

肆、聲明事項：

☐ 本案係符合專利法第二十條第一項 ☐ 第一款但書或 ☐ 第二款但書規定之期間，其日期為： 年 月 日。

☒ 本案申請前已向下列國家(地區)申請專利：

1.美國；2002 年 08 月 29 日；10/234,001

2.

3.

4.

5.

☒ 主張國際優先權(專利法第二十四條)：

【格式請依：受理國家(地區)；申請日；申請案號數 順序註記】

1.美國；2002 年 08 月 29 日；10/234,001

2.

3.

4.

5.

☐ 主張國內優先權(專利法第二十五條之一)：

【格式請依：申請日；申請案號數 順序註記】

1.

2.

☐ 主張專利法第二十六條微生物：

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

☐ 熟習該項技術者易於獲得，不須寄存。

玖、發明說明：

【發明所屬之技術領域】

一般而言，本發明關於此等電腦記憶體裝置，尤其，關於此等必須被定期更新，以避免遺失資料之記憶體裝置。

【先前技術】

某些記憶體裝置能無限期維持該資訊儲存於該記憶體，甚至當該記憶體裝置的電源被關掉。此等記憶體裝置是熟知的非揮發性記憶體裝置。一些非揮發性記憶體裝置之範例包括此等磁性隨機存取記憶體(MRAM)，此等可抹除可程式化唯讀記憶體(EPROM)，及其變化。

其他記憶體裝置需要電源以維持資訊儲存於該記憶體。此等記憶體裝置是熟知的揮發性記憶體裝置，必須被定期更新以避免遺失資料。一揮發性記憶體裝置之一常見範例是一動態隨機存取記憶體(DRAM)，其中儲存於此等電容器之電壓代表資訊之此等數位位元。因逾時消耗儲存於一電容器之電壓，一DRAM之該等電容器必須被定期再充電，以維持資訊儲存於該DRAM。

此等慣用之揮發性記憶體裝置包括複數個配置成一陣列之獨立記憶體單元，通常包括此等交叉之列與行之配置。關於維持該資訊儲存於一慣用之揮發性記憶體裝置，通常每秒數次更新該陣列中之各記憶體裝置。例如，於一些慣用之DRAM中，各記憶體裝置必須被更新大約每64毫秒一次，以避免遺失儲存於該DRAM中之資訊。因此，此等慣用之揮發性記憶體裝置通常包括更新電路，以確保每當需

於另一實施例中，一種包括複數個必須被定期更新之單元的記憶體裝置，其中為回應來自一處理器之一寫指令或一抹除指令，配置的該複數個單元被更新，但其中未配置該記憶體裝置，產生或接收一不同於該寫指令或抹除指令之更新控制信號。該複數個單元之每一個皆包括一可程式化之金屬化單元，該可程式化之金屬化單元包括一具有上表面的單元主體，其中該單元主體包括一硫屬金屬離子玻璃及兩配置在該上表面之電極，其中該等電極彼此有距離間隔開。

於另一實施例中，一種包括複數個必須被定期更新之單元的記憶體裝置，其中為回應來自一處理器之一寫指令或一抹除指令，配置的該複數個單元被更新。

【實施方式】

圖1說明一包括一慣用之揮發性記憶體110之電腦系統100。該電腦系統100也包括一經由一匯流排130耦合該記憶體裝置110之處理器120。該記憶體裝置110包括一位址/資料/控制模組140與記憶體更新電路150，兩者皆經由該匯流排130耦合該處理器120。該記憶體裝置110也包括經由線路200耦合該位址/資料/控制模組140，並經由線路210耦合記憶體更新電路150之一記憶體陣列160。熟悉此項技藝者應瞭解該等線路200，210可實行於該記憶體裝置110，作為一信號實體匯流排的部分。該記憶體陣列160包括複數個記憶體單元170，由複數條相交的列180與行190互連。

於該記憶體裝置110中執行此等作業，該處理器120經由

匯流排 130 傳輸一些信號至該記憶體裝置 110。例如，讀取儲存在一特定記憶體位址的資料，該處理器 120 發出一讀取命令，連同該記憶體位址給該匯流排 130。該位址/資料/控制模組 140 經由線路 200 接收並處理存取該記憶體陣列 160 之讀取命令。尤其，該位址/資料/控制模組 140 在線路 200 產生一讀取控制信號，並藉由啟動該適當之列 180 與行 190，以定址希望的記憶體單元 170。該位址/資料/控制模組 140 接著經由線路 200 接收儲存在該定址記憶體單元 170 之資料，並經由匯流排 130 將該資料傳遞給該處理器 120。

另外，將資料寫至一特定的記憶體位址，該處理器 120 對該匯流排 130 發出一寫命令，連同該記憶體位址與該儲存的資料。如上述，該位址/資料/控制模組 140 接收並處理該寫命令，而在線路 200 上產生一寫的控制信號並定址希望的記憶體單元 170。該位址/資料/控制模組 140 接著經由線路 200 傳遞儲存在該定址記憶體單元 170 之資料。

因為儲存於該記憶體單元 170 之資料浪費過多時間，所以必須定期更新該記憶體單元 170，以避免遺失儲存於該記憶體裝置 110 之資料。該記憶體更新電路 150 主要的功能是執行該週期性的更新作業。當該記憶體更新電路 150 決定需要執行一記憶體更新作業時，該記憶體裝置 110 在該處理器 120 執行其他作業時變成不可用，例如一讀作業或一寫作業。

對於執行一記憶體更新作業，該記憶體更新電路 150 通常連續定址該記憶體陣列 160 之該等列 180。當該記憶體陣列 160 之一特定列 180 被定址，該記憶體更新電路在線路 200 產

由一匯流排280耦合至該記憶體裝置260之處理器270。該電腦系統250進一步包括一組實行於軟體中，能夠由該處理器270執行之記憶體更新指令290。該記憶體裝置260包括一經由該匯流排280耦合至該處理器270之位址/資料/控制模組300。該記憶體裝置260也包括一經由線路350耦合至位址/資料/控制模組300之記憶體陣列310。該記憶體陣列310耦合複數個記憶體單元320，由複數列330與行340互連。

說明於圖2之電腦系統250，使用描述於上面有關圖1之同一方法，執行此等記憶體讀與寫作業。然而，除了此等方法之外，該電腦系統250能夠以熟悉此項技藝者所熟知的各種其他方法執行此等記憶體之讀與寫作業。

於一些實施例中，該軟體更新之記憶體裝置260之該等記憶體單元320包括此等比慣用之DRAM記憶體單元更好更穩定之揮發性記憶體單元。例如，於一實施例中，該等記憶體單元320能避免遺失資料，即使此等連續記憶體更新作業之間的時間間隔大約是0.1秒。於另一實施例中，此等連續記憶體更新作業之間的時間間隔大約是1秒。於另一實施例中，此等連續記憶體更新作業之間的時間間隔大約是1小時。於另一實施例中，此等連續記憶體更新作業之間的時間間隔大約是1星期。

