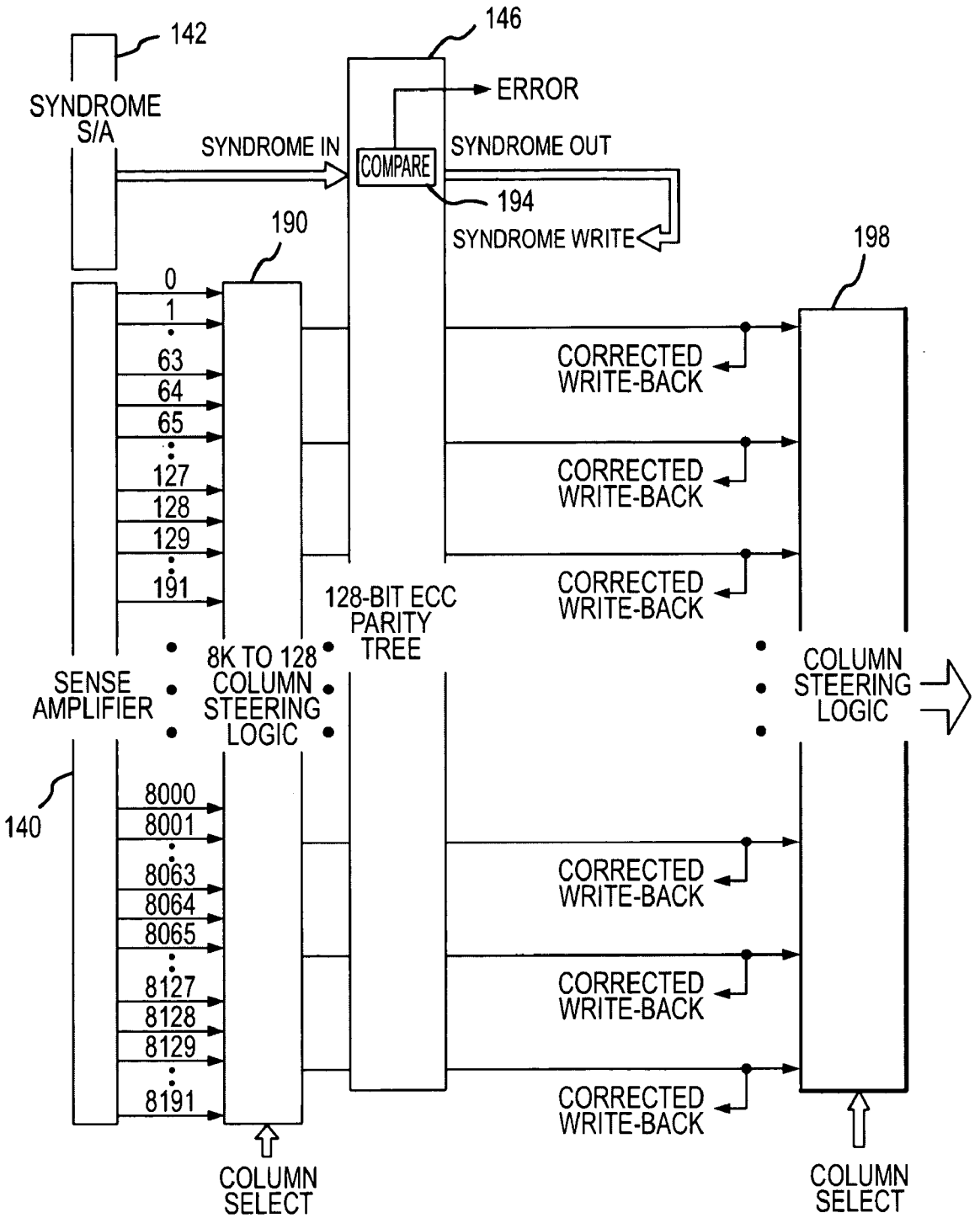

142
SYNDROME S/A
146
ERROR
SYNDROME IN
COMPARE
SYNDROME OUT
194
SYNDROME WRITE
190
198
0
1
63
64
65
127
128
129
191
CORRECTED WRITE-BACK
CORRECTED WRITE-BACK
CORRECTED WRITE-BACK
128-BIT ECC PARITY TREE
SENSE AMPLIFIER
8K TO 128 COLUMN STEERING LOGIC
COLUMN STEERING LOGIC
140
8000
8001
8063
8064
8065
8127
8128
8129
8191
CORRECTED WRITE-BACK
CORRECTED WRITE-BACK
CORRECTED WRITE-BACK
COLUMN SELECT
COLUMN SELECT

