

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

在自動刷新期間或通過使用隱藏的寫入命令同時針對具有內部錯誤校正碼(ECC)位元的 DRAM 進行控制器型記憶體清除 /

CONTROLLER-BASED MEMORY SCRUB FOR DRAMS WITH
INTERNAL ERROR-CORRECTING CODE (ECC) BITS

CONTEMPORANEOUSLY DURING AUTO REFRESH OR BY USING
MASKED WRITE COMMANDS

【相關申請案交叉參照】

【0001】 相關申請案

【0002】 本申請案也關於編號 14/963023，於 2015 年 12 月 8 日所提出，標題為「METHOD FOR SCRUBBING AND CORRECTING DRAM MEMORY DATA WITH INTERNAL ERROR-CORRECTING CODE (ECC) BITS CONTEMPORANEOUSLY DURING SELF-REFRESH STATE」，發明人為 David Reed 以及 Alok Gupta，並且代理人案號為 NVID-P-SC-14-0382-US1 的美國專利申請案。該申請案在此以引用方式完整併入本文中。

【0003】 本申請案也關於編號 14/963067，於 2015 年 12 月 8 日所提出，標題為「METHOD FOR MEMORY SCRUB OF DRAM WITH INTERNAL ERROR CORRECTING CODE (ECC) BITS DURING EITHER MEMORY ACTIVATE AND/OR PRECHARGE OPERATION」，發明人為 David Reed 以及 Alok Gupta，並且代理人案號為 NVID-P-SC-14-0382-US3 的美國專利申請案。該申請案在此以引用方式完整併入本文中。

【技術領域】

無

【背景技術】

【0004】 傳統上，在動態隨機存取記憶體(DRAM，Dynamic Random Access Memory)內，某些記憶體單元的小弱點或像是電磁或粒子輻射的外部

干擾，都可導致無法避免的隨機位元翻轉。該錯誤率通常隨著老化以及記憶體頻繁使用而增加。位元錯誤會導致系統當機，但是即使若位元錯誤未導致系統當機，還是會導致嚴重問題，因為錯誤會留在系統中，導致不正確的計算，並將錯誤本身乘到其他資料中。這尤其在某些應用中是個問題，例如金融、醫療、汽車等等。毀損的資料也會傳播至儲存媒體，並且成長至難以診斷與復原的程度。大部分 DRAM 錯誤都屬於暫態，並且在系統重新開機之後就會消失，而所造成的損壞則會留下。

【0005】 根據 DRAM 供應商，在伺服器與其他高可靠度環境中，目前將錯誤修正碼(ECC)整合到記憶體來保護記憶體。ECC 通常用於在易出錯或高可靠性系統中增強資料完整性。數十年來，工作站以及電腦伺服器平台藉由將額外 ECC 通道新增至其資料匯流排，使其資料完整性得到提昇。諸如家用電腦、平板電腦以及智慧型電話這些主流計算裝置都依賴商業 DRAM 的低基線位元錯誤率，並且不實施堅固或任何錯誤修正。當在這些裝置之一者內發生 DRAM 資料錯誤時，會導致靜默損毀或潛在裝置當機強迫重新開機。

【0006】 通常，ECC 添加與偵測及/或校正故障的數據一起儲存之校驗和。例如利用將該處理器的資料匯流排從 64 位元加寬到 72 位元，在每 64 位元字元內容納一個 8 位元校驗和，來實現此錯誤修正。該記憶體或系統處理器通常將配置邏輯，來產生 ECC 校驗和，並且使用這些校驗和，確認並修正從該記憶體讀取的資料。

【0007】 直到現在，DRAM 並未在 DRAM 裝置內部實施任何錯誤修正。DRAM ECC 總是利用增加資料(更多 DRAM 裝置)從外部執行，來建立較寬的通道。隨著處理節點收縮，尤其是在行動應用情況下，則每位元儲存的電荷越來越小，因此更容易受到內部與外部雜訊的影響。

【0008】 非揮發性記憶體具有比 DRAM 還要高的錯誤可能性，這些裝置在每區塊都加入大量額外位元，允許進行錯誤維修。然而，該維修本身發生在快閃記憶體控制器內，不在該快閃記憶體本身內。

【0009】 因此，隨著 DRAM 接近 20 nm 節點以及更小處理節點，DRAM 產業已經準備將 ECC 加入 DRAM 內。然而，隨著 DRAM 供應商朝

列。因此，對於要讓裝置有更長供電時間的系統，仍舊具有資料毀壞導致無法修正錯誤的風險。

【簡述】

【0010】 因此需要有一種方法及設備，以修復方法增加修正能力給 DRAM，該方法增加額外的彈性並為 DRAM 提供長期操作可靠性。

【0011】 傳統 DRAM 的問題在於，在讀取操作期間，內部 ECC 可偵測並修正錯誤，但是其也將該資料寫回該記憶體陣列，此行為導致在跨越多次存取時該錯誤仍舊留在該記憶體陣列內，並且可導致稍後該 DRAM 故障。

【0012】 本發明的具體實施例有利地利用清除該等錯誤來解決此問題。換言之，本發明的具體實施例將該修正過的資料寫回記憶體內適當列，並用修正過的資料更新該記憶體，取代僅讀出該修正版本。例如：在刷新或自我刷新循環期間，取代僅讀取來自選取列的該資料進入感應放大器並寫回該對應列，而是在將該資料寫回至該感應放大器之前，將該資料插入通過一 ECC 修正模組，並通過該感應放大器進入該記憶體陣列內的該對應列。因此，本發明的具體實施例利用讀取來自一已知儲體內該選取列的現有資料、根據該等 ECC 位元檢查並修正資料，並使用已修正的資料更新該記憶體儲體，來執行一「ECC 清除」。

【0013】 在一個具體實施例內，呈現更新一 DRAM 記憶體陣列的一方法，該方法包含：a) 根據來自一記憶體控制器的一命令，將該 DRAM 記憶體陣列轉換成一刷新模式；b) 在該 DRAM 記憶體陣列上執行一刷新循環；以及 c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有列上，執行該 DRAM 記憶體陣列內一選取欄的錯誤修正碼(ECC)清除。

【0014】 在不同的具體實施例內，呈現更新一 DRAM 記憶體陣列的一方法，該方法包含：a) 根據來自一記憶體控制器的一命令，將該 DRAM 記憶體陣列轉換成一刷新模式；b) 在該 DRAM 記憶體陣列上執行一刷新循環；以及 c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有列上，執行該 DRAM 記憶體陣列內複數個選取欄的錯誤修正碼(ECC)清除。在進一步具體實施例內，該方法也包含：d) 用該等複數個選取欄的數量遞增一刷

新欄計數器，將該 DRAM 記憶體陣列內用於清除的一後續欄定址。尚且在進一步具體實施例內，該方法也可包含針對有關該後續欄的一後續刷新循環，重複 a)、b)、c)和 d)。

【0015】 在一個具體實施例內，揭示用於更新一 DRAM 記憶體陣列之設備，該設備包含該 DRAM 記憶體陣列，其中該 DRAM 記憶體陣列設置成：a) 根據來自一記憶體控制器的一命令，轉換成一刷新模式；b) 執行一刷新循環；以及 c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有該等列上，執行該 DRAM 記憶體陣列內一選取欄的一 ECC 清除。

【0016】 以下詳細描述結合附圖，將對本發明的性質與優點有更佳瞭解。

【圖式簡單說明】

【0017】 本發明的具體實施例藉由範例進行說明並且不受其限制，在附圖中的數據以及其中相同的參考編號指示相同的元件。

【0018】 第一圖為根據本發明具體實施例的示範電腦系統。

【0019】 第二圖為例示傳統 DRAM 的一硬體方塊圖。

【0020】 第三圖為例示傳統 DRAM 操作的狀態圖。

【0021】 第四圖為根據本發明的具體實施例，例示可在一刷新或自我刷新循環期間，使用來自一 ECC 清除的已修正資料更新一記憶體陣列之 DRAM 的硬體方塊圖。

【0022】 第五圖為根據本發明的具體實施例，例示可在自我刷新模式內執行一 ECC 清除的一 DRAM 操作之狀態圖。

【0023】 第六圖為根據本發明的具體實施例，例示可在由該記憶體控制器起始的自我刷新模式內，執行一 ECC 清除的一 DRAM 操作之狀態圖。

【0024】 第七圖為根據本發明的具體實施例，例示可在一列啟動或預先充電操作期間，執行一 ECC 清除的一 DRAM 操作之狀態圖。

【0025】 第八圖根據本發明的具體實施例，顯示在自我刷新模式下，在一 DRAM 記憶體模組內實現一 ECC 清除的示範處理流程圖。

【0026】 第九圖根據本發明的具體實施例，顯示在由該記憶體控制器起始的刷新模式下，在一 DRAM 記憶體模組內實現一 ECC 清除之示範處

理流程圖。

【0027】 第十圖根據本發明的具體實施例，顯示在一列啟動操作期間，在一 DRAM 記憶體模組內實現一 ECC 清除的示範處理流程圖。

【0028】 第十一圖根據本發明的具體實施例，顯示在一列預先充電操作期間，在一 DRAM 記憶體模組內實現一 ECC 清除的示範處理流程圖。

【實施方式】

【0029】 實施方式

【0030】 在此將詳細參考本發明的許多具體實施例，附圖內將說明其範例。雖然本發明將結合這些具體實施例來說明，吾人將瞭解這並不用於將本發明限制在這些具體實施例上。相反地，本發明用於涵蓋申請專利範圍域與精神內所包含之變化、修改與同等配置。更進一步，在下列本發明的詳細說明中，將公佈許多特定細節以提供對本發明有通盤了解。不過，吾人了解在沒有這些特定細節的情況下也可實施本發明。在其他實例中，已知的方法、程序、組件和電路並未詳述，如此就不會模糊本具體實施例的域。

【0031】 某些詳細說明部分都以可在電腦記憶體上執行的資料位元上操作之程序、邏輯區塊、處理以及其他符號表示之方式來呈現。這些說明與代表為精通資料處理技術人士用來將其工作內容灌輸給其他精通此技術人士的最有效方式。在本申請案中，程序、邏輯區塊、處理等等一般係認為是導致所要結果的自治步驟或指令序列。這些步驟為所需的物理量之物理操縱。通常，雖然非必要，不過這些量採用可以儲存、傳輸、結合、比較以及在電腦系統內操縱的電或磁性信號形式。為了時間上方便起見，原則上因為常用，所以這些信號代表交易、位元、數值、元件、符號、字元、樣板、像素等等。

【0032】 不過，吾人應該瞭解，所有這些與類似詞彙都與適當的物理量相關連，並且僅為適用這些量的便利符號。除非特別聲明，否則從下列討論中可瞭解，整個本詳細說明當中，利用像是「轉換」、「執行」、「遞增」和「重複」等等詞彙所做的討論都是提及電腦系統或類似電子計算裝置或處理器(例如第一圖的系統 110)之動作與處理(像是第八圖的流程圖)。電腦

系統或類似電子計算裝置操縱並轉換呈現為電腦系統暫存器、記憶體或其他這種資料儲存、傳輸或顯示裝置內實體(電子)量的資料。

【0033】 本說明書內描述的具體實施例可在位於例如程式模組這類某些電腦可讀取儲存媒體形式上，由一或更多電腦或其他裝置所執行的電腦可執行指令之一般內容內討論。藉由範例並且不受限於此範例，電腦可讀取儲存媒體可包含非暫態電腦可讀取儲存媒體以及通訊媒體、包含暫態、傳播信號以外所有電腦可讀取媒體的非暫態電腦可讀取媒體。一般而言，程式模組包含常式、程式、物件、組件、資料結構以及執行特定工作或實施特定摘要資料類型等。程式模組的功能性可在許多具體實施例內結合或分散。