於一些實施例中，該軟體更新之記憶體裝置260之該等記憶體單元320包括此等可程式化之導體隨機存取記憶體(PCRAM)單元，描述於屬於Kozicki及其他人("此等Kozicki專利案")之美國專利案號5,761,115，5,896,312，5,914,893

，6,084,796，屬於Moore及其他人("Moore專利案")之美國專利案號6,348,365，及下面此等聯合待案的美國專利申請案：2002年4月10日提出申請，序號10/121,792，標題"Method of Manufacture of Programmable Conductor Memory"，2002年4月10日提出申請，序號10/121,790，標題"Programmable Conductor Memory Cell Structure and Method Therefor"，及2002年4月10日提出申請，序號10/121,794，標題"Thin Film Diode Integrated with Chalcogenide Memory"。該等Kozicki專利案，該Moore專利案，及此等聯合待案的專利申請案因而以引用的方式併入本文中。於此等參考中更詳細描述一包括一對電極之PCRAM單元，且能存在兩種可能狀態之一。於該第一狀態中，該PCRAM單元之該等電極之間存在一短路。於該第二狀態中，該PCRAM單元之該等電極之間存在一開路。

一PCRAM單元之一些實施例，包括一玻璃離子電極，例如一硫屬金屬離子玻璃，及兩配置在該玻璃離子導體表面，由一距離間隔隔開的電極。於一實施例中，一PCRAM單元包括鍺矽化物與一第IV群金屬(例如銀)溶解在其中，例如Ag/Ge₃Se₇。該等電極之一最好包括一第IV群金屬，且該PCRAM單元之玻璃元素包含同一金屬。

於作業中，將具有一第一極性之電壓加入穿過一PCRAM單元之該等電極，在形成於其中之玻璃元件之通道側壁中的該等電極之間建立一傳導路徑。當將具有相反極性之一電壓加入穿過此等電極，該金屬離子再溶解成該單元主體

，因而導致該傳導路徑消失。藉由測量該等電極之間的電阻，以偵測一PCRAM單元中存在或不存在一傳導路徑。當一傳導路徑存在，一短路存在該等電極之間，而該等電極之間的電阻是低的(例如，千分之一歐姆)。另一方面，當一傳導路徑不存在，一開路存在該等電極之間，而該等電極之間的電阻是高的(例如，百萬歐姆)。

下面論述的此等作業程序提及圖2之該等特徵。

通常，該記憶體裝置260之該等記憶體單元320能存在兩狀態之一，即，一"寫"狀態或一"抹除"狀態。例如，如果一記憶體單元320包括一能夠保持電荷之電容器，則對應該寫狀態，會有一電荷存在該電容器，而對應該抹除狀態，不會有一電荷存在該電容器。同樣地，如果一PCRAM單元作為一記憶體單元320，則對應該寫狀態，該等電極之間會存在一傳導路徑，而對應該抹除狀態，該等電極之間不會存在一傳導路徑。熟悉此項技藝者會了解，一般而言，對應該寫狀態，一影響性元件存在一記憶體單元320，反之對應該抹除狀態，一影響性元件不存在。

如上面有關圖1之論述，當放置一慣用之揮發性記憶體裝置110之一記憶體單元170於一特定狀態，該記憶體單元在一段相當短的時間內，仍然處於一特定狀態，例如，大約64毫秒。由於此類記憶體單元170在此段短的時間，保留其指定狀態，通常必須更新各記憶體單元170，例如，通常超過每秒15次。

相較之下，圖2說明之軟體更新之記憶體裝置260之一記

記憶體單元320被置於一特定狀態，有利於該記憶體單元320在一段相當短的時間內，仍然處於該特定狀態。例如，於一些實施例中，各記憶體單元320能維持一特定狀態數秒，數分，數小時，數天，數星期，或更長的期間。因此，能夠保護資料儲存於該記憶體裝置260，而且能較不頻繁地執行此等記憶體更新作業，例如，大約每幾星期一次，而不是每秒幾次。

由於此等記憶體更新作業之發生較不頻繁，所以該電腦系統250有利於藉由執行一組實行於軟體中而不是硬體中之記憶體更新指令290，以執行此等記憶體更新作業。例如，於一些實施例中，該等記憶體更新指令290構成該電腦系統250之作業系統的部分。

藉由實行該等記憶體更新指令290於軟體中，而不是硬體中，有利於簡化該記憶體裝置260。例如，有利於排除線路210上所需要的該記憶體更新電路150與一唯一的更新控制信號，如圖1所說明。實行該等記憶體更新指令290於軟體中，而不是硬體中之另一優點，是該處理器270(圖2)與該記憶體更新電路150(圖1)相比，能執行更複雜的演算法，以決定何時執行一記憶體更新作業。

圖3說明一種根據本發明之一實施例更新一記憶體單元320(圖2)之方法。於一第一步驟400中，該程序開始。於下一步驟402，該處理器270讀取該記憶體單元320之狀態。指定該記憶體單元320之狀態為一對應資料之一數位位元之邏輯值。例如，於一些實施例中，一記憶體單元320之寫狀

態可對應一邏輯"1"，而該記憶體單元320之抹除狀態可對應一邏輯"0"。另一方面，於一些替代實施例中，一記憶體單元320之寫狀態可對應一邏輯"0"，而該記憶體單元320之抹除狀態可對應一邏輯"1"。

於步驟404，該處理器270判斷儲存於記憶體單元320之資料位元是"1"或是"0"。如果該資料位元是"1"，接著，於步驟406，該處理器270將"1"寫入記憶體單元320。例如，如果"1"對應該寫入狀態，接著，於步驟406期間，該處理器270發出一"寫指令"至該記憶體單元，即，一寫命令被發出，而該記憶體單元320被置於該寫入狀態。於步驟404期間，如果判斷儲存於記憶體單元320之資料位元是"0"，接著，於步驟408，該處理器270將"0"寫入記憶體單元320。例如，如果"0"對應該抹除狀態，接著，於步驟408期間，該處理器270發出一"抹除指令"至該記憶體單元，即，一抹除命令被發出，而該記憶體單元320被置於該抹除狀態。

於一記憶體更新作業期間發出一寫指令與在一些標準作業期間發出一寫指令至該記憶體裝置260一樣。同樣地，於一記憶體更新作業期間發出一抹除指令與在一些標準作業期間發出一抹除指令至該記憶體裝置260一樣。因此，如上述，有利於排除僅使用於此等記憶體更新作業期間所需要之一唯一的更新控制信號。該處理器270完成步驟406或步驟408之後，接著，於最後步驟410，該程序結束。