FIG.5

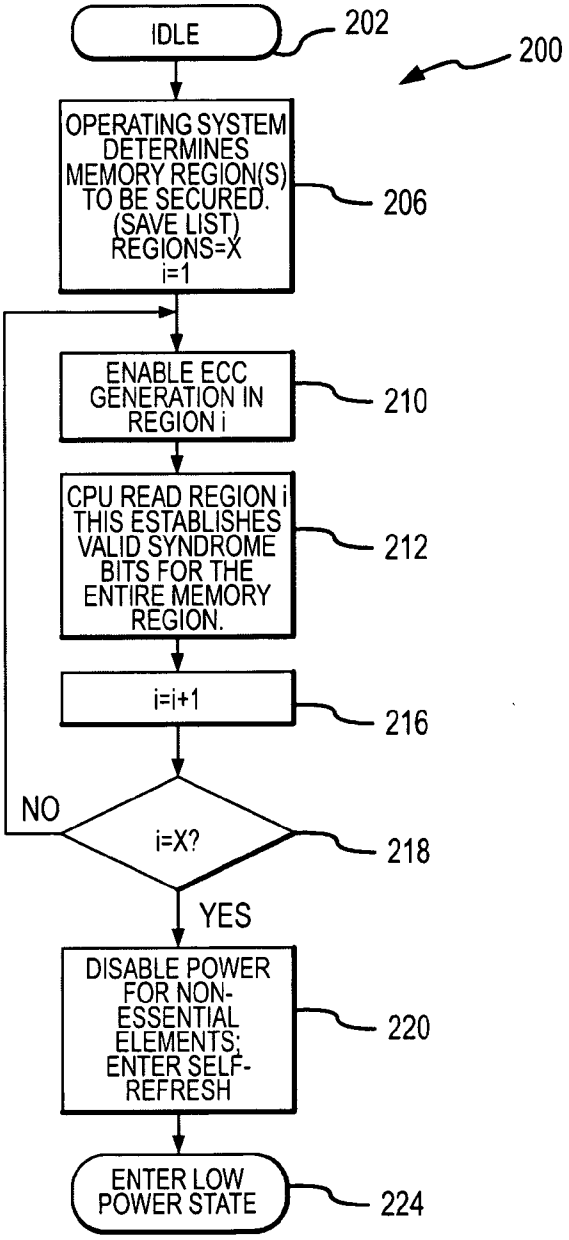

IDLE
202
200
OPERATING SYSTEM DETERMINES MEMORY REGION(S) TO BE SECURED. (SAVE LIST) REGIONS=X i=1
206
ENABLE ECC GENERATION IN REGION i
210
CPU READ REGION i THIS ESTABLISHES VALID SYNDROME BITS FOR THE ENTIRE MEMORY REGION.
212
i=i+1
216
NO
i=X?
218
YES
DISABLE POWER FOR NON-ESSENTIAL ELEMENTS; ENTER SELF-REFRESH
220
ENTER LOW POWER STATE
224

FIG.6

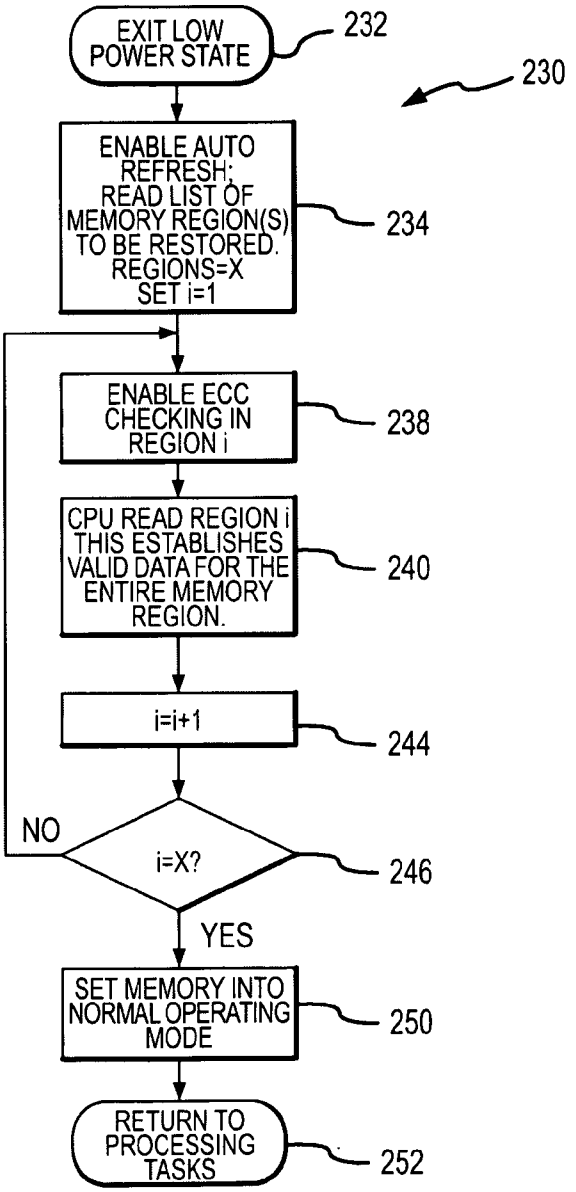

EXIT LOW POWER STATE
232
230
ENABLE AUTO REFRESH; READ LIST OF MEMORY REGION(S) TO BE RESTORED. REGIONS=X SET i=1
234
ENABLE ECC CHECKING IN REGION i
238
CPU READ REGION i THIS ESTABLISHES VALID DATA FOR THE ENTIRE MEMORY REGION.
240
i=i+1
244
NO
i=X?
246
YES
SET MEMORY INTO NORMAL OPERATING MODE
250
RETURN TO PROCESSING TASKS
252

FIG.7

MEMORY SYSTEM AND METHOD HAVING SELECTIVE ECC DURING LOW POWER REFRESH

TECHNICAL FIELD

[0001] This invention relates to dynamic random access memory ("DRAM") devices and systems, and, more particularly, to a method and system for allowing DRAM cells to be refreshed at a relatively low rate to reduce power consumption.

BACKGROUND OF THE INVENTION

[0002] As the use of electronic devices, such as personal computers, continue to increase, it is becoming ever more important to make such devices portable. The usefulness of portable electronic devices, such as notebook computers, is limited by the limited length of time batteries are capable of powering the device before needing to be recharged. This problem has been addressed by attempts to increase battery life and attempts to reduce the rate at which such electronic devices consume power.

[0003] Various techniques have been used to reduce power consumption in electronic devices, the nature of which often depends upon the type of power consuming electronic circuits that are in the device. Electronic devices, such notebook computers, typically include dynamic random access memory ("DRAM") devices that consume a substantial amount of power. As the data storage capacity and operating speeds of DRAM devices continues to increase, the power consumed by such devices has continued to increase in a corresponding manner. In general, the power consumed by a DRAM device increases with both the capacity and the operating speed of the DRAM devices.

[0004] The power consumed by DRAM devices is also affected by their operating mode. A DRAM device, for example, will generally consume a relatively large amount of power when the DRAM cells are being refreshed. As is well-known in the art, DRAM cells, each of which essentially consists of a capacitor, must be periodically refreshed to retain data stored in the DRAM device. Refresh is typically performed by essentially reading data bits from the memory cells in each row of a memory cell array and then writing those same data bits back to the same cells in the row. A relatively large amount of power is consumed when refreshing a DRAM because rows of memory cells in a memory cell array are being actuated in the rapid sequence. Each time a row of memory cells is actuated, a pair of digit lines for each memory cell are switched to complementary voltages and then equilibrated. As a result, DRAM refreshes tends to be particularly power-hungry operations. Further, since refreshing memory cells must be accomplished even when the DRAM is not being used and is thus inactive, the amount of power consumed by refresh is a critical determinant of the amount of power consumed by the DRAM over an extended period. Thus many attempts to reduce power consumption in DRAM devices have focused on reducing the rate at which power is consumed during refresh.