【0034】 電腦儲存媒體包含在任何方法或技術內實施的揮發與非揮發性、可移除與不可移除媒體，用於儲存像是電腦可讀取指令、資料結構、程式模組或其他資料等等資訊。電腦儲存媒體包含但不受限於隨機存取記憶體(RAM)、唯讀記憶體(ROM)、電可抹除式可程式編輯 ROM (EEPROM)、快閃記憶體或其他記憶體技術、小型碟片 ROM (CD-ROM)、數位視訊光碟(DVD)或其他光學儲存裝置、磁匣、磁帶、磁片儲存或其他磁性儲存裝置，或可用於儲存所要資訊以及可存取來擷取這些資訊的其他任何媒體。

【0035】 通訊媒體可具體實施電腦可執行指令、資料結構以及程式模組，並且包含任何資訊傳遞媒體。藉由範例並且不受限於此範例，通訊媒體包含像是有線網路或直接有線連線的有線媒體，以及像是聲音、射頻(RF)、紅外線以及其他無線媒體的無線媒體。上述任何媒體的通訊也應該包含在電腦可讀取媒體的域內。

【0036】 第一圖為可實施本發明明具體實施例的計算系統 110 之範例方塊圖。計算系統 110 廣泛代表可執行電腦可讀取指令的任何單一或多處理器計算裝置或系統。計算系統 110 的範例包含但不受限於工作站、膝上型電腦、用戶端終端機、伺服器、分散式計算系統、手持式裝置、遊戲系統、可變刷新率顯示系統或任何其他計算系統或裝置。在最基本的組態中，計算系統 110 可包含至少一處理器 114 以及一系統記憶體 116。

【0037】 處理器 114 一般代表可處理資料或解析並執行指令的任何

類型或型式之處理單元。例如：處理單元 114 可代表一中央處理單元(CPU, central processing unit)、一圖形處理單元(GPU, graphics processing unit)或這兩者。在一個具體實施例內，本發明的該 DC 不平衡偵測與 DC 不平衡矯正程序都程式編輯至 CPU (或 GPU) 114。在特定具體實施例內，處理器 114 可接收來自軟體應用程式或模組的指令。這些指令可導致處理器 114 執行本說明書所描述及/或例示的一或更多範例具體實施例之功能。

【0038】 系統記憶體 116 一般代表可儲存資料及/或其他電腦可讀取指令的任何類型或形式之揮發性或非揮發性儲存裝置或媒體。系統記憶體 116 的範例包含但不受限於 RAM、ROM、快閃記憶體或任何其他合適的記憶體裝置。雖然並未要求，不過在特定具體實施例內，計算系統 110 可包含揮發性記憶體單元(像是例如系統記憶體 116)以及非揮發性儲存裝置(像是例如主要儲存裝置 132)。在一個具體實施例內，系統記憶體 116 可包含合併本發明具體實施例的 DRAM 模組。

【0039】 除了處理器 114 和系統記憶體 116 以外，計算系統 110 也可包含一或多個組件或元件，例如在圖 1 的具體實施例內，計算系統 110 包含一記憶體控制器 118、一輸入/輸出(I/O)控制器 120 以及一通訊介面 122，這每一者都透過一通訊基礎設施 112 互連。通訊基礎設施 112 通常代表可促進計算裝置的一或更多組件之間通訊之任何類型或形式之基礎設施。通訊基礎設施 112 的範例包含但不限制為一通訊匯流排(像是工業標準架構 (ISA)、週邊組件互連(PCI)、PCI Express (PCIe)或類似匯流排)以及一網路。

【0040】 記憶體控制器 118 一般代表可操縱記憶體或資料或控制計算系統 110 的一或多個組件間之通訊的任何類型或形式之裝置，例如：記憶體控制器 118 可控制處理器 114、系統記憶體 116 以及 I/O 控制器 120 之間透過通訊基礎設施 112 的通訊。

【0041】 I/O 控制器 120 通常代表可協調及/或控制計算裝置的輸入與輸出功能之任何類型或形式之模組，例如：I/O 控制器 120 可控制或促進計算系統 110 的一或更多元件之間的資料傳輸，像是處理器 114、系統記憶體 116、通訊介面 122、顯示配接器 126、輸入介面 130 以及儲存介面 134。

【0042】 通訊介面 122 廣泛代表可促進範例計算系統 110 與一或更多

額外裝置之間通訊的任何類型或形式之通訊裝置或配接器，例如：通訊介面 122 可促進計算系統 110 與包含額外計算系統的私人或公用網路間之通訊。通訊介面 122 的範例包含但不受限於有線網路介面(像是網路介面卡)、無線網路介面(像是無線網路介面卡)、數據機以及任何其他合適的介面。在一個具體實施例內，通訊介面 122 透過直接連結至網路，例如網際網路，提供直接連線至遠端伺服器。通訊介面 122 也可透過任何其他合適的連線，間接提供連線。

【0043】 通訊介面 122 也可代表一主配接器，其設置成透過外部匯流排或通訊通道，促進計算系統 110 與一或更多額外網路或儲存裝置之間的通訊。主配接器的範例包含但不受限於小型電腦系統介面(SCSI)主配接器、萬用序列匯流排(USB)主配接器、IEEE (電氣與電子工程師協會) 1394 主配接器、序列先進技術附件(SATA)與外接 SATA (eSATA)主配接器、先進技術附件(ATA)與並列 ATA (PATA)主配接器、光纖通道介面配接器、乙太網路配接器等等。通訊介面 122 也可允許計算系統 110 參與分散式或遠端計算，例如：通訊介面 122 可接收來自遠端裝置的指令，或將指令傳送至遠端裝置來執行。

【0044】 如第一圖內所例示，計算系統 110 也可包含至少一個顯示裝置 124，例如透過一顯示配接器 126 連結至通訊基礎設施 112 的一可變刷新率顯示裝置。顯示裝置 124 一般代表可視覺顯示由顯示配接器 126 所轉送資訊的任何類型或形式之裝置。類似地，顯示配接器 126 通常代表設置成轉送圖形、文字與其他資料來顯示在顯示裝置 124 上的任何類型或形式之裝置。在一個具體實施例內，顯示裝置 124 可為具有可變刷新率的一 LCD 裝置。在一個具體實施例內，本發明的該 DC 不平衡偵測與 DC 不平衡矯正程序都程式編輯至顯示裝置 124 或顯示配接器 126 的韌體。因為該 DC 不平衡避免、偵測與矯正程序通常針對個別 LCD 系統定制，在較佳具體實施例內，該等程序將直接程式編輯至顯示裝置 124 或顯示配接器 126 的韌體。例如：該等程序可直接在顯示裝置 124 或顯示配接器 126 內一可程式處理器或固定功能專屬硬體內程式編輯。

【0045】 如第一圖內所例示，計算系統 110 也可包含至少一個輸入裝

置 128，其透過一輸入介面 130 連結至通訊基礎設施 112。輸入裝置 128 一般代表可提供不管是電腦或人類產生的輸入至計算系統 110 之任何類型或形式的輸入裝置。輸入裝置 128 的範例包含但不受限於鍵盤、指標裝置、語音辨識裝置或任何其他輸入裝置。

【0046】 如第一圖內所例示，計算系統 110 也可包含一主要儲存裝置 132 和一備用儲存裝置 133，其透過一儲存介面 134 連結至通訊基礎設施 112。儲存裝置 132 和 133 一般代表可儲存資料及/或其他電腦可讀取指令的任何類型或形式之儲存裝置或媒體，例如：儲存裝置 132 和 133 可為磁碟機(例如俗稱的硬碟)、軟碟機、磁帶機、光碟機、隨身碟等等。儲存介面 134 一般代表用於在儲存裝置 132 和 133 與計算裝置 110 的其他組件之間傳輸資料之任何類型或形式的介面或裝置。

【0047】 在一個範例中，資料庫 140 可儲存在主要儲存裝置 132 內。資料庫 140 可代表單一資料庫或計算裝置的部份，或可代表多個資料庫或計算裝置。例如：資料庫 140 可代表(儲存在)計算系統 110 的部分及/或第二圖內範例網路架構 200 (底下)的部分。另外，資料庫 140 可代表(儲存在)能夠由計算裝置，像是計算系統 110 及/或網路架構 200 的部份所存取之一或多個實際分隔裝置。

【0048】 繼續參閱第一圖，儲存裝置 132 和 133 可設置成從可移除式儲存單元中讀取及/或寫入至此，其中該儲存單元設置成儲存電腦軟體、資料或其他電腦可讀取資訊。合適的可移除式儲存單元之範例包含但不受限於軟碟、磁帶、光碟、隨身碟等等。儲存裝置 132 和 133 也可包含其他類似結構或裝置，允許電腦軟體、資料或其他電腦可讀取指令載入計算系統 110。例如：儲存裝置 132 和 133 可設置成讀取與寫入軟體、資料或其他電腦可讀取資訊。儲存裝置 132 和 133 也可為計算系統 110 的一部分，或可為透過其他介面系統存取的個別裝置。

【0049】 許多其他裝置或子系統都可連接至計算系統 110。相對地，並非第一圖內所例示的所有組件與裝置都需要存在才能實踐本文所述的具體實施例。上面引用的裝置與子系統也可用與第一圖內所示不同的方式互連。計算系統 110 也可運用任何數量的軟體、韌體及/或硬體組態，例如：

本文內揭示的該範例具體實施例可編碼為電腦可讀取媒體上的電腦程式(也稱為電腦軟體、軟體應用程式、電腦可讀取指令或電腦控制邏輯)。

【0050】 內含該電腦程式的該電腦可讀取媒體可載入計算系統 110。然後儲存在該電腦可讀取媒體內的所有或部分該電腦程式可儲存在系統記憶體 116 及/或儲存裝置 132 和 133 的許多部分內。由處理器 114 執行時，已經載入計算系統 110 的一電腦程式可導致處理器 114 執行及/或為執行本說明書內所描述及/或例示的示範具體實施例功能之裝置。此外或另外，本說明書內描述及/或例示的該等範例具體實施例可用軟體及/或硬體實施。

【0051】 例如：用於追蹤並矯正 DC 不平衡的電腦程式可儲存在該電腦可讀取媒體內，然後可儲存在系統記憶體 116 及/或儲存裝置 132 和 133 的許多部分內。當由處理器 114 執行時，該電腦程式可導致處理器 114 執行及/或成為一種手段，用來執行用來進行上面所討論 DC 不平衡避免、偵測與矯正所需之功能。

【0052】 在自動刷新期間或通過使用隱藏的寫入命令同時針對具有內部錯誤校正碼(ECC)位元的 DRAM 進行控制器型記憶體清除

【0053】 本發明的具體實施例提供方法及設備，以修復方法增加修正能力給 DRAM，該方法增加額外的彈性並為 DRAM 提供長期操作可靠性。

【0054】 DRAM 為一種隨機存取記憶體，將每一資料位元儲存在積體電路之內個別電容器內，該電容器可充電或不充電，指示 1 或 0 之值。因為 DRAM 單元上的電荷會由於包含溫度的一些因素而隨時間變弱，該等電容器將緩慢放電，並且該資訊最終會消失。

【0055】 為了避免發生此情況，需要提昇每一單獨記憶體單元內含的電荷，定期刷新 DRAM。發生刷新所需的頻率取決於用來製造該核心記憶體晶粒的矽技術，以及該記憶體單元本身的設計。為了避免資料損失，該 DRAM 記憶體的所有儲體都必須使用一系列刷新操作，在指定刷新週期之內刷新。

