

電阻式非揮發性8T SRAM之憶阻器相關錯誤模型化、測試與診斷

Modeling, Testing, and Diagnosis of Memristor-Related Faults in Resistive Nonvolatile-8T SRAMs

李昱霆 陳詠孝 李進福 吳冠德 駱致彥 蒯定明 周永發
Yu-Ting Li, Yong-Xiao Chen, Jin-Fu Li, Kuan-Te Wu, Chih-Yen Lo, Ding-Ming Kwai, Yung-Fa Chou

中文摘要

電阻式非揮發性八電晶體靜態隨機存取記憶體已經被提出用以減輕靜態功率、電源關閉時保留資料，且提供快速開機的速度。在本文中，我們定義數個針對非揮發性八電晶體靜態隨機存取記憶體的錯誤模型化，其中包含新的憶阻器相關錯誤模型。另外提出了一類行軍式測試演算法可涵蓋簡單的靜態隨機存取記憶體的錯誤與憶阻器相關的錯誤。在16kb之待測記憶體下，所提出的類行軍式測試演算法與現有的方法相比雖然需要25.9%的額外測試時間，但在這些探討的錯誤模型下可提供100%的測試涵蓋率。另外，我們亦提出類行軍式診斷測試，以區分非揮發性八電晶體靜態隨機存取記憶體中憶阻器相關的錯誤。

Abstract

Resistive nonvolatile-8T (Rnv8T) static random access memory (SRAM) has been proposed to alleviate static power and preserve data in power-down mode and provide fast power-on speed. In this paper, we define several faults for the Rnv8T SRAM including new memristor-related fault models. Also, a March-like test algorithm which can cover simple SRAM faults and defined memristor-related faults are proposed. Considering 16kb memory under test, the proposed March-like test needs 25.9% test time overhead as compare with existing test algorithm, but provides 100% fault coverage on the targeted faults. Furthermore, we also propose a March-like diagnosis test for distinguishing memristor-related faults of Rnv8T SRAMs.

關鍵詞(Key Words)

憶阻器 (Memristor)
非揮發性靜態隨機存取記憶體 (Nonvolatile SRAM)
錯誤模型化 (Fault model)
測試演算法 (Test algorithm)
行軍式測試 (March test)

1 · 前言

對於移動或可穿戴的電子設備，減少待機功耗是很重要的。常見減少待機功耗的做法之一，可利用非揮發性記憶體在電源關閉前保存

資料，然後將電源關閉。需要使用時可開機回復到常規模式。為了進一步提升效能，將非揮發性元件整合至SRAM能夠結合非揮發性元件的非揮發性與SRAM快速存取的優點[1]、[2]。由於與CMOS的製程相容性高，電阻式非揮發

性元件被視為是一個好的候選者[3]-[9]。許多電阻式非揮發性SRAM已經被提出[1]，[10]-[14]。該記憶體細胞包含了一個SRAM細胞與電阻式非揮發性(憶阻器)元件。其中，又以電阻式非揮發性八電晶體(Resistive nonvolatile 8T, Rnv8T) SRAM能提供較多的優點。

Rnv8T SRAM的測試應考慮憶阻器相關的錯誤模型。憶阻器記憶體的瑕疵分析和錯誤模型化在[15]、[16]、[19]被討論。文獻[17]針對交叉結構的憶阻器提出利用潛行路徑的測試方法。然而，用於SRAM與憶阻器記憶體的錯誤模型與測試方法並不能直接用於Rnv8T SRAM。在文獻[18]中，有針對Rnv8T SRAM進行瑕疵分析的調查，並提出測試演算法。然而，該文獻僅探討後段製程(Back end of line, BEOL)的瑕疵，且測試演算法不能完全涵蓋SRAM的藕合錯誤。在本文中，我們考慮可能的電氣瑕疵進行分析，並定義Rnv8T SRAM中憶阻器相關的錯誤模型，其中提出兩種新型憶阻器相關錯誤模型。然後，我們提出測試演算法以涵蓋簡單的SRAM與所定義的錯誤模型。同時亦提出診斷演算法，以區分Rnv8T SRAM中憶阻器相關的錯誤。而所提出之測試演算法雖然需要付出額外的測試時間，但可以提高憶阻器相關錯誤及簡單SRAM錯誤之錯誤涵蓋率來間接提高測試品質。