[0005] Refresh power can, of course, be reduced by reducing the rate at which the memory cells in a DRAM are being refreshed. However, reducing the refresh rate increases the risk of data stored in the DRAM cells being lost. More specifically, since, as mentioned above, DRAM cells are essentially capacitors, charge inherently leaks from the memory cell capacitors, which can change the value of a data bit stored in the memory cell over time. However, current leaks from capacitors at varying rates. Some capacitors are essentially short-circuited and are thus incapable of storing charge indicative of a data bit. These defective memory cells can be detected during production testing, and can then be repaired by substituting non-defective memory cells using conventional redundancy circuitry. On the other hand, current leaks from most DRAM cells at much slower rates that span a wide range. A DRAM refresh rate is chosen to ensure that all but a few memory cells can store data bits without data loss. This refresh rate is typically once every **64** ms. The memory cells that cannot reliably retain data bits at this refresh rate are detected during production testing and replaced by redundant memory cells. However, the rate of current leakage from DRAM cells can change after production testing, both as a matter of time and from subsequent production steps, such as in packaging DRAM chips. Current leakage, and hence the rate of data loss, can also be effected by environmental factors, such as the temperature of DRAM devices. Therefore, despite production testing, a few memory cells will typically be unable to retain stored data bits at normal refresh rates.

[0006] One technique that has been used to correct data errors in DRAMs is to generate an error correcting code "ECC from each item of stored data, and then store the ECC, known as a syndrome, along with the data. The use of ECC techniques during refresh could allow the power consumed by a DRAM device to be reduced because the ability of ECC to correct data retention errors would allow the refresh rate to be slowed to such an extent that errors can occur. Significantly reducing the refresh rate of a DRAM device would result in a substantial reduction in the power consumed by the DRAM device.

[0007] Although the use of ECC techniques during refresh could substantially reduce power consumption during refresh, it could impose significant cost penalties in both the cost and the performance of DRAM devices. In particular, the development cost and manufacturing cost of a DRAM device or a memory controller would be increased by the cost to develop and manufacture the additional circuitry needed to perform the ECC function. The increase in manufacturing cost for additional features in DRAM devices is normally manifested in a larger semiconductor die size, which reduces the yield from each semiconductor wafer. It can also be anticipated that the performance of DRAM devices would be impaired by reduced operating speeds resulting from the need to check and possibly correct all data read from the DRAM devices as well as the need to create syndromes for all data written to the DRAM devices.

[0008] There is therefore a need for a system and method for reducing power consumption by refreshing DRAM cells at a reduced rate without incurring the cost and performance penalties needed to check and possibly correct all of the data read from the DRAM device and to create syndromes for all data written to the DRAM device.

SUMMARY OF THE INVENTION

[0009] A method and system for refreshing memory cells in a dynamic random access memory ("DRAM") device is coupled to a processor in a computer system. The memory

cells in the DRAM are refreshed at a reduced power rate that is sufficiently slow that data retention errors can be expected to occur during refresh. However, the expected data retention errors are corrected using ECC techniques applied only to memory cells containing essential data that should be protected from data retention errors. More specifically, prior to refreshing the memory cells at the reduced power rate, a determination is made, preferably by the processor, which memory cells are storing the essential data. ECC techniques are then used to check and correct the essential data without using ECC techniques to check and correct data stored in other memory cells. Prior to refreshing the memory cells at the reduced power rate, the essential data are read, and corresponding syndromes are generated and stored. When departing from the low power refresh rate, the essential data are again read, and the stored syndromes are used to check the read data for errors. If any errors are found, the syndromes are used to correct the data, and the corrected data are written to the DRAM.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] **FIG. 1** is a block diagram of a computer system according to one embodiment of the invention.

[0011] **FIG. 2** is a block diagram of a computer system according to another embodiment of the invention.

[0012] **FIG. 3** is a block diagram of a DRAM device according to one embodiment of the invention.

[0013] **FIG. 4** is a schematic diagram illustrating the data structure of control data that may be stored in a control register in the DRAM of **FIG. 3**.

[0014] **FIG. 5** is a block diagram of a portion of column interface circuitry used in the DRAM of **FIG. 3**.

[0015] **FIG. 6** is a flow chart showing the operation of the DRAM of **FIG. 3** when entering a low power refresh mode using ECC techniques to correct data retention errors for essential data.

[0016] **FIG. 7** is a flow chart showing the operation of the DRAM of **FIG. 3** when exiting the low power refresh mode.

DETAILED DESCRIPTION

[0017] A computer system **10** according to one embodiment of the invention is shown in **FIG. 1**. The computer system **10** includes a central processor unit ("CPU") **14** coupled to a system controller **16** through a processor bus **18**. The system controller **16** is coupled to input/output ("I/O") devices (not shown) through a peripheral bus **20** and to an I/0 controller **24** through an expansion bus **26**. The I/O controller **24** is also connected to various peripheral devices (not shown) through another I/0 bus **28**.

[0018] The system controller **16** includes a memory controller **30** that is coupled to several dynamic random access memory ("DRAM") device **32***a-c* through an address bus **36**, a control bus **38**, and a data bus **42**. The locations in each of the DRAMs **32***a-c* to which data are written and data are read are designated by addresses coupled to the DRAMs **32***a-c* on the address bus **36**. The operation of the DRAMs **32***a-c* are controlled by control signals coupled to the DRAMs **32***a-c* on the control bus **38**.

[0019] In other embodiments of the invention, the memory controller **30** may be coupled to one or more memory modules (not shown) through the address bus **36**, the control bus **38**, and the data bus **42**. Each of the memory modules would normally contain several of the DRAMs **32**.