【0056】 第二圖為例示傳統 DRAM 的一硬體方塊圖。位址 291 已接收並鎖存至位址鎖存器 231，而命令 291 則由命令解碼器和邏輯模組 232 接

收並解碼。該列位址已鎖存至該列位址鎖存器和多工器模組 233，而該欄位址已鎖存至欄位址鎖存器模組 271。

【0057】 為了減少存取延遲，DRAM 分成多個大小相等的單元，稱為儲體。大多數 DRAM 晶片舉有一多儲體架構，並且可組織成儲體、列與欄。儲體選擇邏輯模組 239 使用該位址資訊從儲體[0:n] 235 當中選擇適當儲體，來存取。例如典型 512 MB SDRAM 晶片包含四個獨立的 16 MB 記憶體儲體，儲體內每列都為 16,384 位元的陣列。一儲體可為待命、啟動或從一個狀態改變成另一個狀態。

【0058】 使用列解碼器 234，該列位址套用至從儲體[0:n]當中選取的儲體。該列位址解碼器選擇要傳送至感應放大器 237 的適當列。「啟動」命令啟動一待命儲體，例如：其可呈現一二位元儲體位址以及一 13 位元列位址，並導致將該列讀入 16,384 欄感應放大器 237。這也稱為「開放」該列。

【0059】 感應放大器 237 也就是已知的「列緩衝區」，並提供存取至在該儲體上已經開啟的該列。在可讀取記憶體位置之前，包含該記憶體位置的整列都開啟並讀入該列緩衝區。該頁(列)停留在該列緩衝區內，直到該頁(列)明確關閉為止。若對於該開啟頁的存取到達該儲體，則可從單一記憶體循環之內的該列緩衝區立即進行。若對另一列的存取到達，則目前的列必須關閉並且新列必須在實施該要求之前開啟。

【0060】 一旦該列已經啟動，則「讀取」與「寫入」命令可至該列。讀取與寫入命令都需要一欄位址，該欄位址使用欄解碼器 280 提供給 I/O 選擇模組 238。針對讀取操作，然後讀取要求的資料，並在該輸出資料放到 DQ 線 290 之前，使用讀取資料/ECC 邏輯模組 236 修正錯誤。針對寫入操作，已計算 ECC 並且將寫入資料和已計算的 ECC 寫入至該選取欄位址上的該列緩衝區。

【0061】 請注意，雖然 ECC 模組 236 在一讀取操作期間輸出該資料之前已經錯誤修正該資料，不過該修正資料絕不會寫回至感應放大器，或至從選取儲體 235 讀出的該對應列。這是個問題，因為若該 ECC 邏輯偵測並修正一錯誤，就不會將該資料寫回該列緩衝區或 DRAM 儲體 235。因此，若稍後在該列內有更多位元失敗，則該現有與後續錯誤則不受偵測。無法

維修壞掉的位元是個問題，因為若每字元有更多位元失敗，就會造成錯誤傳播通過整個記憶體系統，導致該 DRAM 失敗。

【0062】 寫入命令伴隨有要寫入驅動到 DQ 線 290 上的資料。該記憶體控制器的責任就是確定並未驅動 SDRAM 讀取資料至該等 DQ 線，在此同時需要驅動寫入資料至這些線。透過使用寫入資料模組 241 的 IO 選擇模組 238，將該資料寫入至感應放大器 237。同樣，欄解碼器 280 用來選擇可寫入資料的適當欄。在寫入至特定單元期間，同時感應到一系列內的所有欄，就如同讀取期間，如此雖然只有單一欄的儲存單元電容器電荷可能改變，不過整個列都已刷新(寫回)。ECC 邏輯 240 用來計算選取該欄並且將資料寫入列緩衝區 237 之前之該校驗和。

【0063】 如先前所提及，由於許多會影響漏電率的因素，包含溫度，該 DRAM 記憶體單元上的電荷將隨時間經過自然流失。儲存電荷的顯著減少可導致資料遺失或毀損，為了避免發生此情況，需要提昇每一單獨記憶體單元內含的電荷，定期刷新 DRAM。一般而言，製造商指定每列的儲存單元電容器必須每 64ms 或更短時間刷新。

【0064】 讀取或寫入至記憶體單元具有與發出刷新(REF)命令來刷新選取單元相同的效果。因為在正常操作期間不是所有單元都被讀取或寫入，所以必須在刷新間隔期滿之前存取和寫回(恢復)陣列中的每個單元。在大多數情況下，刷新循環恢復整個頁面的電荷。在整個間隔的過程中，存取每個頁面並接著恢復。在間隔的結尾上，再次開始該處理。

【0065】 DRAM 通常也包含一刷新列計數器 244，保持追蹤已經刷新的最後列 – 此列計數器用來決定下一個必須刷新的列。在套用該刷新(REF)命令之前，一儲體必須待命最短週期，該刷新命令由該記憶體控制器內的電路所處置。刷新計數器 244 通常包含要刷新的該列之位址，其要套用至晶片的列位址線，以及一計時器，將該計數器遞增以逐步通過該等列。當已經完成刷新，對應的儲體留在待命狀態下。某些控制器可使用一「全部刷新」命令，同時刷新該 DRAM 內所有儲體。

【0066】 當未主動使用一 DRAM 時，其將進入一低電源模式，在此期間該 DRAM 內部執行一定期刷新來維持資料完整度，這也就是自我刷

新。當 DRAM 在自我刷新狀態下，該記憶體控制器不會啟動明確刷新。通常使用自我刷新邏輯模組 243 結合該 DRAM 內部刷新列計數器 244，持續追蹤已經刷新的該等列。

【0067】 第三圖為例示傳統 DRAM 操作的狀態圖。在該 DRAM 準備回應讀取與寫入命令之前，必須先開啟(啟動)一儲體。該記憶體控制器利用傳送該適當命令(ACT)，指定要存取的等級、儲體和頁面(列)。「啟動」命令啟動待命的儲體，並將該記憶體從待命狀態 310 轉換成啟動狀態 320。在狀態 340 上，該儲體啟動並且該開啟儲體在該感應放大器之內包含一完整記憶體頁面，長度大約是 1-8 KB。

【0068】 在該啟動狀態下，該記憶體控制器可發出一每儲體刷新命令，導致該 DRAM 轉換成狀態 395。吾人應該注意，一每儲體刷新命令只能用於特定種類的 DRAM，例如 LPDDR (低功率 DDR) DRAM。若該 DRAM 支援每儲體刷新的增強模式，該每儲體刷新模式可刷新該儲體位準上的單元。這可讓一儲體要存取，而該 DRAM 內另一個儲體已經刷新，例如：多個儲體內的列可在狀態 395 內以循環方式刷新，而不會停止 DRAM 存取。

【0069】 每列每次只能開啟一頁。存取相同儲體內其他頁需要先關閉已開啟的頁面。一旦該頁面維持開啟，該記憶體控制器可發出讀取或寫入命令的任意組合，直到不再需要該開啟頁面之時，或對於來自同一儲體中另一頁面的讀取/寫入資料之未決請求需要關閉當前頁面，以便可以存取另一個頁面。這可利用發出一預充電(PRE)命令來只關閉特定儲體，或發出一預充電全部(PREA)命令來關閉裝置內所有已開啟儲體來達成。

【0070】 在接收來自該記憶體控制器的一預充電命令時，該記憶體從狀態 340 轉換成預充電模式 350。預充電準備該等資料線以及感應電路，將該等感應放大器內該已儲存的電荷傳輸回個別記憶體單元的該開啟頁面，復原先前損毀的讀取，讓該 DRAM 核心準備對要存取的下一個記憶體頁面取樣。

【0071】 在預充電之後，一儲體轉換為待命狀態 310。從待命模式中，該記憶體控制器可在狀態 370 內命令該記憶體執行每儲體刷新，或在狀態 380 內命令該記憶體執行全部儲體刷新。如上述，若該 DRAM 有一段

時間未使用，則可切換成一低功率自我刷新模式 345，在此期間該 DRAM 內部執行定期刷新來維持資料完整度。

【0072】 第四圖為根據本發明的具體實施例，例示可在一刷新或自我刷新循環期間，使用來自一 ECC 清除的已修正資料更新一記憶體陣列之 DRAM 的硬體方塊圖。

【0073】 如上述，傳統 DRAM 的問題為 ECC 模組 236 通常可偵測並修正錯誤，但是無法將該資料寫回該等感應放大器，或寫回已經從儲體 235 讀出的該對應列。這導致錯誤仍舊存在，並且未修正永久留在該 DRAM 儲存陣列之內。因此，若隨著時間經過在一系列內有很多位元失敗，則無法偵測到該錯誤累積，並且可能導致 DRAM 失效。

【0074】 本發明的具體實施例有利地利用清除該等錯誤來解決此問題。換言之，本發明的具體實施例將該修正過的資料寫回記憶體內適當列，並用修正過的資料更新該記憶體，取代僅讀出該修正版本。例如：在刷新或自我刷新循環期間，取代僅讀取來自選取列的該資料進入感應放大器 437 並寫回該對應列，而是在將該資料寫回至感應放大器 437 之前，使用 ECC 邏輯模組 440 將該資料通過一 I/O 選擇模組 438 和 ECC 修正，並通過該感應放大器進入該對應列。因此，本發明的具體實施例利用讀取來自一已知儲體內該選取列的現有資料、根據該等 ECC 位元檢查並修正資料，並使用已修正的資料更新該記憶體儲體，來執行一 ECC 清除。

【0075】 在一個具體實施例內，取代讓該資料通過 IO 選擇模組 438 與使用 ECC 邏輯模組 440，不同模組可設置成利用讀取來自該已知儲體內該選取列的現有資料來執行一 ECC 清除、檢查並根據該等 ECC 位元修正資料以及使用已修正資料更新該記憶體儲體。

【0076】 在一個具體實施例內，該 DRAM 將包含一額外組件，如第四圖內所示的刷新欄計數器 431。刷新欄計數器 431 可用來同時執行一系列內一或多欄的 ECC 清除，例如在一個具體實施例內，刷新循環期間只有單一欄要清除，而在不同具體實施例內，則可清除多個或全部欄。

【0077】 針對 LPDDR4，DRAM 供應商計畫在兩路徑上實現 ECC。在寫入期間，從完全未隱藏的資料突發或已遮蔽資料突發與該儲體/列/欄的

當前內容之組合，可建立 ECC 位元。在讀取期間，從該 DRAM 陣列內儲存的該資料和該 ECC 產生已修正資料。

【0078】 如上述，本發明的具體實施例利用將該已修正版本寫回記憶體，有利地「清除」該等錯誤，如此降低一錯誤留在該 DRAM 內的可能性。換言之，本發明的具體實施例實現一背景修復機構，在自我刷新與全部儲體刷新模式期間，將該現有的 ECC 區塊重複用在讀取/寫入操作內。若該 DRAM ECC 實現為依照儲體，則本發明的具體實施例也可在每儲體刷新模式期間執行清除。

【0079】 因此本發明的具體實施例特別有利於多年連續供電、其中大部份時間都在待命狀態下的系統內使用的 DRAM。該等 DRAM 例如在自我刷新模式內執行 ECC 清除，並降低錯誤隨時間累積的機會。在無本發明的具體實施例之下，SOC 設計師要在該 SOC 內實現一專屬清除器，並消耗額外功率喚醒該 DRAM 並明確定期清除整個陣列。

【0080】 第五圖為根據本發明的具體實施例，例示可在自我刷新模式內執行一 ECC 清除的一 DRAM 操作之狀態圖。如上述，若該 DRAM 有一段時間未使用，控制器可將該 DRAM 切換成一低功率自我刷新模式 545，在此期間該 DRAM 內部執行定期刷新來維持資料完整度。