於圖3說明之方法中，假設一記憶體單元320(圖2)不會無限期持續處於該寫狀態或該抹除狀態，除非藉由發出一寫

指令或一抹除指令至記憶體單元320，不管哪個是適當的，皆可定期更新記憶體單元320。然而，於一些實施例中，記憶體單元320傾向在整個時間回復到一特定狀態。例如，於一些實施例中，記憶體單元320傾向在整個時間回復到一抹除狀態。於此等實施例中，當一記憶體單元320被置於該寫狀態時，該記憶體單元320不會無限期保持在寫狀態，除非藉由在該記憶體單元320執行一寫作業，以定期更新該記憶體單元320。另一方面，由於該記憶體單元320之自然傾向是回復到該抹除狀態，因此置於該抹除狀態之一記憶體單元320無限期保持於該狀態，不需被更新。於此等實施例中，因排除步驟406或步驟408，而有助於簡化圖3所說明的該方法，不管哪個步驟對應該抹除狀態。

此外，於圖3說明之方法中，假設各記憶體單元320(圖2)僅存在兩狀態之一。然而，於本揭露觀點中，熟悉此項技藝者應瞭解，如果記憶體單元320能存在兩種以上的狀態，應如何修改說明於圖3之方法。

如果一記憶體更新作業包括說明於圖3之方法，則個別更新各記憶體單元320(圖2)，而不是同時更新大量記憶體單元320，如上面有關圖1之論述。因個別評估各記憶體單元320，該處理器270能決定各獨立記憶體單元320需要被更新，因而有助於避免對不需要被更新之記憶體單元320，執行不必要之更新作業。

圖4說明一種根據本發明之一實施例更新複數個記憶體單元320(圖2)之方法。為了執行該方法，該電腦系統250維

護一具有對應該行190之一特定位址之值的計數器。於第一步驟450，使用說明於圖3之方法或另一適合的方法，更新對應該計數器之現行值之位址的該記憶體單元320。於下一步驟452，增加該計數器之值，而於步驟454，重設並啟動一計時器。

於步驟456，該處理器270判斷該計時器是否已超過一預定之最小等待時間。藉由考慮一些因素，例如，該記憶體單元320能保持其指定狀態的最大時間，更新一記憶體單元320所需要的時間，被更新之記憶體單元320的數量等等，以決定該最小等待時間之適當值。如上述，該預定之最小等待時間的優點是可以是一相當長的時間，例如，數秒、數分、數小時、數天或更長的期間。於一實施例中，該預定之最小等待時間大約是一分鐘。於另一實施例中，該預定之最小等待時間大約是一小時。於另一實施例中，該預定之最小等待時間大約是一天至一星期。

如果還未到達該最小等待時間，接著於步驟458中，該處理器270判斷對應該計數器之現行值之位址的記憶體單元320是否需要被更新。一些不同的情況指出該現行位址之記憶體單元320不需要被更新。例如，如上述，於一些實施例中，當一記憶體單元320正處於該抹除狀態時，該記憶體單元320不需要被更新。此外，在步驟454重設並啟動該計時器之後，如果該處理器270偶爾對該記憶體單元320執行一寫作業，則直到下一記憶體更新週期前，該記憶體單元320不需要被更新。

如果在等待該計時器達到該最小等待時間時，該處理器270決定該現行位址的記憶體單元320不需要被更新452，則增加對應該現行記憶體位址之該計數器之值，而且繼續執行該程序，如上述。另一方面，如果一但該計時器到達該最小等待時間，該現行位址之記憶體單元320仍然需要被更新，則該程序處理到步驟460，該處理器270判斷此等系統資源是否可用於更新該記憶體單元320。於執行該判斷中，該處理器270可評估廣泛的各種變化因素，例如，該電腦系統250之處理器270與記憶體裝置260之需求，等等。

如果此等系統資源是可用的，則該程序回到步驟450，當該現行位址之記憶體單元320被更新，且該程序繼續，如上述。另一方面，如果此等系統資源不可用於更新記憶體單元320，則於步驟462中，該處理器270判斷該計時器是否已超過一預定之最大等待時間。與該最小等待時間相同，可藉由考慮一些因素，例如，一記憶體單元320能保持其指定狀態的最大時間，更新一記憶體單元320所需要的時間，被更新之記憶體單元320的數量等等，以決定該最大等待時間之適當值。如上述，該預定之最大等待時間的優點是可以為一相當長的時間，例如，數秒、數分、數小時、數天、數星期或更長的期間。於一實施例中，該最大等待時間大約是一小時。於另一實施例中，該最大等待時間大約是一星期。於另一實施例中，該最大等待時間大約是一個月。

如果還未到達該最大等待時間，接著於步驟464中，該處理器270判斷對應該計數器之現行值之位址的記憶體單元

320是否需要被更新。如上面有關步驟458之描述，一些不同的情況指示該現行位址之記憶體單元320不需要被更新。

如果該現行位址之記憶體單元320不需要被更新，則該程序回到步驟452，則增加該計數器對應該現行記憶體位址之值，且該程序繼續，如上述。另一方面，如果該現行位址之記憶體單元320需要被更新，則該處理器270繼續監測此等系統資源是否變成可用於更新該記憶體單元320。

一旦該計時器到達該預定的最大等待時間，如果此等系統資源還是不可用，且該現行位址之記憶體單元320仍然需要被更新，則該程序繼續執行步驟466，該處理器270迫使其他程序釋放某些系統資源，使更新該記憶體單元320所需要的該等資源變成可用的。該程序接著回到步驟450，更新該現行位址之記憶體單元320，且該程序繼續，如上述。

於一實施例中，該處理器270不停地再三執行圖4所說明之程序。於該實施例中，該記憶體更新作業是一不間斷的程序，不斷發生於該處理器270所執行之其他程序的幕後。選擇與調整該最小等待時間與該最大等待時間，有利於盡可能少更新各記憶體單元320，而保護資訊儲存於該記憶體裝置310。

圖5說明另一種根據本發明之一實施例更新複數個記憶體單元320之方法。於第一步驟500中，該程序開始，而於下一步驟502，重設並啟動一計時器。與圖4說明之方法相同，圖2之電腦系統250維護一具有對應該記憶體陣列310中之一特定位址之值的計數器，以執行說明於圖5之方法。於

步驟504，使用說明於圖3之方法或另一適合的方法，更新對應該計數器之現行值之位址的記憶體單元320。於下一步驟506，增加該計數器之值。

於步驟508，處理器270判斷該記憶體更新作業是否已完成。一些情況指出該記憶體更新作業已完成。例如，當一記憶體陣列310中之各記憶體單元320已被更新，或者當記憶體之一特定區塊中之各記憶體單元320已被更新，則該記憶體更新作業被視為已完成。

如果該記憶體更新作業尚未完成，接著於步驟510中，該處理器270判斷此等系統資源是否可用於更新對應該計數器之現行值之位址之記憶體單元320。如果此等系統資源是可用的，則該程序回到步驟504，該現行位址之記憶體單元320被更新，且該程序繼續，如上述。

另一方面，如果此等系統資源不能用於更新該記憶體單元320，則於步驟512中，該處理器270判斷該計時器是否已超過一預定的最大等待時間。如上面有關圖4之說明，可藉由考慮一些因素，例如，一記憶體單元320能保持其指定狀態的最大時間，更新一記憶體單元320所需要的時間，被更新之記憶體單元320的數量等等，以決定該最大等待時間之適當值。此外，如上述，該預定之最大等待時間的優勢是可以為一相當長的時間，例如，數秒、數分、數小時、數天、數星期或更長的期間。