[0020] With further reference to **FIG. 1**, the DRAMs **32***a-c* each include a syndrome memory **50**, a memory array **52** and ECC logic **54**. The ECC logic **54** generates a syndrome from write data received from the memory controller **30**, and stores the syndrome in the syndrome memory **40** while the write data are being stored in the memory array **52** of the DRAM **32**. When data are read from the DRAM **32**, the read data are coupled from memory array **52** to the ECC logic **54** and the syndrome are coupled from the syndrome memory **40** to the ECC logic **54**. The ECC logic **54** then uses the syndrome to determine if the read data contains an erroneous data bit, and, if more than one data bit is not in error, to correct the erroneous data bit. The corrected read data are then coupled to the memory controller **30** through the data bus **42**. Although the syndrome memory **50** may be a separate memory array in the DRAM **32** as shown in **FIG. 1**, it may alternatively be included in the same array of DRAM cells that are used to store data, as explained in greater detail below. The use of ECC allows the refresh rate of the memory cells in the memory array **52** to be reduced to a rate at which some data retention errors can occur since such errors can be corrected using the syndromes stored in the syndrome memory **50** and the ECC logic **52**.

[0021] The DRAM **32** is able to use ECC techniques with relatively little of the costs and performance impairments associated with conventional ECC techniques because the CPU **14** uses ECC techniques to check and correct only the data stored in the memory array **52** that needs to be correct. More specifically, conventional computer systems and other electronic systems use DRAM devices to store a variety of data types. For example, a DRAM device may be used as a "scratch pad" memory or to store audio data or video data that is being coupled to a display. This type of data is usually overwritten with new data quite frequently, and is therefore generally not used after an extended refresh. Other data, such as program instructions, spreadsheet data, word processing documents, must be protected during refresh. Therefore, the CPU **14** applies ECC techniques to this type of data.

[0022] The CPU **14** can determine which data to protect through a variety of means. For example, the CPU **14** can store essential data that needs to be protected in only certain regions of the memory array **52**, and then apply ECC techniques to these regions. The CPU **14** can also keep track of where in the memory array **52** essential data are stored, and then apply ECC techniques to these regions. Other techniques may also be used.

[0023] In operation, prior to each of the DRAMs **32***a-c* entering a low power refresh mode, the DRAMs **32***a-c* perform a read and syndrome generating operation. More specifically, the CPU **14** enables the ECC logic **54** by suitable means, such as by coupling a command signal to the DRAMs **32***a-c* through the memory controller **30** and control bus **38** that enables a control register in the DRAM **32**. However, the CPU may enable the ECC logic **54** by other means, such as by coupling control signals directly to the ECC logic **54**, by coupling an unsupported command to the DRAM **32**, use of a specific sequence of operations, or by other means. In any case, once the ECC logic **54** has been

enabled, the CPU **14** performs a read operation to the regions of the memory array **52** that store essential data that needs to be protected. The read operation is preferably performed in a burst read mode to minimize the time required for the read operation. Regions of the memory array **52** that store non-essential data, such as regions used as image buffers, temporary buffers and screen buffers, are not read. During the read operation, the DRAM **32** generates syndromes from the read data, and stores the syndromes in the syndrome memory **50**. The DRAM **32** then enters a low power refresh mode in which the memory cells in the array **52** are refreshed at a rate that is sufficiently low that data retention errors may occur. In one embodiment of the invention, the CPU **14** leaves the ECC logic **54** enabled during the low power refresh mode to correct any data retention errors as they occur. In another embodiment of the invention, the CPU **14** disables the ECC logic **54** after all of the syndromes have been stored and before entering the low power refresh mode. In this embodiment, the CPU **14** corrects any data retention errors that have occurred when exiting the low power refresh mode, as explained in greater detail below.

[**0024**] When exiting the low power refresh mode, the DRAMs **32***a-c* perform a read and correct operation. More specifically, the CPU **14** enables the ECC logic **54** if it was not enabled during the refresh mode. The CPU **14** then reads data from the protected regions of the memory array **52**, again preferably using a burst read mode. During these read operations, the ECC logic **54** receives the read data from the memory array **52** and the corresponding syndromes from the syndrome memory **50**. The ECC logic **54** then uses the syndromes to check the read data and to correct any errors that are found. The ECC logic **54** then writes the corrected data to the memory array **52**. Once the protected regions of the memory array **52** have been read, and the refresh rate increased to the normal refresh rate, the CPU **14** can disable the ECC logic **54**.

[**0025**] In other embodiments of the invention, the CPU **14** initiates a read operation prior to entering the low power refresh mode, but the actual reading of data from the protected areas is accomplished by sequencer logic in the DRAMs **32** or in a memory module containing the DRAMs **32**. The operation of the sequencer logic could be initiated by commands from the CPU **14** other than a read command, such as by issuing commands for a "dummy" operation, i.e., an operation that is not actually implemented by the DRAMs **32**.

[**0026**] In still another embodiment of the invention, the data stored in ECC protected areas regions of the memory array **52** are not checked and corrected when exiting the low power refresh mode. Instead, the ECC mode remains active during normal operation, and the data stored in the ECC protected regions are checked using the stored syndromes whenever that data are read during normal operation. This embodiment requires that the syndrome memory **50** remain powered during normal operation, at least until all of the data stored in the protected regions have been read. A complicating factor is the possibility of the data stored in the protected region being changed by a write to that region without the syndrome also being changed accordingly. Of course, if the write is for a data word equal in size to the data words used to create the syndromes, there would be no problem because the protected data would be entirely replaced after exiting the low power refresh mode. However, if a write occurs for part of a word used to create a stored syndrome, then the syndrome will no longer correspond to the modified word. As a result, when the word is subsequently read, the ECC logic **54** would report a data retention error even if the data are not in error. There are several ways of handling this problem. For example, the operating system being executed by the CPU **14** could determine which stored data words will be the subject of a partial write. The CPU **14** could then check these words for errors when exiting the low power refresh mode, and correct any errors that are found. Another technique would be to check and correct each stored data word just before a partial write to the data word occurs. Other techniques may also be used.