【0081】 一儲體刷新等同於使用內部列計數器的啟動與預充電操作，不含讀取或寫入。在自我刷新循環 545 期間，本發明的具體實施例執行一虛擬讀取與寫入操作，其通過該 ECC 維修/生產邏輯重複循環來自該陣列的資料，並在該不明確預充電操作之前存回該陣列。該虛擬讀取與寫入操作以及該隨附 ECC 清除都在狀態 560 內執行，如第五圖內所示。刷新欄計數器 431 可用來保持追蹤已經清除的該等欄，在任何已知自我刷新循環期間，可清除一或多個欄。換言之，在自我刷新循環期間，該 DRAM 將自動清除全部列之上一或多個欄。此操作將根據該欄與列的隨附錯誤修正資料，自動修正已知欄內的任何位元錯誤。在下一個自我刷新循環上，更新該下一欄直到所有欄都完成。該刷新欄計數器相應遞增(根據刷新的欄數)，以維持並保持追蹤在後續自我刷新循環期間要更新的該欄。根據清除間隔需求，在何時遞增以及遞增多少該欄計數器上實施不同演算法。如上述，

當該 DRAM 待命時會進入自我刷新模式，如此該記憶體控制器不會涉入該刷新處理或發生於狀態 560 內的伴隨 ECC 清除。

【0082】 在自我刷新模式內可執行一 ECC 清除的 DRAM 額外硬體為一刷新欄計數器 431，從每通過所有列與儲體的完整循環就遞增一次之該啟動列中選擇一欄。換言之，該刷新欄計數器會在通過所有列與儲體的循環結尾上加一，如此可選擇用於 ECC 清除的下一欄。在於一已知循環內清除多欄的實例中，該欄計數器以遞增超過一的方式遞增。在只有單一 ECC 邏輯路徑而非每儲體都有的一個具體實施例內，也需要儲體計數器，其中該儲體計數器在每次通過所有列的完整循環之後加一。

【0083】 第六圖為根據本發明的具體實施例，例示可在由該記憶體控制器起始的自我刷新模式內，執行一 ECC 清除的一 DRAM 操作之狀態圖。在一個具體實施例內，該記憶體控制器可經過程式編輯，如此該控制器在一常規刷新循環期間定期執行該 DRAM 的 ECC 清除。類似於上面討論的自我刷新具體實施例，針對每欄執行該 ECC 清除，並且重複用於橫跨所有列的所有欄，來修正位元錯誤。換言之，該控制器定期指示該 DRAM，在一刷新循環期間跨越一已知欄的所有列執行一清除操作。然後接著，該控制器對下一欄執行相同動作，並重複此程序直到針對所有列與欄已更新整個記憶體。類似自我刷新具體實施例，DRAM 內部可實現一欄計數器。在其他具體實施例內，該 DRAM 協定可更新，並且控制器可提供含刷新命令的明確欄位址。

【0084】 如上面所討論，需要提昇每一單獨記憶體單元內含的電荷，定期刷新 DRAM。典型的 DRAM 記憶體需要每 16-64ms 刷新一次。在一個具體實施例內，一記憶體控制器可起始一刷新循環，例如第六圖內所示，從待命模式 610 中，該記憶體控制器可在狀態 670 內命令該 DRAM 執行每儲體刷新，或在狀態 685 內執行全部儲體刷新。在狀態 690 內每儲體刷新期間可執行一 ECC 清除，或可在狀態 680 內全部儲體刷新期間內執行。例如：若一刷新循環每 64 m 執行一次並且一列內有 64 欄，則可在 4,096 ms (64 x 64)內清除記憶體的全部儲體。請注意，針對要在所有儲體刷新狀態 685 內執行的清除，需要額外 ECC 邏輯，以便調節多個儲體同時清除。

【0085】 在一個具體實施例內，也可在儲體啟動狀態 640 期間執行一 ECC 清除 630，例如：若列 0 啟動，則在未啟動儲體的一列上可用一 ECC 清除 630 執行每儲體刷新 695。

【0086】 一記憶體控制器起始 ECC 刷除工作，類似於在自我刷新模式期間執行的清除，然而，由該記憶體控制器起始該刷新，取代在自我刷新狀態內使用內部邏輯由該 DRAM 自我啟動。如上面所解釋，在該待命狀態期間由該 DRAM 內部管理自我刷新模式，並且不是由該記憶體控制器所啟動。然而，類似於上面討論的該自我刷新具體實施例，讓一 ECC 清除可在記憶體控制器起始的刷新循環期間執行所需之該唯一額外硬體為刷新欄計數器 431，選擇性為一儲體計數器。

【0087】 此外，類似於自我刷新模式內的一 ECC 清除，在一控制器起始的刷新循環期間，本發明的具體實施例執行一虛擬讀取與寫入操作，其通過該 ECC 維修/生產邏輯重複循環來自該陣列的資料，並在該不明確預充電操作之前存回該陣列。該 ECC 清除在步驟 690、680 或 630 內執行，如第六圖內所示。刷新欄計數器 431 可用來保持追蹤在已知循環內已經清除的該欄(或該等欄)。

【0088】 在此具體實施例內，該刷新欄計數器可受該記憶體控制器的控制。在一刷新循環期間，可清除一或多個欄。該記憶體控制器使用一內部管理欄計數器，類似於刷新欄計數器 431，發出一刷新命令。其使用該等刷新命令，將此欄計數器的值提供給該 DRAM，強迫對一選取欄進行清除循環。

【0089】 在一個具體實施例內，執行該重複循環讀取/寫入操作的該硬體應該已經存在於 DRAM 設計中，因為其為所有位元組啟用都已停止的該隱藏寫入(讀取-修改-寫入)操作的子集。該虛擬讀取與寫入將有效做為一清除器，來修正位元錯誤並且降低一不可修正錯誤持續並且提高未來因為錯誤累積造成故障的機率之可能性。

【0090】 在一隱藏寫入或部分寫入操作期間，首先從選取欄讀取資料，並使用儲存的 ECC 位元修正任何錯誤，接著與傳入的部分寫入資料合併。此操作有時也稱為讀取-修改-寫入。在將此資料寫回 DRAM 陣列之前，

在該合併的資料上重新計算 ECC。在一個具體實施例內，控制器可在所有位元組啟用都停止之下發出隱藏寫入(有效的一空部分寫入命令)，有效使用內建機構來執行該記憶體的清除操作。

【0091】 在此方案中，該控制器發出一空部分寫入至記憶體，強迫讀取該啟動列內一欄並且檢查 ECC。若偵測到錯誤，則修正該資料。接著，重新計算該 ECC，並且將該已修正資料寫回該記憶體。在清除一欄之後，在該控制器內的該欄計數器遞增，並且在下一個循環期間，清除該後續欄。重複此操作，直到清除所有欄。在此方式中，該記憶體針對位元錯誤自動更新。在此具體實施例內，由於清除該記憶體控制器內 DRAM 的該邏輯，因此任何 DRAM 都可現成使用。

【0092】 在一個具體實施例內，使用該記憶體控制器所起始的一隱藏寫入操作(所有位元組啟用都停止的一讀取-修改-寫入命令)，執行該 ECC 清除操作。該記憶體控制器在啟動一已知列之後，將空部分寫入操作強迫至該選取欄，以便清除該選取欄內整個資料塊。針對每一欄重複此處理，直到清除所有欄。在不同的具體實施例內，由該控制器傳輸一專屬欄命令取代使用隱藏寫入操作，以便起始該 ECC 清除。

【0093】 因此，第五圖和第六圖內討論的具體實施例可用來運用保證發生的該刷新循環，或利用將專屬隱藏寫入循環插入該 DRAM 流量中，安靜清除該 DRAM 陣列。

【0094】 第七圖為根據本發明的具體實施例，例示可在一列啟動或預先充電操作期間，執行一 ECC 清除的一 DRAM 操作之狀態圖。在本發明的一個具體實施例內，錯誤修正電路與邏輯可加入該感應放大器，如此在列啟動或預充電操作期間，可檢查所有列資料(通過所有欄或其子集)是否有位元錯誤。在此情況下，可修正該等錯誤並寫回該列，來修正通過該列的所有資料。隨著時間推移，隨著整個記憶體使用，將檢查和修正位元錯誤。一控制器使用現有的 DRAM 協定或定義一組新的命令給此操作，來起始此操作。

【0095】 在此具體實施例內，在記憶體控制器控制之下，執行直接及/或強迫清除。在一個具體實施例內，對啟動與預充電循環建立一標記擴展，

其中該記憶體控制器提供該儲體與列位址，但是該 DRAM 內部提供該欄位址。當該再循環修復操作為暗示，則在該命令匯流排上不會發出讀取或寫入命令(底下有進一步解釋)。因此，不會消耗資料匯流排上的頻寬。

【0096】 雖然關於第五圖和第六圖所討論的具體實施例可用對於該 DRAM 硬體設計有最少或無改變之方式執行，但是第七圖的具體實施例則要求利用新增錯誤修正電路與邏輯至該感應放大器，來修改該 DRAM 設計。

【0097】 如第七圖內所示，在一個具體實施例內，該記憶體控制器可將一啟動清除命令傳送至該 DRAM。該啟動清除命令使用新加入的每儲體 ECC 邏輯，在已啟動進入該感應放大器的該列上執行 ECC 清除。因此，在狀態 730 內執行該清除，這屬於轉換至啟動狀態 720 與儲體啟動狀態 740 的一部分。在一個具體實施例內，讀取整列並且檢查該列的該 ECC。若偵測到錯誤，則修正該資料。接著，重新計算該 ECC，並且將該已修正資料寫入該感應放大器。然後，在預充電狀態 750 期間，將該已修正資料更新至該記憶體。因此，在狀態 740 內該啟動列上執行讀取或寫入操作之前，對該列進行錯誤修正。

【0098】 然而若要同時對整列執行一 ECC 清除，相較於先前具體實施例，此具體實施例需要多個 ECC 引擎。因此，此具體實施例可需要修改該硬體內含額外 ECC 引擎。在一個具體實施例內，該列可細分並且可在多個循環之上(例如 2 或 4 循環)執行該列的清除。進一步，取代一刷新欄計數器 431，此具體實施例需要一內部欄計數器可循環通過所有欄位址，並在寫回該資料之前，使用該 ECC 邏輯來修正任何錯誤。

【0099】 在一個具體實施例內，該控制器可命令該 DRAM 來執行一預充電清除。在此具體實施例內，取代在將該列讀入該感應放大器期間執行該 ECC 清除，而是在預充電期間從該感應放大器將該列寫回該記憶體時執行該 ECC 清除。如第七圖內所示，在該儲體轉換成待命狀態之前，於預充電 750 期間在狀態 760 內執行該 ECC 清除。檢查該列的 ECC 並且若偵測到錯誤，則使用該 ECC 位元修正該資料。接著，重新計算該 ECC，並且從該感應放大器將該已修正資料寫回記憶體，這屬於該預充電處理一部分。

【0100】 在一個具體實施例內，該控制器可程式編輯，如此每次有啟

動或預充電時執行清除。然而這導致較佳的位元錯誤率，這會產生更多延遲以及功率效能問題。另外，該控制器可用命令程式編輯，在定期啟動或預充電狀態期間執行清除，但並非全部。如此提供良好效率與位元率的妥協。對於效率敏感的系統可排定清除頻率較低，而另一個則可決定提高清除率，以降低錯誤的可能性。