這篇論文的編排如下：第二節簡述憶阻器的特性以及Rnv8T SRAM；第三節說明所定義的憶阻器相關的錯誤；第四節描述我們所提出的測試與診斷演算法；第五節顯示模擬與分析結果；最後總結本文。

2. 背景

2.1 憶阻器(Memristor)

憶阻器是一個阻值會隨累積電荷改變的電阻[20]、[21]。而易阻器的特性可以用下列的公式來表示：

$$d\phi = M(q) \times dq$$

$$v(t) = M(q) \times i(t)$$

其中， ψ 、 M 、 q 、 v 、 i 分別代表通量、憶阻器阻值、電荷、憶阻器跨壓與流經電流。該憶阻

器的電阻值由兩端點的跨壓與供應的時間長度的決定。根據憶阻器的特性，一個理想的憶阻器可以用兩個串聯的電阻當作模型，如圖1所示。假設憶阻器的厚度為 D 。憶阻器包含了厚度為 w 的參雜電阻與厚度為 $(D-w)$ 的未參雜電阻。兩區域的邊界會根據憶阻器兩端點的跨壓與跨壓提供的時間，因參雜物的漂移而改變。因此，憶阻器的電阻值可以用下列式子表示[21]-[23]：

$$R_{mem} = R_{on}X + R_{off}(1 - X)$$

$$X = \frac{w(t)}{D} \in (1,0)$$

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D} i(t)$$

其中， R_{mem} 代表憶阻器電阻值； R_{on} 和 R_{off} 分別代表憶阻器完全參雜與完全未參雜的電阻值； X 代表參雜區域對憶阻器厚度的比例； μ_v 代表參雜物的移動率。

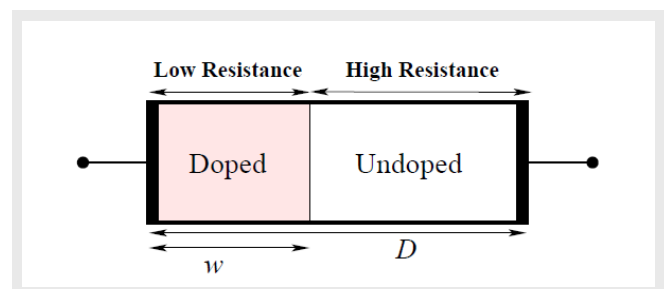


圖1 憶阻器之電阻模型 [21]

2.2 電阻式非揮發性八電晶體靜態隨機存取記憶體(Rnv8T SRAM)之細胞

圖2所示為一Rnv8T SRAM之細胞，其中包含了一個六電晶體的靜態隨機存取記憶體、兩個憶阻器(R_L 與 R_R)與兩個提供給憶阻器使用之存取電晶體(M_2 與 M_3)。該記憶體細胞有四種功能性操作：讀、寫、儲存、還原四種操作。讀寫操作與一般6T SRAM相同。儲存操作可以將6T SRAM細胞中的資料寫入憶阻器中。還原操作可將憶阻器的資料寫入至6T SRAM的細胞中。因此，Rnv8T SRAM在正常模式下可如同6T SRAM進行存取。在進入待機模式之前，Rnv8T SRAM細胞會執行儲存操作，將6T

SRAM細胞中的資料寫入至憶阻器。然後，該記憶體的供應電壓可以關閉，如此一來該記憶體不會有靜態功率消耗。接著，如果供應電壓被打開，Rnv8T SRAM細胞會執行還原操作，將儲存在憶阻器的資料寫入至6T SRAM的細胞。然後，該記憶體可以操作在正常模式。一般來說，為了要減少額外的時間，儲存與還原

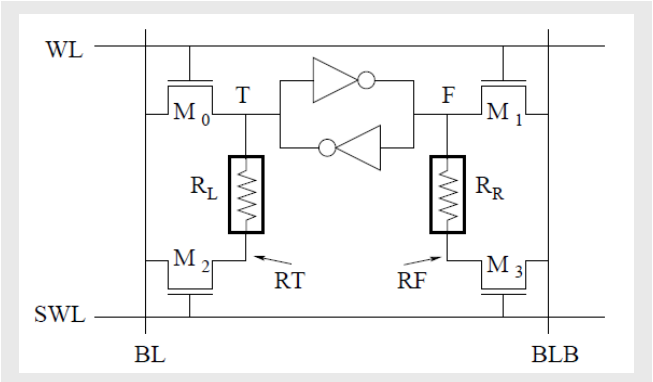


圖2 電阻式非揮發性八電晶體靜態隨機存取記憶體之細胞結構 [1]