於一實施例中，上面有關圖4所論述之最大等待時間相當於該處理器270在其迫使此等系統資源變成可用於更新一

個別的記憶體單元320之前，可等待的最大時間。另一方面，於圖5說明的程序中，該最大等待時間相當於該處理器270在其迫使此等系統資源變成可用於更新該記憶體陣列310中或該記憶體區塊中之所有記憶體單元320之前，可等待的最大時間。因此，於一實施例中，該最大等待時間大約是一星期。於另一實施例中，該最大等待時間大約是一個月至兩個月。

如果尚未到達該最大等待時間，則程序回到步驟510，如上述，該處理器270繼續監測此等系統資源是否已變成可用於更新該記憶體單元320。一旦該計時器到達該預定最大等待時間，如果此等系統資源還是不可用，則該程序繼續執行步驟514，該處理器270迫使其他處理釋放某些系統資源，使該等需要的資源變成可用於更新記憶體單元320。該程序接著回到步驟504，更新該現行位址上之記憶體單元320，且該程序繼續，如上述。

該程序重複直到步驟508該處理器270判斷該記憶體更新作業已完成為止。一旦該判斷被執行，該程序接著繼續執行步驟516，重設該記憶體位址計數器。於最後步驟518，該程序結束。

於一實施例中，該處理器270定期重複執行說明於圖5之程序。於該實施例中，是定期執行該記憶體更新作業，而不是不斷地進行處理，如上面有關圖4之描述。處理器270有助於在一段固定的時間間隔呼叫並執行該處理，根據該電腦系統250的需要及資料儲存於該記憶體單元320之期間。

說明於圖3，4與5之程序只是實行該等記憶體更新指令290之演算法之範例。熟悉此項技藝者瞭解，藉由增加、刪除或變化某些步驟，即可容易地改變此等示範演算法。此外，於本揭露之觀點中，熟悉此項技藝者瞭解，可如何發展廣泛的各種替代演算法。

【圖式簡單說明】

圖1說明一包括一揮發性記憶體裝置之電腦系統。

圖2說明根據本發明之一實施例之一具有軟體更新之記憶體裝置之電腦系統。

圖3說明一種根據本發明之一實施例，更新一記憶體單元之方法。

圖4說明一種根據本發明之一實施例，更新複數個記憶體單元之方法。

圖5說明一種根據本發明之一實施例，更新複數個記憶體單元之方法。

【圖式代表符號說明】

100, 250	電腦系統
110	揮發性記憶體
120, 270	處理器
130, 280	匯流排
140, 300	位址/資料/控制模組
150	記憶體更新電路
160, 310	記憶體陣列
170, 320	記憶體單元

I263221

180, 330	列
190, 340	行
200, 210, 350	線路
260	記憶體裝置
290	記憶體更新指令

伍、中文發明摘要：

本發明揭露一種軟體更新之記憶體裝置，其包括複數個必須被定期更新以避免遺失資料之記憶體單元。最好，該等記憶體單元能避免遺失資料，即使此等連續記憶體更新作業之間的時間間隔，與一慣用揮發性記憶體裝置如一DRAM之此等連續記憶體更新作業之間的時間間隔相比是相當長。一處理器藉由執行一組實行於軟體而不是硬體之記憶體更新指令，以執行此等定期的記憶體更新作業。因此，有助於簡化該記憶體裝置，因為有助於忽略記憶體更新電路的需要與一唯一的更新控制信號。此外，與實行於硬體之記憶體更新電路相比，執行該等記憶體更新指令之該處理器通常能執行更複雜的演算法，以決定何時執行一記憶體更新作業。例如，該處理器能判斷各獨立之記憶體單元是否需要更新，因而，有助於避免對此等不需要更新作業之記憶體單元，執行不需要的更新作業。

陸、英文發明摘要：

A software refreshed memory device comprises a plurality of memory cells that must be periodically refreshed to avoid losing data. Preferably, the memory cells can avoid losing data even though the time interval between successive memory refresh operations is relatively long, as compared to the time interval between successive memory refresh operations in a conventional volatile memory device, such as a DRAM. A processor can perform periodic memory refresh operations by executing a set of memory refresh instructions implemented in software, rather than in hardware. Accordingly, the memory device can advantageously be simplified, because the need for memory refresh circuitry and for a unique refresh control signal are advantageously eliminated. Moreover, the processor executing the memory refresh instructions can typically perform more sophisticated algorithms, as compared to memory refresh circuitry implemented in hardware, for determining when to perform a memory refresh operation. For example, the processor can determine whether each individual memory cell needs to be refreshed, thereby advantageously avoiding performing unnecessary refresh operations on memory cells that do not need to be refreshed.