[**0027**] A computer system **60** according to another embodiment of the invention is shown in **FIG. 2**. The computer system **60** is very similar in structure and operation to the computer system **10** of **FIG. 1**. Therefore, in the interests of brevity, corresponding components have been provided with the same reference numerals, and a description of their operation will not be repeated. The computer system **60** of **FIG. 2** differs from the computer system **10** of **FIG. 1** primarily in the addition of a syndrome bus **64** and the elimination of the ECC logic **54** from the DRAMs **32**. In the computer system **60**, the ECC logic **54** is located in the memory controller **30**. Therefore, prior to entering the low power refresh mode, read data from the protected regions of the memory array **52** are coupled to the ECC logic **54** in the memory controller **30** through the data bus **42**. The ECC logic **54** then generates the syndromes, which are coupled to the syndrome memory **50** in the DRAMs **32** through the syndrome bus **64**. When exiting the low power refresh mode, the read data in the protected regions are coupled to the ECC logic **54** through the data bus **42**, and the corresponding syndromes are coupled from the syndrome memory **50** to the ECC logic **54** through the syndrome bus **64**. The ECC logic **54** then checks the read data, corrects any errors that are found, and couples corrected data through the data bus **42**, which are written to the memory array **52**. The advantage of the computer system **10** of **FIG. 1** over the memory device **60** is that the DRAMs **32** used in the system **10** of **FIG. 1** are plug compatible with conventional DRAMs, thus making it unnecessary to physically alter the computer system to accommodate selective ECC during low power refresh.

[**0028**] A synchronous DRAM **100** ("SDRAM") according to one embodiment of the invention is shown in **FIG. 3**. The SDRAM **100** includes an address register **112** that receives bank addresses, row addresses and column addresses on an address bus **114**. The address bus **114** is coupled to the memory controller **30** (**FIG. 1**). Typically, a bank address is received by the address register **112** and is coupled to bank control logic **116** that generates bank control signals, which are described further below. The bank address is normally coupled to the SDRAM **100** along with a row address. The row address is received by the address register **112** and applied to a row address multiplexer **118**. The row address multiplexer **118** couples the row address to row address latch & decoder circuit **120***a-d* for each of several banks of memory cell arrays **122***a-d,* respectively. Each bank **120***a-d* is divided into two sections, a data second **124** that is used for storing data, and a syndrome section **126** that is used for storing syndromes. Thus, unlike the DRAM **32** of **FIGS. 1 and 2**, a separate syndrome memory **50** is not used in the SDRAM **100** of **FIG. 3**.

[0029] Each of the latch & decoder circuits **120***a-d* is selectively enabled by a control signal from the bank control logic **116** depending on which bank of memory cell arrays **122***a-d* is selected by the bank address. The selected latch & decoder circuit **120** applies various signals to its respective bank **122** as a function of the row address stored in the latch & decoder circuit **120**. These signals include word line voltages that activate respective rows of memory cells in the banks **122**. The row address multiplexer **118** also couples row addresses to the row address latch & decoder circuits **120***a-d* for the purpose of refreshing the memory cells in the banks **122***a-d*. The row addresses are generated for refresh purposes by a refresh counter **130**. During operation in a self-refresh mode, the refresh counter **130** periodically begins operating at times controlled by a self-refresh timer **132**. The self-refresh timer **132** preferably initiates refreshes at a relatively slow rate in the low power refresh mode, as explained above.

[0030] After the bank and row addresses have been applied to the address register **112**, a column address is applied to the address register **112**. The address register **112** couples the column address to a column address counter/latch circuit **134**. The counter/latch circuit **134** stores the column address, and, when operating in a burst mode, generates column addresses that increment from the received column address. In either case, either the stored column address or incrementally increasing column addresses are coupled to column address & decoders **138***a-d* for the respective banks **122***a-d*. The column address & decoders **138***a-d* apply various signals to respective sense amplifiers **140***a-d* and **142***a-d* through column interface circuitry **144**. The column interface circuitry **144** includes conventional I/O gating circuits, DQM mask logic, read data latches for storing read data from the memory cells in the banks **122** and write drivers for coupling write data to the memory cells in the banks **122**. The column interface circuitry **144** also includes an ECC generator/checker **146** that essentially performs the same function as the ECC logic **54** in the DRAMS **32** of **FIGS. 1 and 2**.

[0031] Syndromes read from the syndrome section **126** of one of the banks **122***a-d* are sensed by the respective set of sense amplifiers **142***a-d* and then coupled to the ECC generator checker **146**. Data read from the data section **124** one of the banks **122***a-d* are sensed by the respective set of sense amplifiers **140***a-d* and then stored in the read data latches in the column interface circuitry **144**. The data are then coupled to a data output register **148**, which applies the read data to a data bus **150**. Data to be written to the memory cells in one of the banks **122***a-d* are coupled from the data bus **150** through a data input register **152** to write drivers in the column interface circuitry **144**. The write drivers then couple the data to the memory cells in one of the banks **122***a-d*. A data mask signal "DQM" is applied to the column interface circuitry **144** and the data output register **148** to selectively alter the flow of data into and out of the column interface circuitry **144**, such as by selectively masking data to be read from the banks of memory cell arrays **122***a-d*.

[0032] The above-described operation of the SDRAM **100** is controlled by control logic **156**, which includes a command decoder **158** that receives command signals through a command bus **160**. These high level command signals, which are generated by the memory controller **30** (**FIG. 1**), are a clock a chip select signal CS#, a write enable signal WE#, a column address strobe signal CAS#, and a row address strobe signal RAS#, with the "#" designating the signal as active low. Various combinations of these signals are registered as respective commands, such as a read command or a write command. The control logic **156** also receives a clock signal CLK and a clock enable signal CKE#, which cause the SDRAM **100** to operate in a synchronous manner. The control logic **156** generates a sequence of control signals responsive to the command signals to carry out the function (e.g., a read or a write) designated by each of the command signals. The control logic **156** also applies signals to the refresh counter **130** to control the operation of the refresh counter **130** during refresh of the memory cells in the banks **122**. The control logic **156** also applies signals to the refresh timer **132** to control the refresh rate and allow the SDRAM **100** to operate in the low power refresh mode. The control signals generated by the control logic **156**, and the manner in which they accomplish their respective functions, are conventional. Therefore, in the interest of brevity, a further explanation of these control signals will be omitted.