操作是以區塊的方式平行執行。

表1總結了Rnv8T SRAM細胞在不同操作被執行的偏壓條件[1]。若執行寫的動作時，WL與SWL會升至 V_{DD} ， M_2 與 M_3 會導通以提供平行寫入的路徑提升寫入邊界。若執行讀的動作時，WL會升至 V_{DD} 而SWL會降為 V_{SS} ，以增加讀取穩定度。

當執行儲存動作時，WL會降為 V_{SS} 而SWL會升至 V_{DD} ，將存於SRAM細胞內的資料寫至憶阻器中，而儲存動作可以分為兩個階段：Store-set與Store-reset。當Store-set執行時，BL與BLB均會升至 V_{DD} 。反之，當Store-reset執行時，BL與BLB均會降至 V_{SS} 。若SRAM的資料為1時，在Store-set時會因電流由BLB流向F，使得 R_R 的電阻變成 R_{on} 。在Store-reset時會因電流由T流向BL，使得 R_L 的電阻變成 R_{off} 。若SRAM的資料為0時，在Store-set與Store-reset時，分別使得 R_L 與 R_R 的電阻會被改變為 R_{on} 與 R_{off} 。當執行還原動作時，WL會降為 V_{SS} 而SWL會升至 V_{DD} ，BL與BLB會降至 V_{SS} 以清除在T/F兩點的殘存電荷使兩點等電位。接著，細胞陣列的供應電壓被提升，T/F兩點會藉由PMOS充電。由於 R_L 與 R_R 電阻不同，在電源打開的過程T/F兩點的放電速度不同，使得T/F兩點會有電壓

差。當憶阻器為低(高)阻抗時，可提供大(小)的放電電流，造成較低(高)的節點電壓。因此當憶阻器的電阻值被改變為 $R_{on}(R_{off})$ 時，憶阻器的狀態可視為邏輯0(邏輯1)。

表1 Rnv8T SRAM不同操作之偏壓條件

Operation	WL	SWL	BL	BLB
Write	V_{DD}	V_{DD}	-	-
Read	V_{DD}	V_{SS}	-	-
Store-set	V_{SS}	V_{DD}	V_{DD}	V_{DD}
Store-reset	V_{SS}	V_{DD}	V_{SS}	V_{SS}
Restore	V_{SS}	V_{DD}	V_{SS}	V_{SS}

3 · 細胞層級錯誤模型

3.1 電氣瑕疵與模擬模型

在驗證的錯誤模型的錯誤行為部分，我們使用電路層級模擬器(Hspice)並注入下列瑕疵。1) 電源短路:表示有不必要的阻抗出現在電路節點及 V_{DD} 或 V_{SS} 之間。2) 節點短路:表示有不必要的阻抗出現在兩個電路節點之間。3) 電晶體導通:不論給定任何控制訊號，電晶體皆保持導通狀態，在此我們將NMOS (PMOS)之閘極接至 V_{DD} (V_{SS})以模擬電晶體導通。4) 電晶體開路:不論給定任何控制訊號，電晶體皆保持開路狀態，在此我們將NMOS (PMOS)之閘極接至 V_{SS} (V_{DD})以模擬電晶體開路。

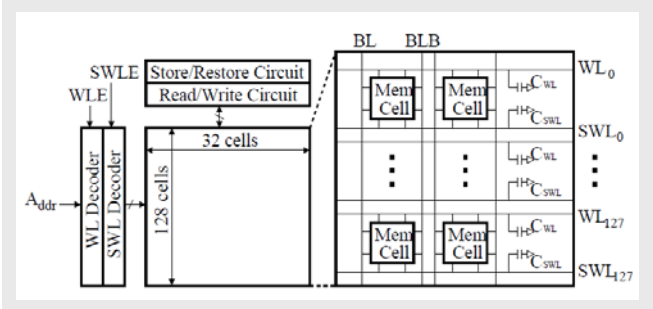


圖3 HSPICE之模擬模型

圖3所示為Hspice之模擬模型，在此論文中，我們使用之憶阻器行為模型發表於[23]，此行為模型包含了憶阻器之臨界電壓限制可使此模型更加接近真實雙極性憶阻器之特性。我們使用一128x2之Rnv8T SRAM陣列加上電容負載於WL及SWL來模擬128x32之Rnv8T SRAM陣列。在此分別使用128個Rnv8T SRAM模擬於兩行，理由為在執行儲存及還原動作時