拾壹、圖式：

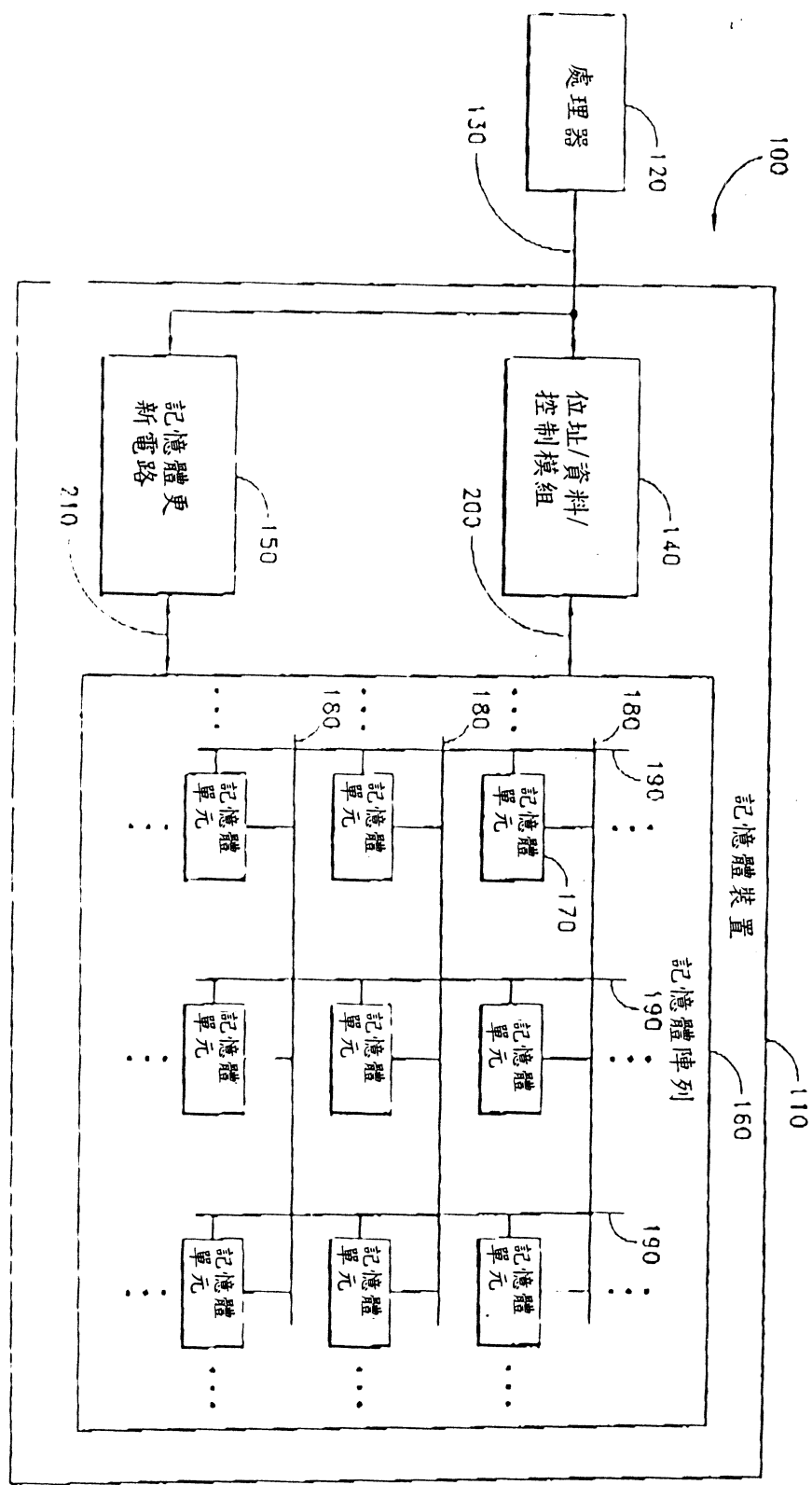


圖 1

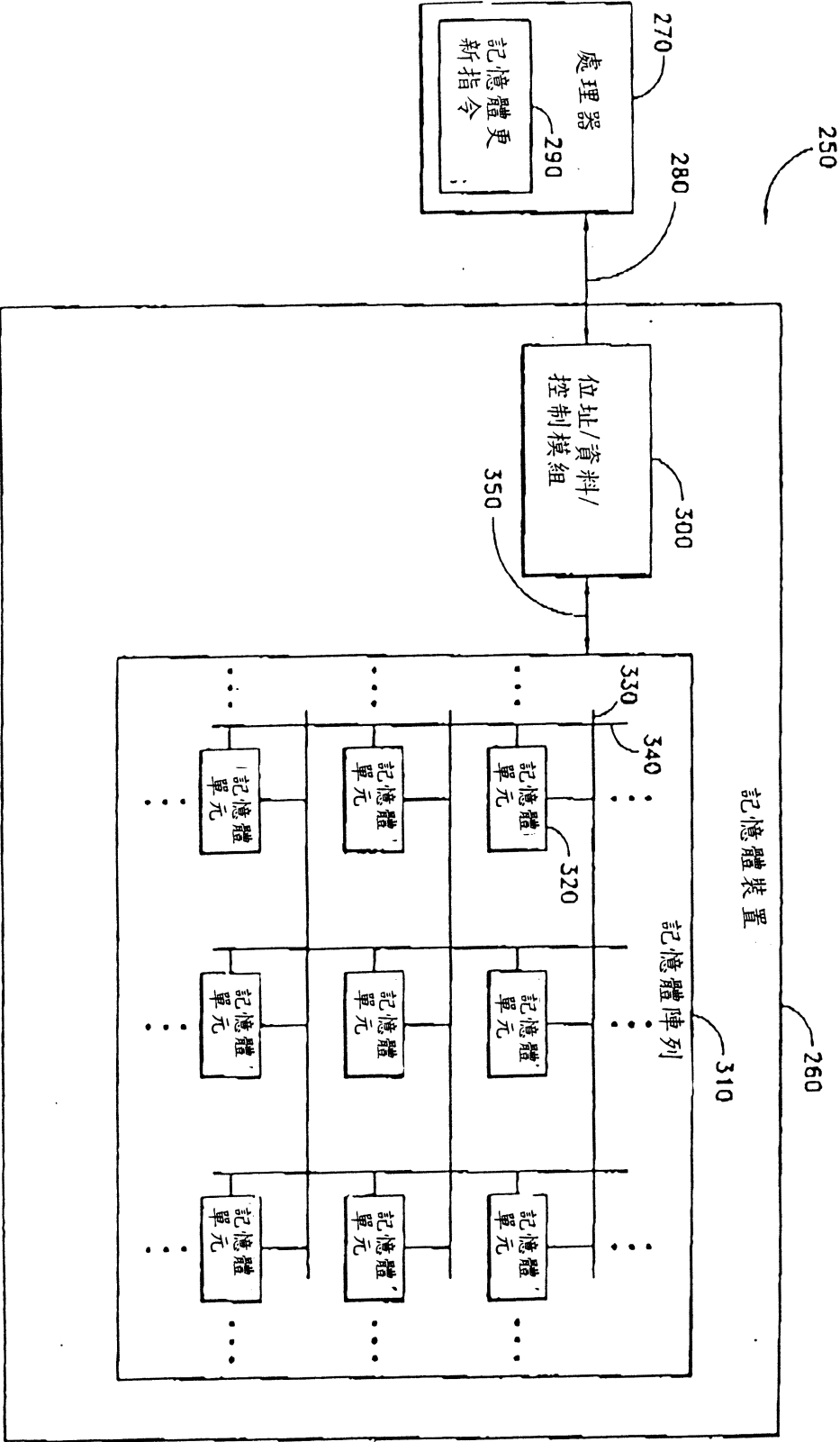


圖 2

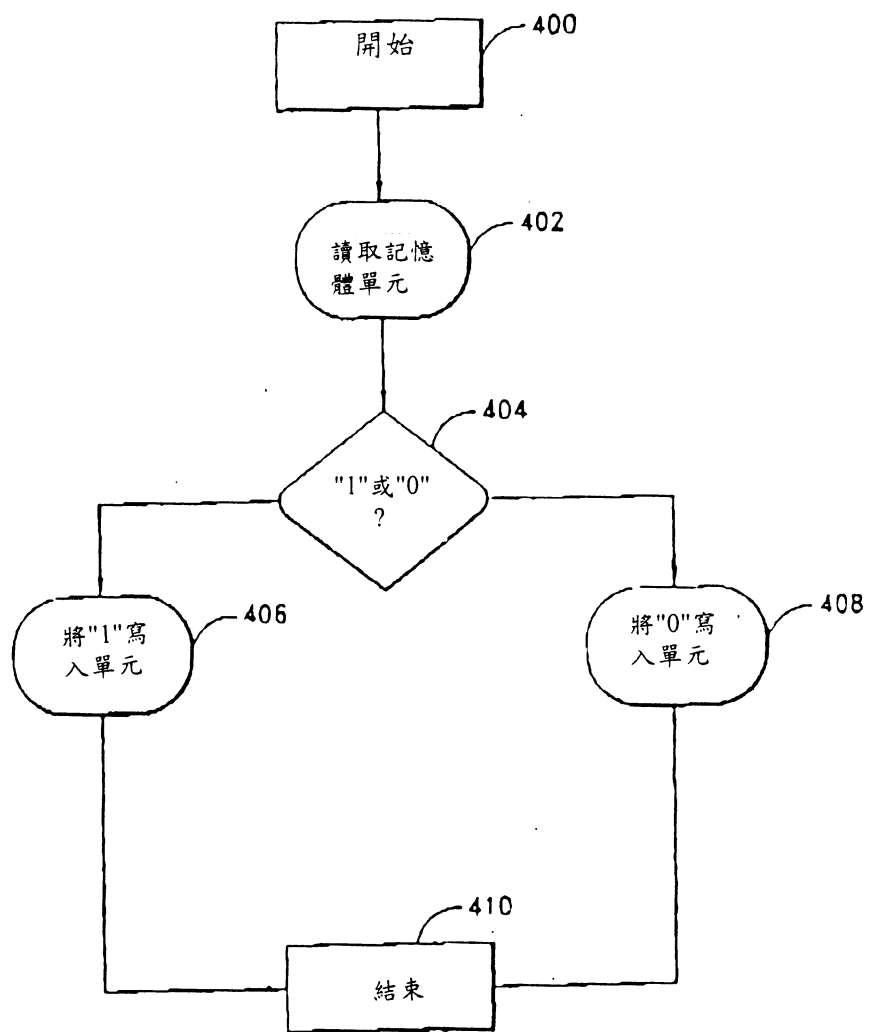


圖 3

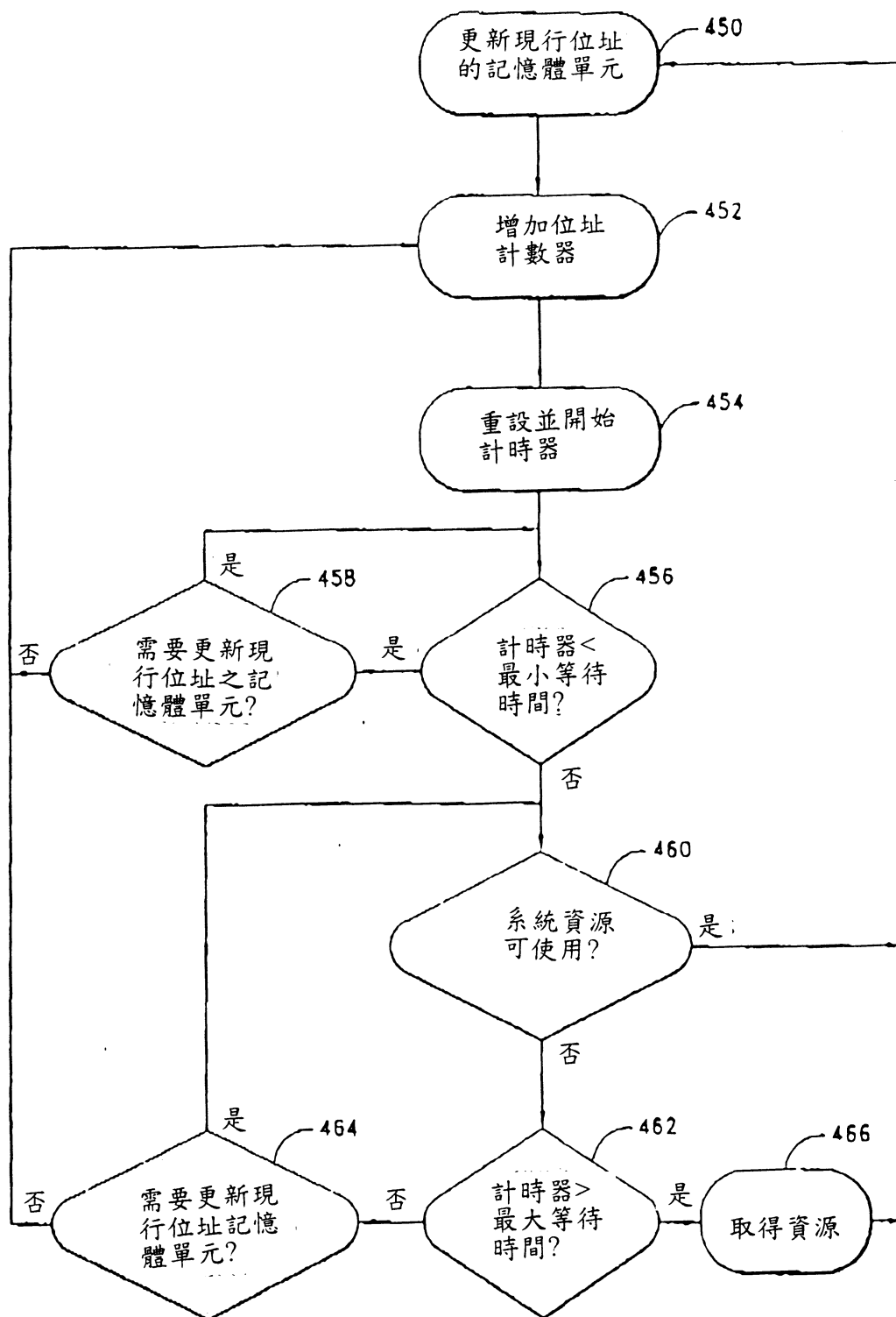


圖 4

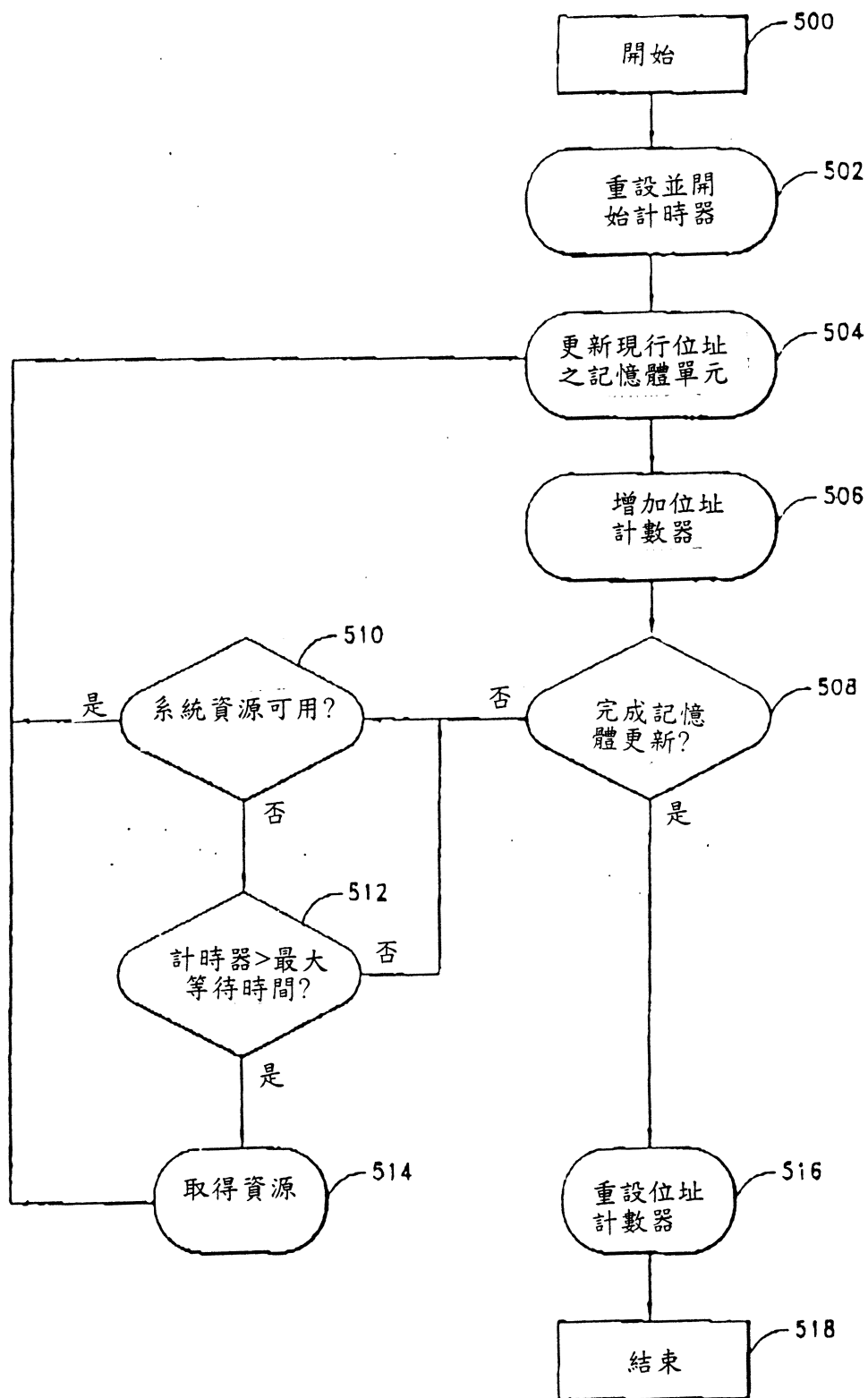


圖 5

柒、指定代表圖：

(一)本案指定代表圖為：第(2)圖。

(二)本代表圖之元件代表符號簡單說明：

- 250 電腦系統
- 260 記憶體裝置
- 270 處理器
- 280 匯流排
- 290 記憶體更新指令
- 300 位址/資料/控制模組
- 310 記憶體陣列
- 320 記憶體單元
- 330 列
- 340 行
- 350 線路

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

(無)



發明專利說明書

中文說明書替換頁(94年4月)



(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：092123963

※ 申請日期：92-08-29

※IPC 分類：G11C 13/00

壹、發明名稱：(中文/英文)

記憶體裝置及其更新方法

MEMORY DEVICE AND METHODS FOR REFRESHING THE SAME

貳、申請人：(共 1 人)

姓名或名稱：(中文/英文)

美商麥克隆科技公司