[0033] The control logic **156** also includes a mode register **164** that may be programmed by signals coupled through the command bus **160** during initialization of the SDRAM **100**. The mode register **164** then generates mode control signals that are used by the control logic **156** to control the operation of the SDRAM **100** in various modes, such as the low power refresh mode. The mode register **164** may also include an ECC controller **170** that causes the control logic **156** to issue control signals to the ECC generator checker **146** and other components to generate syndromes for storage in the syndrome section **126** of the banks **122***a-d*, and to check and correct data read from the data section **124** of the banks **122***a-d* using syndromes stored in the sections **126**. The ECC controller **170** is preferably enabled and disabled by a mode signal from the mode register **164**, as explained above.

[0034] In an alternative embodiment, the control logic **156** may include a control register **174** that can receive control signals from the CPU **14** (**FIG. 1**) to directly control the operation of the ECC generator checker **146**, as explained above. The contents of the control register **174** in one embodiment of the invention are shown in **FIG. 4**. A first bit **180** of the control register **174** is either set or reset to enable or disable, respectively, the ECC mode. The next M+1 bits **184** of the control register **180** selects which memory banks **122** are to be powered, thereby allowing the CPU **14** to selectively apply power to each of the memory banks **122**. In the SDRAM **100** of **FIG. 3**, there are 4 bits bank power control bits **184** for the respective banks **122***a-d*. The final N+1 bits **188** select each region of each bank **122** that is to be ECC protected. The bits **188** could allow only a portion of a bank **122** to be refreshed, or all of a bank to be refreshed.

[0035] The interfaces between the sense amplifiers **140**, **142**, the ECC generator/checker **146** and certain components in the column interface circuitry **144** are shown in greater detail in **FIG. 5**. The sense amplifiers **142** coupled to the data sections **124** of the memory banks **122***a-d* output respective data bits for respective columns, which are applied to column steering logic **190**. In the embodiment shown in **FIG. 5**, the sense amplifiers **142** output respective data bits for 8,192 columns. The column steering logic **190** uses the 6 most significant bits **2-7** of a column address to select 1 of 64 128-bit groups of data bits and couples the data

bits to the ECC generator/checker **146**. The sense amplifiers **140** coupled to the syndrome section of the memory banks **122***a-d* couple a syndrome corresponding to the read data directly to the ECC generator/checker **146**.

[**0036**] The ECC generator/checker **146** includes a comparator **194** that provides an error indication in the event the read data contains an error. The ECC generator/checker **146** then couples the corrected 128-bit word to additional column steering logic **198**, and also couples the corrected 128-bit word back through the column steering logic **180** to the banks **122***a-d* so that the banks will now contain correct data. The column steering logic **198** uses the 2 least significant bits **0-1** of a column address to select 1 of 4 32-bit groups of data bits and couples the data bits to the memory controller **30** (**FIG. 1**), as previously explained. It is not necessary for the column steering logic **198** to couple the syndrome to the memory controller **30** in the computer system **10** of **FIG. 1** so that the operation error checking and correction function is transparent to the memory controller **30**. Also, although 128 bits of write data are used to form the syndrome, it is not necessary for the DRAM **32** to include externally accessible data terminals for each of these 128 bits.

[**0037**] The operation of the SDRAM **100** when entering and exiting the ECC protected low power refresh mode will now be explained with reference to **FIGS. 6 and 7**. With reference to **FIG. 6**, a procedure **200** is initially in an idle state **202** prior to entering the low power refresh mode. Prior to entering the low power refresh mode, the operating system for the CPU **14** determines at step **206** the regions of the memory banks **122** that are to be ECC protected. More specifically, a variable "i" designating the first region of memory that will be protected in initialized to "1", and the last region of memory that will be protected is set to a variable "X." A record of this determination is also stored at step **206** in a suitable location, such as a register internal to the CPU **14** or memory controller **30**. When the ECC protected low power refresh mode is to be entered, the CPU **14** first enables the ECC mode at step **210**. This is accomplished by either writing a mode bit to the mode register **164** or writing appropriate bits to the control register **174**, as explained above with reference to **FIG. 4**. The CPU **14** then reads a first region of memory (i=1) in one of the banks **122** at step **212**, and, in doing so, generates and stores syndromes for the read data. The region to be protected ("i") is then incremented by 1 at step **216**, and a check is made at step **218** to determine if the region currently being ECC protected is the final region that will be protected. If not, the process returns to repeat steps **210-218**.

[**0038**] When all of the protected regions have been read, the procedure branches from step **218** to step **220** where the CPU **14** disables the ECC protection. This can be accomplished by the CPU **14** either writing a mode bit to the mode register **164** or resetting the bit **180** in the control register **174**, as explained above with reference to **FIG. 4**. The CPU **14** then enters the low power refresh mode at step **224**, which is preferably a self-refresh mode, the nature of which is well-known to one skilled in the art. This can be accomplished by the CPU writing an appropriate bit to the mode register **164**. The control logic **156** then issues a control signal to the refresh timer **132** to reduce the refresh rate. The SDRAM **100** then operates in a reduced refresh rate, which substantially reduces the power consumed by the SDRAM **100**.

[**0039**] When the SDRAM **100** is to exit the low power refresh mode, a procedure **230** shown in **FIG. 7** is used. The refresh exit procedure is initiated at step **232**, and the normal refresh rate used for auto refresh is initiated at step **234**. The variable X identifying the regions of memory that were ECC protected is also read at step **234**. The CPU **14** then enables the ECC checking mode at step **238** to check the first region of memory that was ECC protected. Again, this can be accomplished by the CPU **14** either writing a mode bit to the mode register **164** or writing appropriate bits to the control register **174**. The CPU **14** then reads a first region of memory (i=1) in one of the banks **122** at step **240**, and the ECC generator checker **146** checks the read data for errors and corrects any errors that are found. The corrected data is then written to the region of memory being read. The region being checked is then incremented by 1 at step **244**, and a check is made at step **246** to determine if the region currently being checked is the final region that was protected. If not, the process returns to repeat steps **238-246**.

[**0040**] When data from all of the protected regions have been read, checked and corrected if necessary, the procedure branches from step **246** to step **250** in which the CPU **14** switches the SDRAM **100** to the normal operating mode. This can be accomplished by the CPU **14** either writing a mode bit to the mode register **164** or resetting the bit **180** in the control register **174**, as explained above with reference to **FIG. 4**. The CPU **14** then enters the normal operating mode at step **252**.