整個記憶體區塊會同時進行動作，因此，當執行儲存及還原動作時，所有SWL將通時開啟，此時BL及BLB之負載將與憶阻器之阻值相關。我們使用之 R_{on} 及 R_{off} 阻值為 $10K\Omega$ 及 $100K\Omega$ 。

表2 對於Rnv8T SRAM細胞可能的瑕疵

Name	Defect	Symmetric Defect
D_0	M_2 stuck-on	M_3 stuck-on
D_1	M_2 stuck-open	M_3 stuck-open
R_{S0}	RT- V_{DD} short	RF- V_{DD} short
R_{S1}	RT- V_{SS} short	RF- V_{SS} short
R_{S2}	SWL- V_{DD} short	
R_{S3}	SWL- V_{SS} short	
R_{O0}	T- R_L open	F- R_R open
R_{O1}	RT- R_L open	RF- R_R open
R_{B0}	RT-T bridge	RF-F bridge
R_{B1}	RT- R_L bridge	RF- R_R bridge
R_{B2}	RT-T bridge	RF-F bridge
R_{B3}	RT-F bridge	RF-T bridge
R_{B4}	RT-BL bridge	RF-BLB bridge
R_{B5}	RT-BLB bridge	RF-WL bridge
R_{B6}	RT-WL bridge	RF-SWL bridge
R_{B7}	RT-SWL bridge	SWL-BLB bridge
R_{B8}	SWL-T bridge	SWL-F bridge
R_{B9}	SWL-WL bridge	

表2彙整了模擬之瑕疵。可能發生於傳統SRAM並造成錯誤之瑕疵不在此處討論。如圖2所示，Rnv8T SRAM為對稱之架構，因此，列於右方欄位的對稱性瑕疵無另外進行模擬。列於表2中之部分瑕疵已被模擬及分析於文獻[18]，但僅考慮後段製程所產生之瑕疵。因此，在此論文中，我們考慮了更加完整之瑕疵來分析與憶阻器相關之錯誤行為。

3.2 電氣錯誤模型

在此段落中，我們使用圖2所示之Rnv8T SRAM來定義電阻性非揮發性靜態隨機存取式記憶體之憶阻器相關錯誤模型。在此使用 x 及 y 來表示儲存於SRAM的資料(T,F)以及憶阻器(R_L, R_R)，若(T,F)=(0,1)或(1,0)則 $x=0$ 或 $x=1$ 。若(R_L, R_R)=(0,1)或(1,0)則 $y=0$ 或 $y=1$ 。

3.2.1 憶阻器固定型錯誤 (Memristor Stuck-at Fault, MSAF)

若SRAM再執行restore- y 的動作之後 x 值恆為 k ，並且 $k \in \{0,1\}$ 則稱此Rnv8T SRAM發生

MSAF[18]。MSAF可被分為MSA0F以及MSA1F。兩種可能的瑕疵會造成MSAF，第一種為會造成憶阻器之等效阻值會恆大於另一端之憶阻器之瑕疵，例如一電阻性開路瑕疵發生於節點T及憶阻器 R_L 之間(等同於表二之 R_{O0})，若此瑕疵之阻值大於 R_{off} 則會發生MSA1F。相反的若有電阻式節點短路瑕疵發生於節點T及節點RT之間(等同於表二之 R_{B0})，則會造成MSA0F，原因為若 R_{B0} 之阻值小於 R_{on} 則會造成節點T及RT之間的等效組直恆小於 R_R 。

3.2.2 儲存延遲錯誤 (Slow Store Fault, SSF)

若一Rnv8T SRAM執行store- x 動作時在定義的儲時時間內無法將憶阻器轉變至目標阻值則稱此Rnv8T SRAM發生SSF[18]。例如，一電阻式開路瑕疵存在於節點T及憶阻器 R_L 之間(R_{O0})，造成在執行儲存動作時憶阻器之跨壓小於應有之電壓值使得憶阻器 R_L 之阻值發生偏差，導致執行還原動作時無法還原正確的值回到SRAM。SSF可被分為SS0F以及SS1F。若一Rnv8T SRAM發生SSF時，成功的儲存動作依然能夠儲存正確的值進入到憶阻器，原因在於憶阻器的阻值與執行儲存動作時的操作時間及憶阻器跨壓有關。