MICRON TECHNOLOGY, INC.

代表人：(中文/英文)

麥克 L. 林契

MICHAEL L. LYNCH

住居所或營業所地址：(中文/英文)

美國愛達荷州鮑西市南菲德洛路 8000 號

8000 SOUTH FEDERAL WAY, BOISE, IDAHO 83707-0006, U.S.A.

國 籍：(中文/英文)

美國 U.S.A.

參、發明人：(共 1 人)

姓 名：(中文/英文)

泰瑞 L. 吉爾頓

TERRY L. GILTON

住居所地址：(中文/英文)

美國愛達荷州鮑西市東自然大道 3149 號

3149 E. NATURE DRIVE, BOISE, IDAHO 83706, U.S.A.

國 籍：(中文/英文)

美國 U.S.A.

要時能更新每一記憶體單元，以避免資料遺失，通常至少每秒數次。

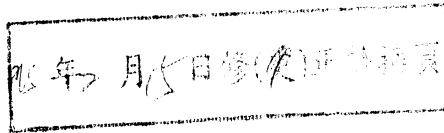
【發明內容】

於一實施例中，一種更新記憶體裝置中之一單元之方法，包括判斷該單元是否處於一寫狀態或處於一抹除狀態，及如果該單元正處於該寫狀態，藉由發出一更新寫指令給該單元，以更新該單元。

於另一實施例中，一種保護儲存於一具有複數個單元之揮發性記憶體裝置之資料之方法，包括對該複數個單元之每一個，判斷該單元是否需要被更新，及如果該單元需要被更新，則更新該單元。

於另一實施例中，一種保護儲存於一具有複數個單元之揮發性記憶體裝置之資料之方法，包括更新該複數個單元，及等待一段持續至少大約1秒之預定時間，其中所有的該複數個單元於該段預定時間沒有一個被更新。