【0101】 第八圖根據本發明的具體實施例，顯示在自我刷新模式下，在一 DRAM 記憶體模組內實現一 ECC 清除的示範處理流程圖。

【0102】 在步驟 802 上，該 DRAM 記憶體從該待命狀態轉換至低功耗自我刷新模式 545。在步驟 804 上，在該 DRAM 上執行一自我刷新模式。

【0103】 在步驟 806 上，於該自我刷新循環期間，在該自我刷新循環期間，於該 DRAM 記憶體儲體所有列之上一 DRAM 記憶體儲體一欄上執行 ECC 清除，如上面所討論。在不同的具體實施例內，可在該自我刷新狀態期間超過單一欄之上執行 ECC 清除。

【0104】 在步驟 808 上，將該刷新欄計數器遞增一。另外，若在一循環期間已清除多欄，則該刷新欄計數器相應遞增。

【0105】 在步驟 810 上，針對下一自我刷新循環期間後續欄再次執行該 ECC 清除，並且該計數器相應遞增。

【0106】 第九圖根據本發明的具體實施例，顯示在由該記憶體控制器起始的刷新模式下，在一 DRAM 記憶體模組內實現一 ECC 清除之示範處理流程圖。

【0107】 在步驟 902 上，根據來自該記憶體控制器的指令，該 DRAM 記憶體轉換成一刷新模式(例如每儲體刷新 670 和 695 或全部儲體刷新 685)。在步驟 904 上，在該 DRAM 模組上執行一刷新循環。

【0108】 在步驟 906 上，於該刷新循環期間，於該 DRAM 記憶體儲體所有列之上一 DRAM 記憶體儲體一欄上執行 ECC 清除，如上面討論的。在不同的具體實施例內，可在該刷新循環期間超過單一欄之上執行 ECC 清除。如上面所討論，在一個具體實施例內，由該記憶體控制器發出明確隱藏寫入命令，可執行該 ECC 清除。

【0109】 在步驟 908 上，將該刷新欄計數器遞增一。另外，若在一循

環期間已清除多欄，則該刷新欄計數器相應遞增。

【0110】 在步驟 910 上，針對下一刷新循環期間後續欄再次執行該 ECC 清除，並且該計數器相應遞增。

【0111】 第十圖根據本發明的具體實施例，顯示在一列啟動操作期間，在一 DRAM 記憶體模組內實現一 ECC 清除的示範處理流程圖。

【0112】 在步驟 1004 上，從記憶體控制器接收一命令來啟動一記憶體列。

【0113】 在步驟 1006 上，根據該啟動命令，在該啟動處理期間於該記憶體列上執行一 ECC 清除。

【0114】 在步驟 1008 上，在預充電模式期間，將該 ECC 清除之後已修正的資料從該感應放大器寫回該記憶體。

【0115】 第十一圖根據本發明的具體實施例，顯示在一列預先充電操作期間，實現一 ECC 清除的示範處理流程圖。

【0116】 在步驟 1102 上，從該記憶體控制器接收一命令來啟動一記憶體列。在步驟 1104 上，該列已啟動並且來自該列的資料已寫入該感應放大器。

【0117】 在步驟 1106 上，從該記憶體控制器接收一命令來將該記憶體列預充電。

【0118】 在步驟 1108 上，根據該預充電命令，在將該已修正資料寫回該記憶體之前，於該記憶體列上執行一 ECC 清除。

【0119】 雖然上面使用特定方塊圖、流程圖以及範例揭示許多具體實施例，不過本說明書中描述及/或例示的每一方塊圖元件、流程圖步驟及/或元件，都可使用廣泛的硬體、軟體或韌體(或任何組合)組態，獨自及/或集中實施。此外，因為可實施許多其他架構來達成相同功能性，所以其他元件內含元件的任何揭露都應該考量當成範例。

【0120】 本說明書中描述及/或例示的處理參數以及步驟順序僅為範例，例如：雖然本說明書中例示及/或描述的該等步驟可用特定順序顯示或討論，不過這些步驟並不需要以例示或討論的順序來執行。本說明書中描述及/或例示的許多範例方法也可省略本說明書中描述及/或例示的一或更

多步驟，或加額外步驟。

【0121】 雖然本說明書中已經在全功能計算系統範圍內描述及/或例示許多具體實施例，一或更多這些範例具體實施例可用許多形式分布當成一程式產品，而不管實際上用於執行該分布的電腦可讀取媒體之特定類型。本說明書中揭示的具體實施例也可使用執行特定任務的軟體模組來實施。這些軟體模組可包含描述檔、批次檔或可儲存在一電腦可讀取儲存媒體或計算系統內的其他可執行檔。這些軟體模組可設置一計算系統，執行本說明書所揭示的一或更多範例具體實施例。本說明書中揭示的一或更多軟體模組可在雲端計算環境下實施。雲端計算環境可透過網際網路提供許多服務與應用程式。透過網路瀏覽器或其他遠端介面，可存取這些雲端服務(例如服務軟體、服務平台、服務基礎設施等等)。本說明書中描述的許多功能可透過遠端桌面環境或任何其他雲端計算環境來提供。

【0122】 在上面的說明中，為了解釋而參考特定具體實施例做說明。不過，上面例示的討論並非用於專屬或限制本發明於所說明的形式中。許多修改與變化都可以上述為依據。具體實施例經過選擇與說明來最佳闡述本發明原理，並且以許多具體實施例讓其他精通技術人士對本系統有最佳了解，這些具體實施例都適合特定使用期待。

【0123】 到此已經說明根據本發明的具體實施例。雖然已經用特定具體實施例說明本發明，吾人應該瞭解，本發明不應受限於這種具體實施例，而是根據下列申請專利範圍來建構。

【符號說明】

110	計算系統	126	顯示配接器
112	通訊基礎設施	128	輸入裝置
114	處理器	130	輸入介面
116	系統記憶體	132	主要儲存裝置
118	記憶體控制器	133	備用儲存裝置
120	輸入/輸出控制器	134	儲存介面
122	通訊介面	140	資料庫
124	顯示裝置	200	網路架構

I643200

發明摘要

※ 申請案號：105140632

※ 申請日：105/12/08

※IPC 分類：G11C 29/42 (2006.01)

【發明名稱】(中文/英文)

在自動刷新期間或通過使用隱藏的寫入命令同時針對具有內部錯誤校正碼 (ECC) 位元的 DRAM 進行控制器型記憶體清除 /
CONTROLLER-BASED MEMORY SCRUB FOR DRAMS WITH
INTERNAL ERROR-CORRECTING CODE (ECC) BITS
CONTEMPORANEOUSLY DURING AUTO REFRESH OR BY USING
MASKED WRITE COMMANDS

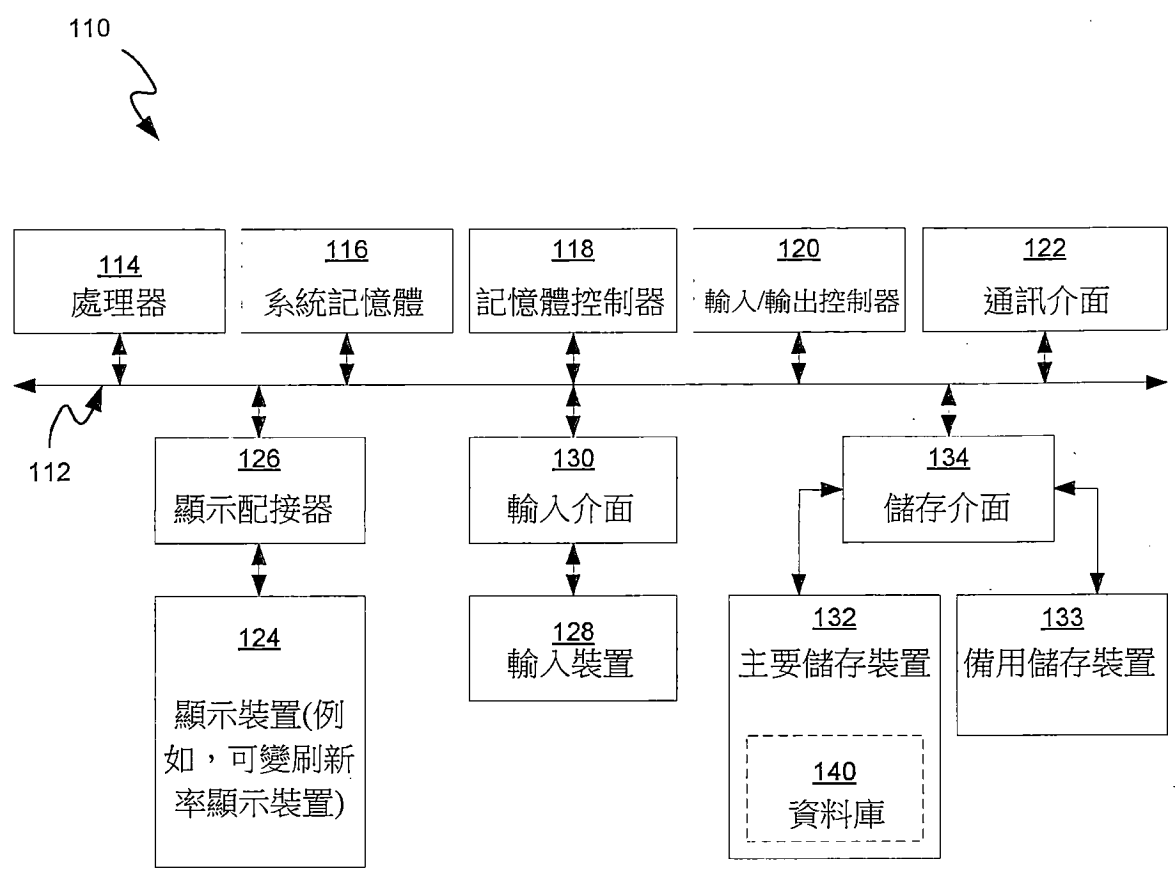
【中文】

本發明揭示一種用於更新一 DRAM 記憶體陣列的方法，該方法包含：
a) 根據來自一記憶體控制器的一命令，將該 DRAM 記憶體陣列轉換成一刷新模式；b) 在該 DRAM 記憶體陣列上執行一刷新循環；c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有列上，執行該 DRAM 記憶體陣列內一選取欄的錯誤修正碼(ECC)清除；以及 d) 將一刷新欄計數器遞增，選擇該 DRAM 記憶體陣列內的一後續欄。