[**0041**] Although the present invention has been described with reference to the disclosed embodiments, persons skilled in the art will recognize that changes may be made in form and detail without departing from the spirit and scope of the invention. Such modifications are well within the skill of those ordinarily skilled in the art. Accordingly, the invention is not limited except as by the appended claims.

1. In a computer system having a processor coupled to a dynamic random access memory ("DRAM") device, a method of reducing the power consumed by the DRAM device, comprising:

refreshing DRAM cells in the DRAM device at a first rate when the DRAM device is active;

refreshing the DRAM cells in the DRAM device at a second rate when the DRAM device is inactive, the second rate being substantially slower than the first rate;

prior to transitioning from the first rate to the second rate, transitioning to an ECC protection mode by:

determining which DRAM cells are storing data that should be protected from data retention errors;

reading data from the DRAM cells determined to be storing data that should be protected;

generating ECC syndromes corresponding to the read data; and

storing the generated syndromes; and

when transitioning from the second rate to the first rate, transitioning from the ECC protection mode by:

reading data from the DRAM cells that are storing data that should be protected;

reading the stored ECC syndromes corresponding to the read data;

using the syndromes to determine if any of the read data are in error;

correcting any read data found to be in error; and

storing the corrected data in the DRAM cells.

2. The method of claim 1 wherein the act of determining which DRAM cells are storing data that should be protected from data retention errors comprises determining which DRAM cells are storing data that should be protected from data retention errors based on a record of the location of the DRAM cells.

3. The method of claim 1 wherein the act of determining which DRAM cells are storing data that should be protected from data retention errors comprises determining which DRAM cells are storing data that should be protected from data retention errors based on a record of the data stored in the DRAM cells.

4. The method of claim 1 wherein the act of determining which DRAM cells are storing data that should be protected from data retention errors comprises using the processor to determine which DRAM cells are storing data that should be protected from data retention errors.

5. The method of claim 1 wherein the act of reading data from the DRAM cells determined to be storing data that should be protected prior to transitioning from the first rate to the second rate comprises reading the data in a burst read operation.

6. The method of claim 1 wherein the act of reading data from the DRAM cells determined to be storing data that should be protected when transitioning to the ECC protection mode comprises using the processor to read the data from the DRAM cells determined to be storing data that should be protected.

7. The method of claim 1 wherein the act of reading data from the DRAM cells determined to be storing data that should be protected when transitioning to the ECC protection mode comprises initiating and controlling a read operation from within the DRAM device without using the processor.

8. The method of claim 1 wherein the act of generating ECC syndromes corresponding to the read data comprises generating the ECC syndromes within the DRAM device.

9. The method of claim 1 wherein the system further comprises a memory controller coupling the processor to the DRAM device, and wherein the act of generating ECC syndromes corresponding to the read data comprises generating the ECC syndromes within the memory controller.

10. The method of claim 1 wherein the act of storing the generated syndromes comprises storing the generated syndromes within the DRAM device.

11. The method of claim 10 wherein the act of storing the generated syndromes within the DRAM device comprises storing the generated syndromes in DRAM cells.

12. The method of claim 1 wherein the act of reading data from the DRAM cells that are storing data that should be protected when transitioning from the ECC protection mode comprises reading the data in a burst read operation.

13. The method of claim 1 wherein the act of reading data from the DRAM cells that are storing data that should be protected when transitioning from the ECC protection mode comprises using the processor to read the data from the DRAM cells that are storing data that should be protected.

14. The method of claim 1 wherein the act of reading data from the DRAM cells that are storing data that should be protected when transitioning from the ECC protection mode comprises initiating and controlling a read operation from within the DRAM device without using the processor.

15. The method of claim 1 wherein the act of reading the stored ECC syndromes corresponding to the data read when transitioning from the ECC protection mode comprises reading the stored ECC syndromes from the DRAM device.

16. The method of claim 1 wherein the act of using the syndromes to determine if any of the read data are in error comprises determining if any of the read data are in error within the DRAM device using the syndromes.

17. The method of claim 1 wherein the system further comprises a memory controller coupling the processor to the DRAM device, and wherein the act of using the syndromes to determine if any of the read data are in error comprises determining if any of the read data are in error within the memory controller using the syndromes.

18. The method of claim 1 wherein the act of correcting any read data found to be in error comprises using the DRAM device to correct any read data found to be in error.

19. The method of claim 1 wherein the system further comprises a memory controller coupling the processor to the DRAM device, and wherein the act of correcting any read data found to be in error comprises using the memory controller to correct any read data found to be in error.

20. The method of claim 1 wherein the act of storing the corrected data in the DRAM cells comprises using the processor to store the corrected data in the DRAM cells.

21. The method of claim 1 wherein the act of storing the corrected data in the DRAM cells comprises writing the corrected data to the DRAM cells in a burst write operation.

22. The method of claim 1 wherein the act of transitioning to an ECC protection mode comprises using the processor to transition to the ECC protection mode.

23. The method of claim 1 wherein the DRAM device further comprises a mode register, and wherein the act of using the processor to transition to the ECC protection mode comprises using the processor to store a first mode control bit in the mode register, the first mode control bit corresponding to the ECC protection mode.

24. The method of claim 1 wherein the DRAM device further comprises a control register, and wherein the act of using the processor to transition to the ECC protection mode comprises using the processor to store control data in the control register, the control data comprising a first bit enabling the ECC protection mode, and a plurality of second bits that specify the DRAM cells determined to be storing data that should be protected.

25. The method of claim 1 wherein the DRAM device further comprises a mode register, and wherein the act of using the processor to transition from the ECC protection mode comprises using the processor to store a second mode control bit in the mode register, the second mode control bit corresponding to a normal operating mode.

26. The method of claim 1 wherein the DRAM device further comprises a control register, and wherein the act of

using the processor to transition from the ECC protection mode comprises using the processor to store a bit disabling the ECC protection mode.