於另一實施例中，一種保護儲存於一具有複數個記憶體單元之揮發性記憶體裝置之資料之方法，其包括定址一第一記憶體單元，從該第一記憶體單元被定址之後，等待一第一時段，及判斷該第一記憶體單元是否需要被更新。如果該第一記憶體單元需要被更新，該方法進一步包括判斷此等系統資源是否可用於更新該第一記憶體單元，及如果該第一記憶體單元需要被更新，且假設該等系統資源不是可用的，於該第一記憶體單元被定址之後的一第二時段內，監測該等系統資源是否變成可用於更新該第一記憶體單



元。如果該第一記憶體單元需要被更新，且假設該等系統資源於該第二時段內未變成可用的，則該方法進一步包括迫使該等資源被釋放，以使該等資源變成可用於更新該第一記憶體單元，及如果該第一記憶體單元需要被更新，使用該等可用的系統資源更新該第一記憶體單元。該方法進一步包括定址一第二記憶體單元。

於另一實施例中，一種避免遺失一揮發性記憶體裝置中之資料的方法，其包括根據必須被更新的該揮發性記憶體裝置，設立最後期限，及監測此等資源是否可用於更新該揮發性記憶體裝置。在該最後期限之前，如果此等資源在一第一預定時段內未變成可用於更新該揮發性記憶體裝置，該方法進一步包括迫使該等資源被釋放，以使該等資源變成可用於更新該揮發性記憶體裝置。該方法進一步包括在該最後期限之前，使用該等可用的資源更新該揮發性記憶體裝置。

於另一實施例中，一種電腦系統，其包括一處理器，一耦合該處理器之記憶體裝置，其中該記憶體裝置包括複數個需要被定期更新之單元，及一由該處理器執行更新該複數個單元之軟體模組。

於另一實施例中，一種包括複數個必須被定期更新之單元的記憶體裝置，其中為回應來自一處理器之一寫指令或一抹除指令，配置的該複數個單元被更新，而且其中未配置該記憶體裝置，產生或接收一不同於該寫指令或抹除指令之更新控制信號。