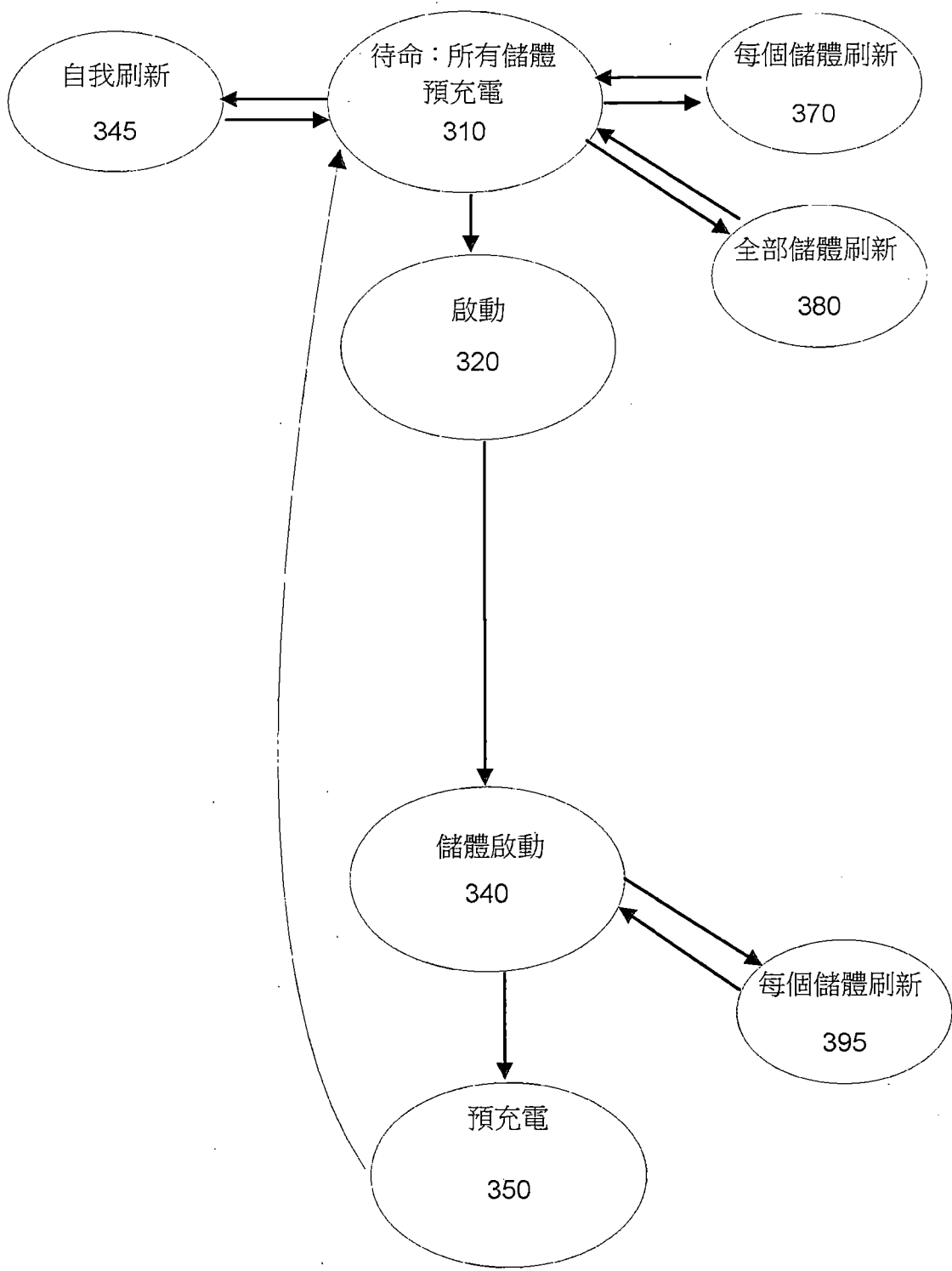
【英文】

A method for updating a DRAM memory array is disclosed. The method comprises: a) transitioning the DRAM memory array to a refresh mode in accordance with a command from a memory controller; b) performing a refresh cycle on the DRAM memory array; c) performing an Error Correction Code (ECC) scrub of a selected column in the DRAM memory array over all rows of the DRAM memory array during the refresh cycle; and d) incrementing a refresh column counter to select a subsequent column in the DRAM memory array.