27. The method of claim 1 wherein the acts of reading data from the DRAM cells that are storing data that should be protected, reading the stored ECC syndromes corresponding to the read data, and using the syndromes to determine if any of the read data are in error when transitioning from the second rate to the first rate are performed only for the DRAM cells storing data words to which data will be written to a part of the stored data word.

28. The method of claim 27 wherein the acts of reading data from the DRAM cells that are storing data that should be protected, reading the stored ECC syndromes corresponding to the read data, using the syndromes to determine if any of the read data are in error, and correcting any read data found to be in error are performed during normal operation of the DRAM for the DRAM cells storing data words to which data will not be written to a part of the stored data word.

29. The method of claim 28, further comprising providing a tag for each data word that should be protected, the tag indicating whether or not a valid syndrome exists for the corresponding data word.

30. The method of claim 29, further comprising setting the tag for each word to indicate a valid syndrome does not exist for the word when data is written to a part of one of the stored data words.

31. A method of refreshing memory cells in a dynamic random access memory ("DRAM") device, the method comprising:

refreshing the memory cells at a reduced power rate that is sufficiently slow that data retention errors can be expected to occur during refresh; and

prior to refreshing the memory cells at the reduced power rate, determining which memory cells are storing essential data that should be protected from data retention errors, and use ECC techniques to check and correct the essential data without using ECC techniques to check and correct data stored in other of the memory cells.

32. The method of claim 31 wherein the act of using ECC techniques to check and correct the essential data comprises, prior to refreshing the memory cells at the reduced power rate:

identifying the memory cells storing essential data;

reading the essential data;

generating syndromes corresponding to the read data; and

storing the generated syndromes.

33. The method of claim 32 wherein the act of using ECC techniques to check and correct the essential data comprises, when no longer refreshing the memory cells at the reduced power rate:

reading the essential data;

retrieving the stored syndromes;

using the stored syndromes to determine if any of the essential data are in error;

if any of the essential data were found to be in error, using the stored syndromes to provide corrected data; and

storing the corrected data in the memory cells.

34. The method of claim 33 wherein the DRAM is coupled to a processor, and wherein the act of reading the essential data comprises using the processor to read the essential data.

35. The method of claim 31 wherein the DRAM is coupled to a processor, and wherein the act of determining which memory cells are storing essential data that should be protected comprises using the processor to determine which memory cells are storing essential data that should be protected.

36. A processor-based system, comprising:

a memory controller;

a dynamic random access memory ("DRAM") device coupled to the memory controller, the DRAM device having a plurality of DRAM cells that are refreshed at a relatively high rate during operation in a normal mode and a relatively low rate during operation in a low power refresh mode; and

a processor coupled to the DRAM device through the memory controller, the processor being operable to couple a first signal to the DRAM device to cause the DRAM device to operate in the low power refresh mode and to couple a second signal to the DRAM device to cause the DRAM device to operate in the normal mode, the processor being operable prior to coupling the first signal to the DRAM device to:

determine which DRAM cells are storing data that should be protected from data retention errors in the low power refresh mode;

couple signals to the DRAM device that cause data to be read from the DRAM cells determined to be storing data that should be protected, the read data being used to generate ECC syndromes corresponding to the read data and being stored for subsequent use; and

the processor being operable after coupling the second signal to the DRAM device to couple signals to the DRAM device that cause data to be read from the DRAM cells that are storing data that should be protected, the read data being checked for errors and any errors corrected using the stored syndromes, the corrected data being stored in the DRAM device.

37. The system of claim 36 wherein the DRAM device comprises:

a syndrome memory that is operable to store the syndromes; and

ECC logic that is operable to:

receive the read data from the DRAM cells;

generate ECC syndromes corresponding to the read data;

cause the generated syndromes to be stored in the DRAM;

use the stored syndromes to check and correct read data from the DRAM cells; and

cause the corrected data to be stored in the DRAM.

38. The system of claim 37 wherein the DRAM device comprises a control register that receives a control bit from

the processor, the control register being operable to store the control bit and to enable the ECC logic responsive to the control bit being set.

39. The system of claim 38 wherein the control register further receives from the processor a plurality of bits identifying the DRAM cells that are storing data that should be protected from data retention errors.

40. The system of claim 37 wherein the DRAM device further comprises:

an ECC controller that is operable to controls the ECC logic; and

a mode register coupled to receive mode control signals from the processor, the mode control signals switching the DRAM device between the normal mode and the low power refresh mode, the mode control signals further enabling and disabling the ECC controller.

41. The system of claim 37 wherein the DRAM device further comprises data steering logic coupled to receive corrected data from the ECC logic, the data steering logic being operable to couple the corrected data back to the DRAM cells for storage in the DRAM device.

42. The system of claim 36 wherein the DRAM device comprises a syndrome memory that is operable to store the generated ECC syndromes corresponding to the read data.

43. The system of claim 36 wherein the memory controller comprises ECC logic that is operable to:

receive the read data from the DRAM cells;

generate ECC syndromes corresponding to the read data;

cause the generated syndromes to be stored in the DRAM;

use the stored syndromes to check and correct read data from the DRAM cells; and

cause the corrected data to be stored in the DRAM.

44. A computer system, comprising:

a memory controller;

a dynamic random access memory ("DRAM") device coupled to the memory controller, the DRAM device having at least one array of DRAM cells that are refreshed at a relatively high rate during operation in a normal mode and a relatively low rate during operation in a low power refresh mode; and

a processor coupled to the DRAM device through the memory controller, the processor being operable to identify at least one region of the array that should be protected from data loss when the DRAM device is operating in the low power refresh mode, the processor being operable to protect the identified region using ECC techniques during the period the DRAM device is operating in the low power refresh mode without protecting regions of the array other than the identified region.

45. The computer system of claim 44 wherein the DRAM device comprises ECC logic that is operable to generate syndromes corresponding to data stored in the identified regions of the array and to use the syndromes to check and correct the data stored in the identified regions of the array.

46. The computer system of claim 44 wherein the memory controller comprises ECC logic that is operable to generate syndromes corresponding to data stored in the identified regions of the array and to use the syndromes to check and correct the data stored in the identified regions of the array.

47. The system of claim 46 wherein the DRAM device comprises a syndrome memory that is operable to store the ECC syndromes generated by the memory controller.

* * * * *