生一更新控制電路，導致該定址列180中之所有的記憶體單元170同時被更新。由於可同時更新大量的記憶體單元170，該記憶體更新電路150有助於縮短執行一記憶體更新作業所需要的時間量，因而縮短該記憶體裝置110在該處理器120執行其他作業時不可用的時間。

由於該記憶體裝置110在一記憶體更新作業期間不可用於執行其他作業，所以希望能安排此等記憶體更新作業時間，使此等更新作業發生在不會與來自該處理器120之需要相衝突的時間。因此，通常配置該記憶體更新電路150，以識別該處理器120不可能對該記憶體裝置110發出請求的此等時間，並且在該等識別的時間內執行此等記憶體更新作業。另一方面，因為執行一記憶體更新作業需要電源，所以希望將執行記憶體更新作業的次數減至最少。因此，通常也配置該記憶體更新電路150，盡可能少執行此等記憶體更新作業，以保護該資料儲存於該記憶體裝置110。

然而，為避免遺失儲存於慣用之揮發性記憶體裝置110之資料，該記憶體更新電路150必須頻繁地執行記憶體更新作業，通常每秒許多次。例如，於一些實施例中，各記憶體單元170至少大約每64毫秒必須被更新一次，以避免遺失資料。於此等實施例中，為避免遺失儲存於該記憶體裝置110之資訊，該記憶體更新電路150必須確保每一記憶體單元170更常被更新，至少超過每秒15次。

圖2說明根據本發明之一實施例之一具有一軟體更新之記憶體裝置260之電腦系統250。該電腦系統250也包括一經

拾、申請專利範圍：

1. 一種更新一記憶體陣列之一記憶體單元之方法，該方法包括：

為該記憶體陣列之複數個記憶體單元之每一個判斷該等複數個記憶體單元之每一個是否需要一更新；

判斷需要一更新之一給定單元是處於一寫入狀態或是處於一抹除狀態；及

如果該給定單元處於該寫入狀態，藉由發出一更新寫入指令給該單元而更新該單元；

自該給定單元被更新之後等待一第一時段，及

判斷該給定單元是否需要另一更新。

2. 如申請專利範圍第1項之方法，進一步包括：如果該單元處於該抹除狀態，藉由發出一更新抹除指令給該單元，以更新該單元。

3. 一種保存資料儲存於一具有複數個記憶體單元之揮發性記憶體裝置之方法，包括：

定址該等複數個記憶體單元之一第一個別記憶體單元；

自該第一個別記憶體單元被定址之後，等待一第一時段；

判斷該第一個別記憶體單元是否需要被更新；

如果該第一個別記憶體單元需要被更新，判斷此等系統資源是否可用於更新該第一個別記憶體單元；

如果該第一個別記憶體單元需要被更新，且假設該等

系統資源不可用，在一段該第一記憶體單元被定址之後的第二時間內，監測該等系統資源是否變成可用於更新該第一個別記憶體單元；

如果該第一個別記憶體單元需要被更新，且假設該等系統資源於該第二時段內未變成可用，則迫使該等資源被釋放，以使該等資源變成可用於更新該第一個別記憶體單元；及

如果該第一個別記憶體單元需要被更新，使用該等可用的系統資源更新該第一個別記憶體單元。

4. 如申請專利範圍第3項之方法，其中該第一時段的範圍是大約一秒至一星期。
5. 如申請專利範圍第3項之方法，其中該第二時段的範圍是一星期至兩個月。
6. 一種避免遺失一揮發性記憶體單元陣列中之資料的方法，包括：

一旦一記憶體單元被更新，為該給定記憶體單元重設一更新計時器，該計時器往下計時該記憶體單元需要再度被更新之前的剩餘時間；

監測此等資源是否可用於更新該給定記憶體單元；

在該剩餘時間中，如果此等資源在一第一頂定時段內未變成可用於更新該給定記憶體單元，則促使該等資源被釋放，以使該等資源變成可用於更新該給定記憶體單元；及

在該剩餘時間中，使用該等可用的資源更新該給定記

憶體單元。

7. 一種包括複數個必須被定期更新之單元的記憶體裝置，其中該等複數個單元係配置以經由來自一處理器之一寫入指令以及一抹除指令之一被更新，其中該處理器係組態以判斷該等複數個單元之每一個是否需要被更新其中該寫入指令在一第一更新操作及在一寫入操作中係為相同指令，其中該抹除指令在一第二更新操作及在一抹除操作中係為相同指令；且其中自從一第一個別記憶體被更新後該處理器亦經係組態以等待一第一時段及判斷該第一個別記憶體是否需要再度被更新。
8. 如申請專利範圍第7項之記憶體裝置，其中該記憶體裝置包括一可程式化之導體隨機存取記憶體單元。
9. 如申請專利範圍第7項之記憶體裝置，其中該複數個單元之每一個皆包括一可程式化之金屬化單元。
10. 如申請專利範圍第8項之記憶體裝置，其中該可程式化之導體隨機存取記憶體單元，包括：
 - 一具有上表面的單元主體，其中該單元主體包括一硫屬金屬離子玻璃；及
 - 兩配置在該上表面之電極，其中該等電極彼此有距離間隔開。
11. 如申請專利範圍第10項之記憶體裝置，其中該硫屬金屬離子玻璃包括鍺亞硒酸鹽，有一第IV群金屬溶解於其中。
12. 如申請專利範圍第10項之記憶體裝置，其中該硫屬金屬離子玻璃包括 $\text{Ag/Ge}_3\text{Se}_7$ 。

13. 一種保存儲存在具有複數個記憶體單元之一揮發性記憶體裝置中之資料之方法，該方法包括：

存取該等複數個記憶體單元之一第一個別記憶體單元；

存取該第一個別記憶體單元之後，等待一第一時段；

判斷該第一時段是否需要被更新；及

如果該第一個別記憶體單元需要被更新，在該第一時段後更新該第一個別記憶體單元。

14. 如申請專利範圍第13項之方法，其中該定址該等複數個記憶體單元之該第一個別記憶體單元之步驟係用於一記憶體陣列之該等複數個記憶體單元之每一個。

15. 一種具有複數個必須被定期更新之記憶體單元的記憶體裝置，該記憶體裝置包括：

一處理器，其係耦合至該等複數個記憶體單元；

其中該處理器係組態以在該第一個別記憶體單元被存取之後等待一第一時段，並判斷該第一個別記憶體單元是否需要被更新；及如果判斷該第一個別記憶體單元需要被更新，在該第一時段之後更新該第一個別記憶體單元。

16. 一種具有複數個必須被定期更新之記憶體單元的記憶體裝置，該記憶體裝置包括：

一處理器，其係耦合至該等複數個記憶體單元，用以更新該等複數個記憶體單元之一第一個別記憶體單元及一旦該第一個別記憶體單元已被更新，為該第一個別記憶體單元重新設定一更新計時器，該更新計時器在該第

一個別記憶體單元應再被更新之前倒數計時剩餘之時間。