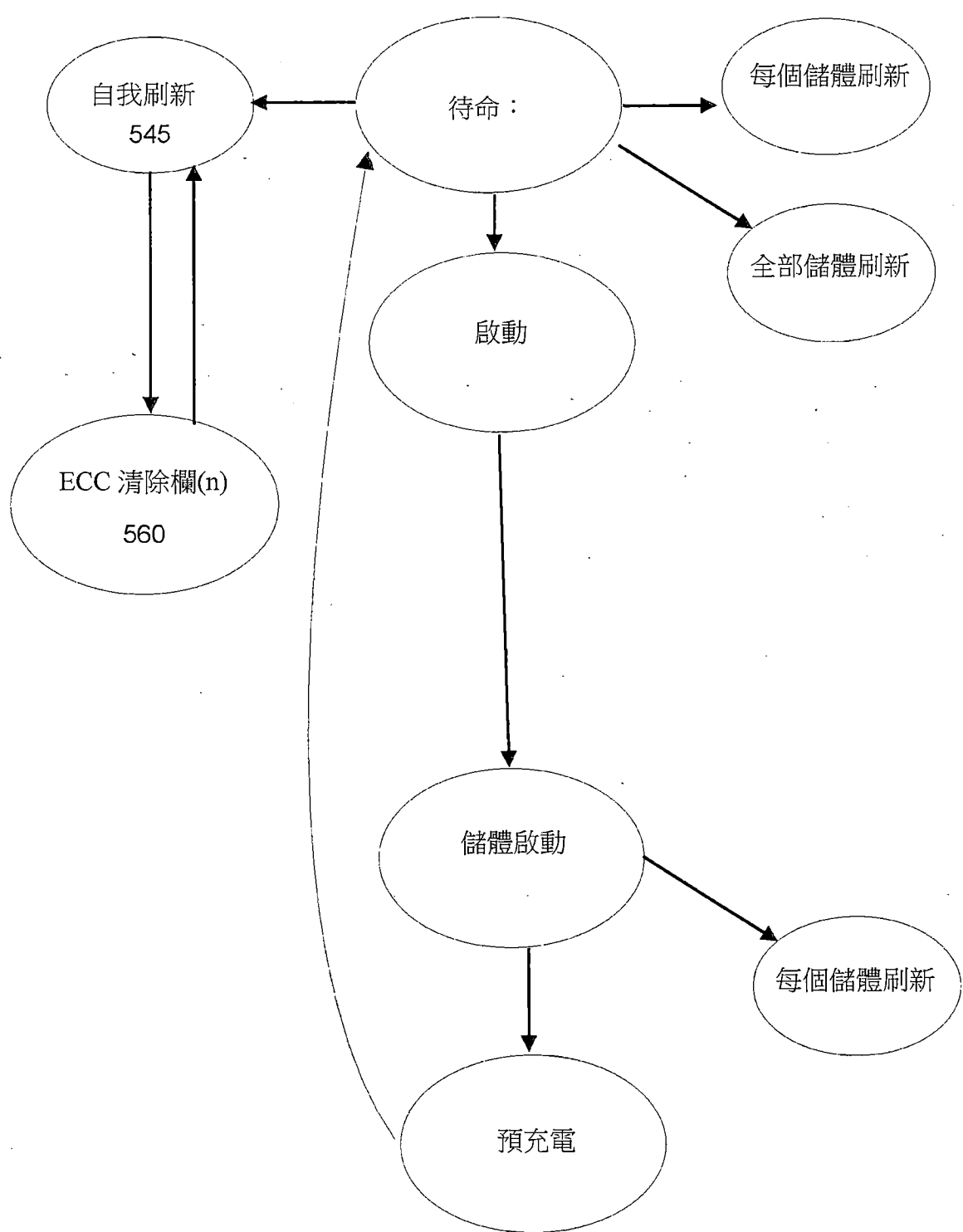
圖式



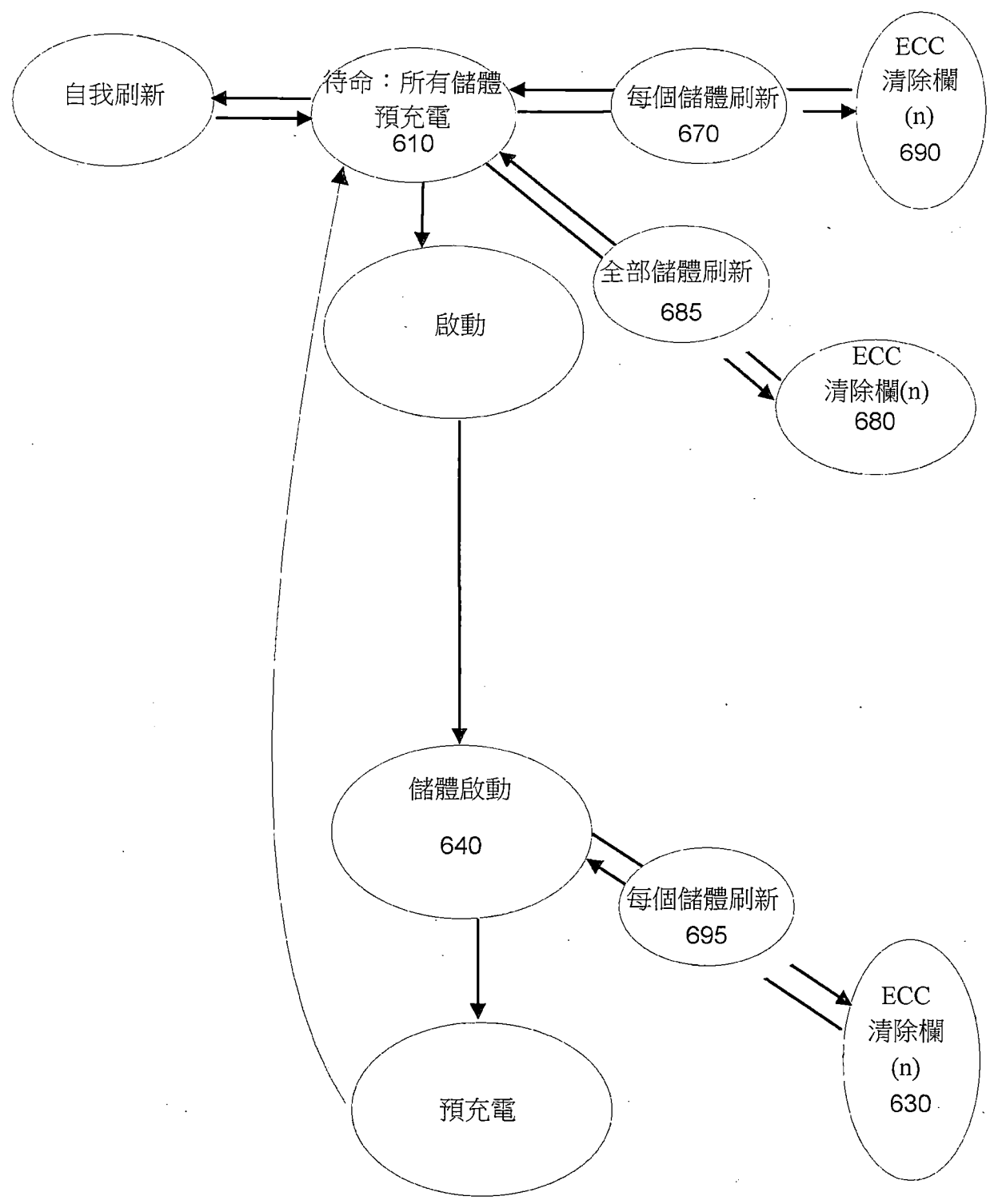
第一圖



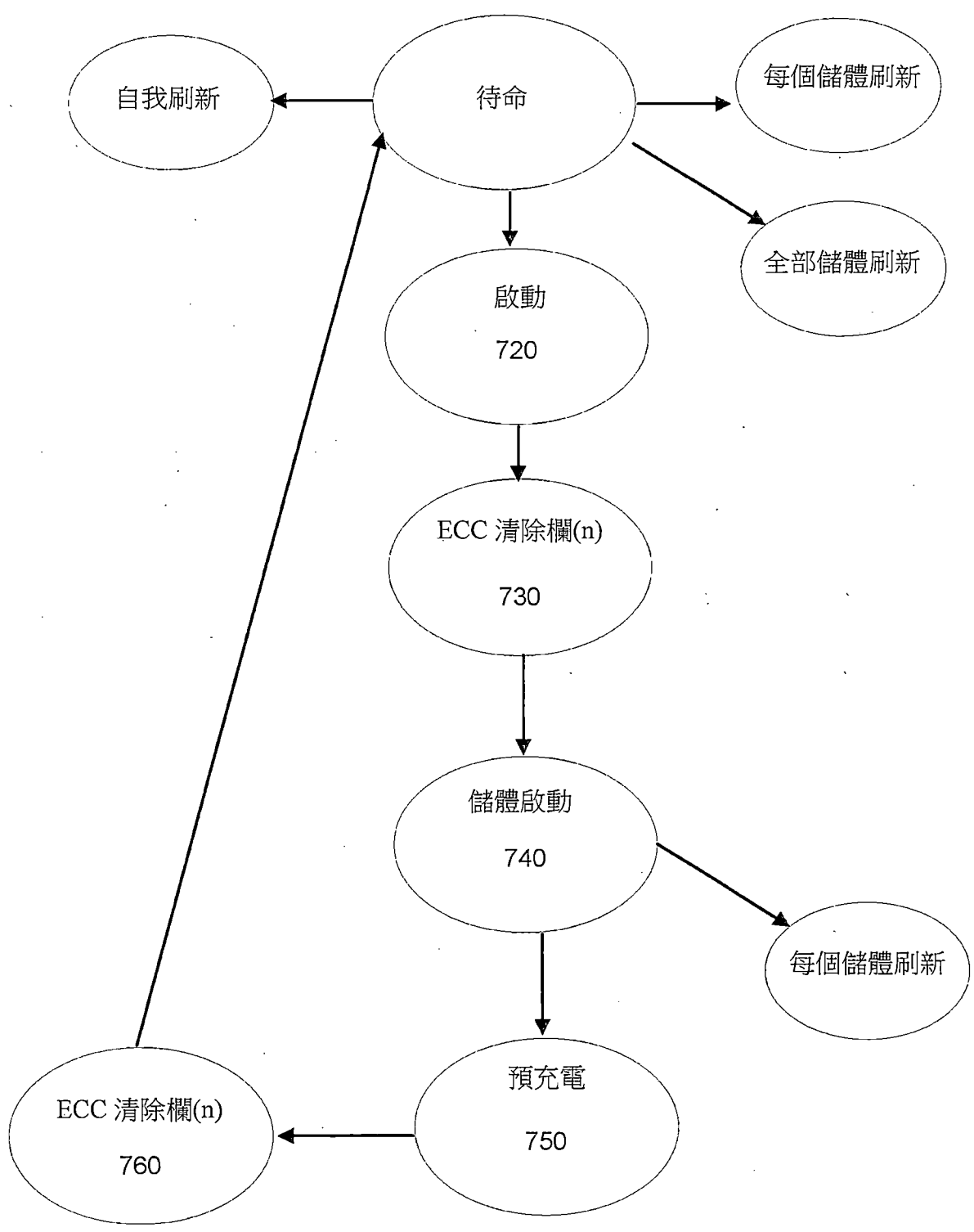
第三圖



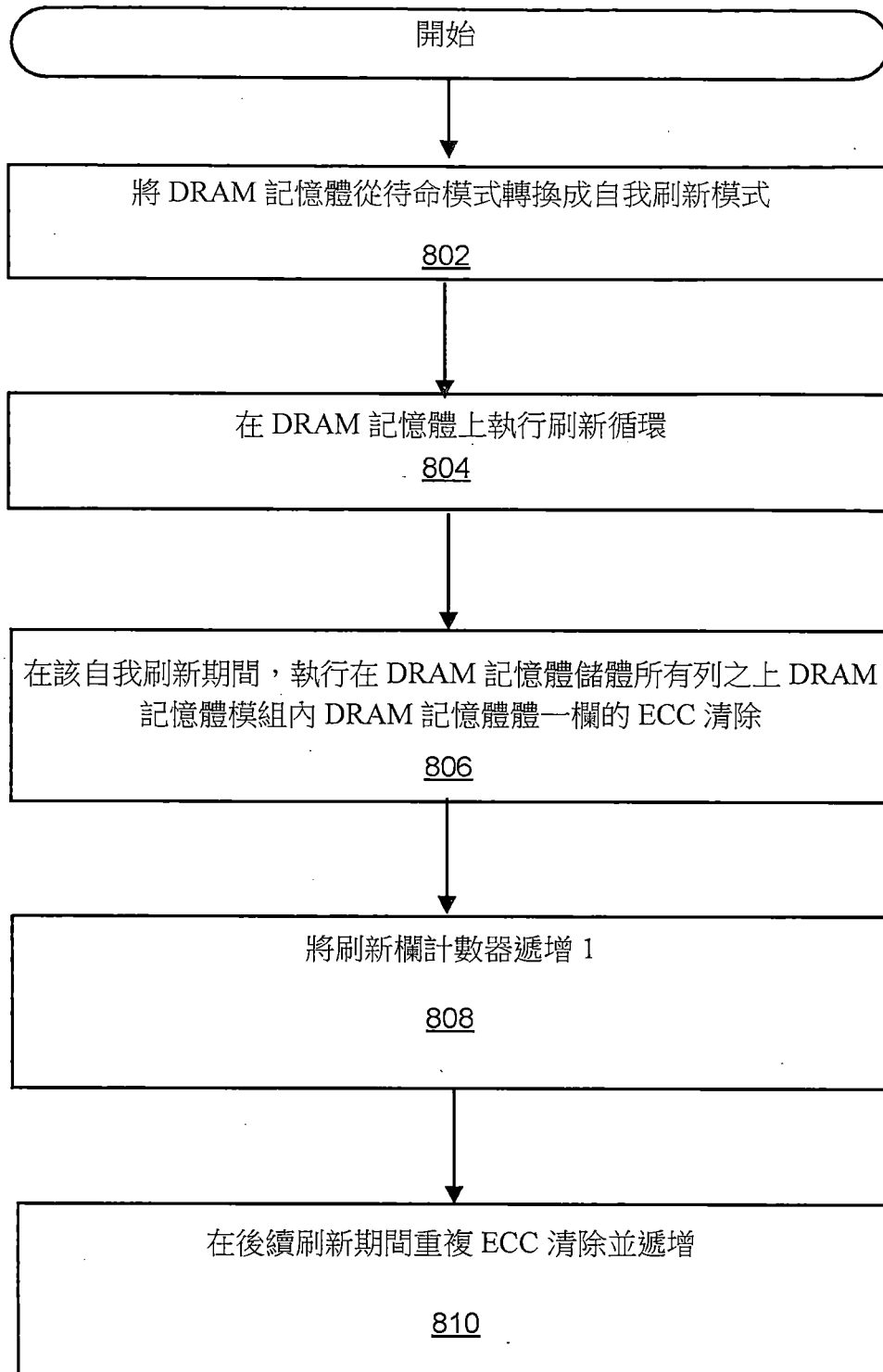
第五圖



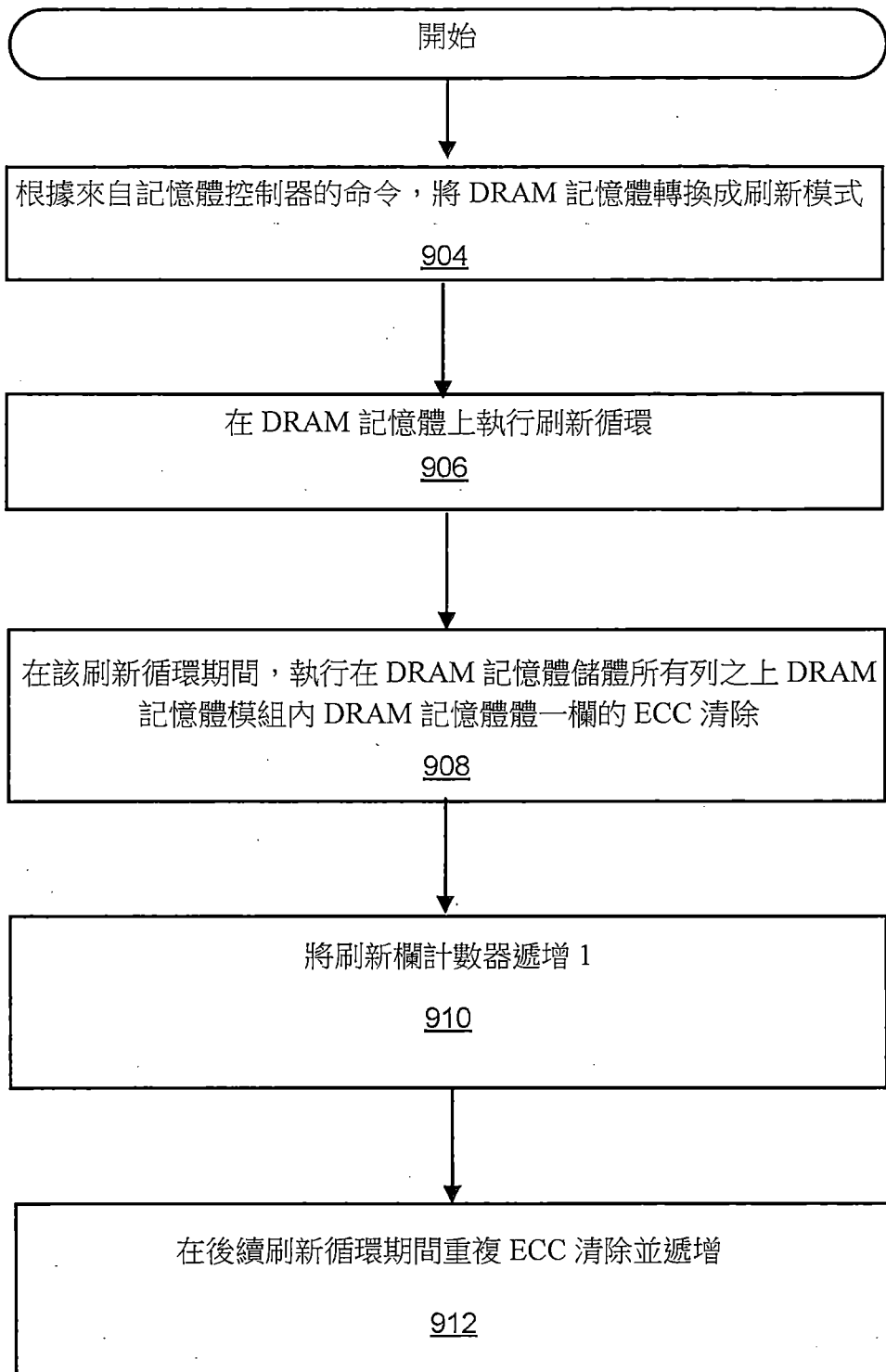
第六圖



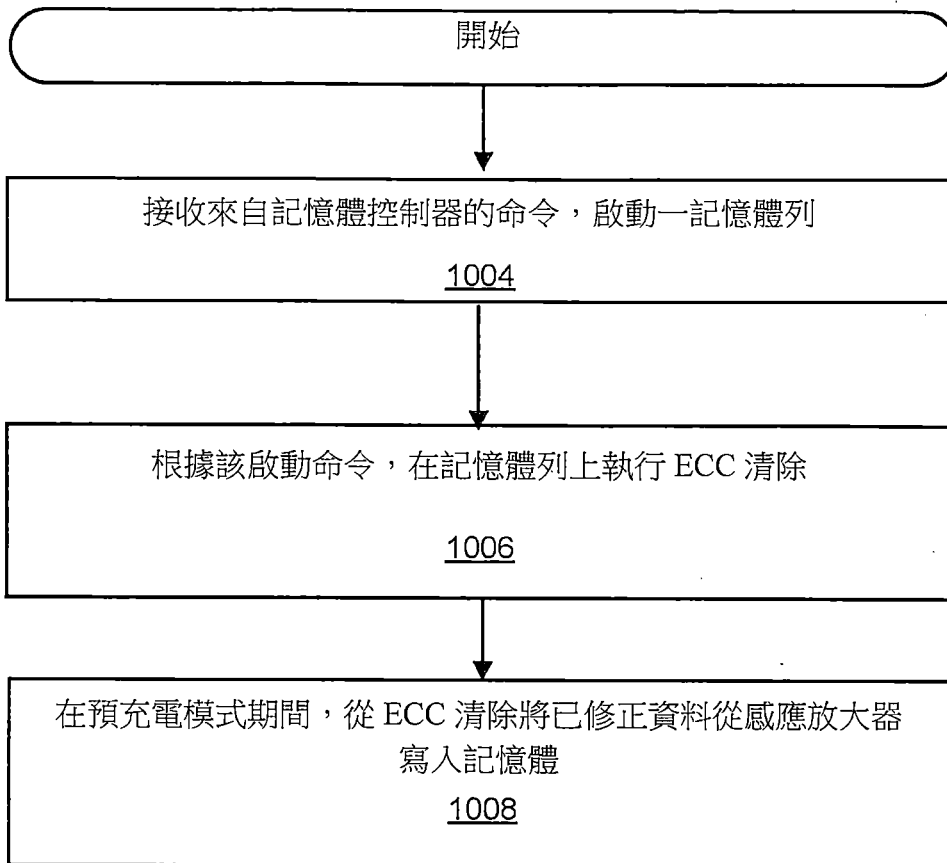
第七圖



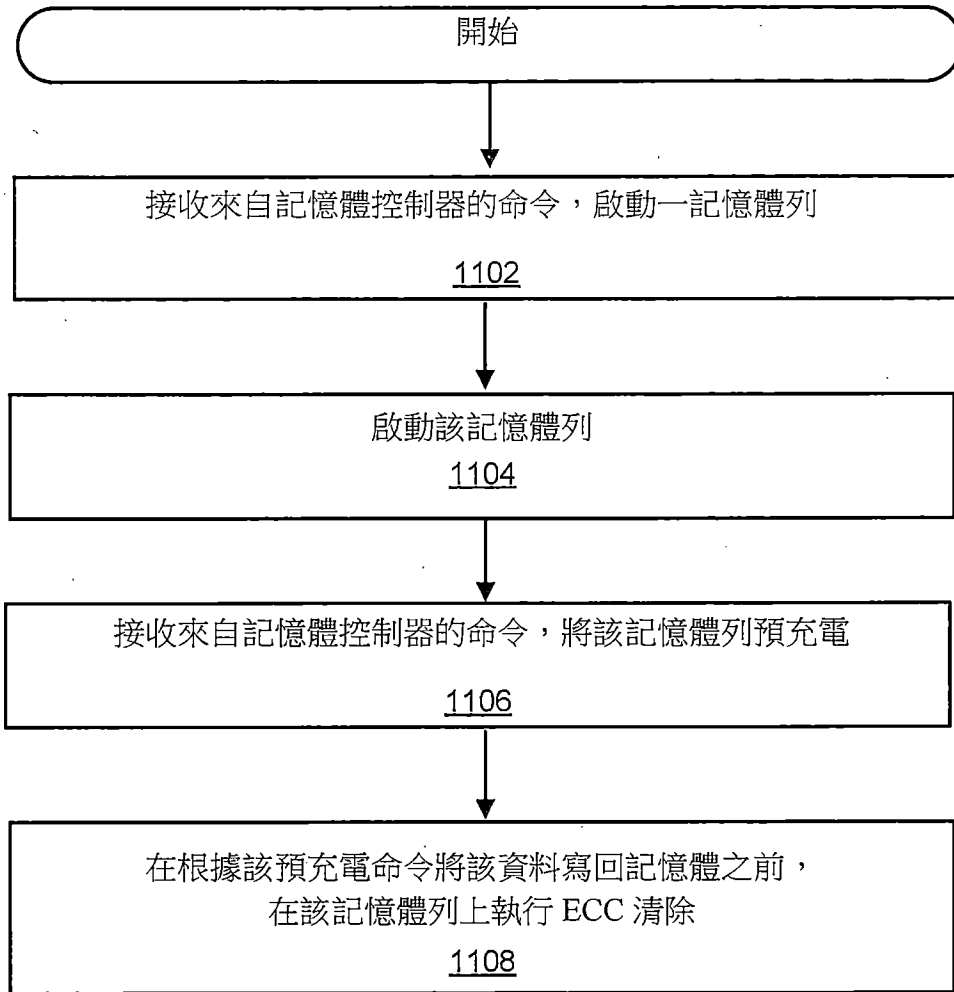
第八圖



第九圖



第十圖



第十一圖

【代表圖】

【本案指定代表圖】：第（一）圖。

【本代表圖之符號簡單說明】：

110	計算系統	126	顯示配接器
112	通訊基礎設施	128	輸入裝置
114	處理器	130	輸入介面
116	系統記憶體	132	主要儲存裝置
118	記憶體控制器	133	備用儲存裝置
120	輸入/輸出控制器	134	儲存介面
122	通訊介面	140	資料庫
124	顯示裝置		

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

140	資料庫	350	預充電模式
200	網路架構	370	狀態
231	位址鎖存器	380	狀態
232	命令解碼器和邏輯模組	395	狀態
233	多工器模組	431	刷新欄計數器
234	列解碼器	437	感應放大器
235	儲體	438	I/O 選擇模組
236	讀取資料/ECC 邏輯模組	440	ECC 邏輯模組
237	感應放大器	545	低功率自我刷新模組
238	I/O 選擇模組	560	狀態
239	儲體選擇邏輯模組	630	ECC 清除
240	ECC 邏輯	640	儲體啟動狀態
241	寫入資料模組	670	狀態
243	自我刷新邏輯模組	680	狀態
244	刷新列計數器	685	狀態
271	欄位址鎖存器模組	690	狀態
280	欄解碼器	695	每儲體刷新
290	DQ 線	720	啟動狀態
291	位址	730	狀態
291	命令	740	儲體啟動狀態
310	待命狀態	750	預充電狀態
320	啟動狀態	760	狀態
340	狀態		

申請專利範圍

1. 一種用於更新一 DRAM 記憶體陣列的方法，該方法包含：
 - a) 根據來自一記憶體控制器的一命令，將該 DRAM 記憶體陣列轉換成一刷新模式；
 - b) 在該 DRAM 記憶體陣列上執行一刷新循環；
 - c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有列上，執行該 DRAM 記憶體陣列內一選取欄的錯誤修正碼(ECC)清除，其中該錯誤修正碼(ECC)清除包括執行一虛擬讀取與寫入操作；以及
 - d) 更新一刷新欄計數器。
2. 如申請專利範圍第 1 項之方法，其中步驟d)進一步包含：

將該刷新欄計數器遞增，選擇該 DRAM 記憶體陣列內要清除的一後續欄。
3. 如申請專利範圍第 2 項之方法，進一步包含：

針對有關該後續欄的一後續刷新循環，重複 a)、b)、c)和 d)。
4. 如申請專利範圍第 3 項之方法，其中該執行一 ECC 清除進一步包含：

從該記憶體控制器發出一隱藏寫入命令，來選擇一記憶體位置；
停用該隱藏寫入命令的所有隱藏位元；以及
讀取該記憶體位置內現有的資料，以便執行該 ECC 清除。
5. 如申請專利範圍第 4 項之方法，其中該刷新循環為從一 DRAM 待命狀態執行的一每儲體刷新之一部分。
6. 如申請專利範圍第 4 項之方法，其中該刷新循環為從一 DRAM 待命狀態執行的全部儲體刷新之一部分。
7. 如申請專利範圍第 4 項之方法，其中該刷新循環為從一 DRAM 啟動狀態執行的一每儲體刷新之一部分。
8. 如申請專利範圍第 4 項之方法，進一步包含：

在針對該 DRAM 記憶體陣列內所有該等欄執行一 ECC 清除之後，遞增一儲體計數器，將一後續 DRAM 記憶體陣列定址。
9. 如申請專利範圍第 4 項之方法，其中該執行一 ECC 清除進一步包含：

讀取該選取欄的資料；

- 檢查對應至該選取欄的 ECC 位元；
若該 ECC 位元指示一錯誤，則針對該選取欄修正該資料；
重新計算用於該選取欄的該 ECC 位元；以及
將已修正資料寫回該 DRAM 記憶體陣列。
10. 一種用於更新一 DRAM 記憶體陣列的方法，該方法包含：
- a) 根據來自一記憶體控制器的一命令，將該 DRAM 記憶體陣列轉換成一刷新模式；
 - b) 在該 DRAM 記憶體陣列上執行一刷新循環；以及
 - c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有列上，執行該 DRAM 記憶體陣列內複數個選取欄的錯誤修正碼(ECC)清除，其中該錯誤修正碼(ECC)清除包括執行一虛擬讀取與寫入操作；以及
 - d) 更新一刷新欄計數器。
11. 如申請專利範圍第 10 項之方法，其中步驟d)進一步包含：
- 用該等複數個選取欄的數量遞增該刷新欄計數器，將該 DRAM 記憶體陣列內用於清除的一後續欄定址。
12. 如申請專利範圍第 11 項之方法，進一步包含：
- 針對有關該後續欄的一後續刷新循環，重複 a)、b)、c)和 d)。
13. 如申請專利範圍第 12 項之方法，其中該刷新循環為從一 DRAM 待命狀態執行的一每儲體刷新之一部分。
14. 如申請專利範圍第 12 項之方法，其中該刷新循環為從一 DRAM 待命狀態執行的全部儲體刷新之一部分。
15. 如申請專利範圍第 12 項之方法，其中該刷新循環為從一 DRAM 啟動狀態執行的一每儲體刷新之一部分。
16. 如申請專利範圍第 12 項之方法，其中該執行一 ECC 清除進一步包含：
- 從該記憶體控制器發出一隱藏寫入命令，來選擇一記憶體位置；
 - 停用該隱藏寫入命令的所有隱藏位元；以及
 - 讀取該記憶體位置內現有的資料，以便執行該 ECC 清除。
17. 如申請專利範圍第 16 項之方法，其中該執行該 ECC 清除進一步包含：
- 讀取該選取欄的資料；

檢查對應至該選取欄的 ECC 位元；

若該 ECC 位元指示一錯誤，則針對該選取欄修正該資料；

重新計算用於該選取欄的該 ECC 位元；以及

將已修正資料寫回該 DRAM 記憶體陣列。

18. 一種用於更新一 DRAM 記憶體陣列的設備，該設備包含：

該 DRAM 記憶體陣列，其中該 DRAM 記憶體陣列設置成：

a) 根據來自一記憶體控制器的一命令，轉換成一刷新模式；

b) 執行一刷新循環；以及

c) 在該刷新循環期間，在該 DRAM 記憶體陣列的所有該等列上，執行該 DRAM 記憶體陣列內一選取欄的一 ECC 清除，其中該錯誤修正碼(ECC)清除包括執行一虛擬讀取與寫入操作；以及

d) 更新一刷新欄計數器。

19. 如申請專利範圍第 18 項之設備，其中所述d)進一步包含：

將該刷新欄計數器遞增，將該 DRAM 記憶體陣列內要清除的一後續欄定址。

20. 如申請專利範圍第 19 項之設備，其中該 DRAM 記憶體陣列進一步設置成針對有關該後續欄的一後續刷新循環重複該 a)、b)、c)和 d)。