3.2.3 破壞性儲存錯誤 (Store Destructive Fault, SDF)

若一Rnv8T SRAM執行store- x 動作時，SRAM的資料發生翻轉的狀況，則稱此Rnv8T SRAM發生SDF。例如若有一電阻性節點短路瑕疵存在於節點T及節點RT之間且此瑕疵之阻值小於一特定值(在我們的電路模擬中 $<9K\Omega$)，造成節點T在執行儲存動作時的電壓小於應有之值導致SRAM之資料發生翻轉的情況。SDF在 $x=0$ 及 $x=1$ 時可被分為SD0F及SD1F。圖4所示為當瑕疵 R_{B0} 發生於Rnv8T SRAM時之模擬波型樣本，SWLE表示SWL批准之訊號， R_L 及 R_R 表示憶阻器 R_L 及 R_R 之阻值。如波型所示，write-1動作先將資料1寫入SRAM之中使得 $T=1$ 。緊接著執行store-1之動作，由於瑕疵存在之關係，SRAM的資料在執行store-reset動作時發生翻轉($T=0$)。

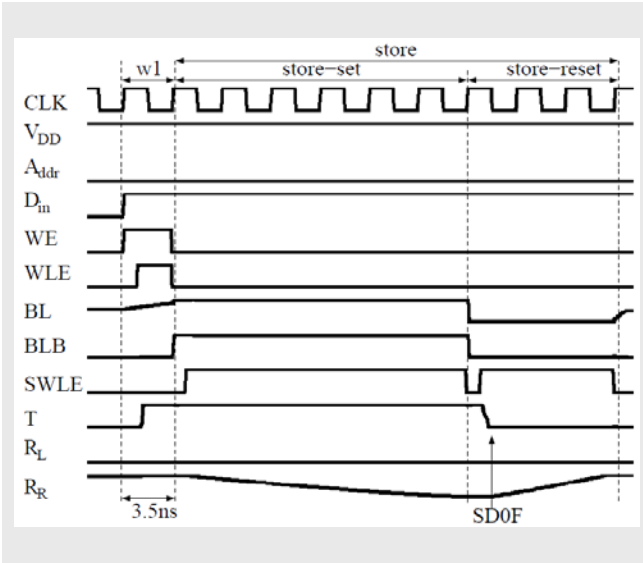


圖4 SDF之模擬波型樣本

3.2.4 憶阻器干擾讀取錯誤 (Memristor Disturb Read Fault, MDRF)

若一Rnv8T SRAM執行read-x動作時回傳錯誤的值，此時在相同行內之另一Rnv8T SRAM細胞之SRAM及憶阻器之資料皆為x'，則稱此Rnv8T SRAM發生MDRF。MDRF可被區分為MDR0F及MDR1F。假設在k行中之一Rnv8T SRAM細胞($cell_j$)發生 M_2 電晶體開路瑕疵(D_0)且存於此細胞之SRAM及憶阻器資料皆為0 ($R_L=R_{on}$, $R_R=R_{off}$)。若相同k行之中的另一Rnv8T SRAM細胞($cell_i$)執行read-1動作，則此動作將回傳錯誤之資料，因此稱 $cell_i$ 發生MDR1F。

圖5所示為一MDRF之模擬波型樣本，我們有位址A0之細胞注入 M_2 電晶體導通(D_0)之瑕疵。 T_0 及 T_1 表示儲存於位址A0及A1的SRAM之資料， R_{0L} 表示在為址A0之憶阻器 R_L 之阻值，存於位址A0憶阻器之初始資料為0 ($R_L=R_{on}$, $R_R=R_{off}$)。如波型所示，write-0及write-1動作先將資料0及資料1寫入位址A0及A1之SRAM中，接著執行read-0及read-1動作至位址A0及A1，可以看出read-0動作可以正確地讀出位址A0之資料($D_{out}=0$)，但緊接著在位址A1的read-1動作卻無法讀出正確的資料(D_{out} 依舊為0無改變)，原因為當位址A1執行read-1動作時，BL及BLB將被預先充電(precharge)至 V_{DD} ，接著WL開啟之後，在一般情況下，由於節點T及節點F之值為1及0，BL會維持在 V_{DD} ，BLB會被放電至0，但

由於 M_2 電晶體導通之瑕疵，BL也經由A0的節點T同時放電，由於BL及BLB同時放電，讀出之資料將發生錯誤，在此例子中，執行於位址A1之read-1動作讀出的資料為0。

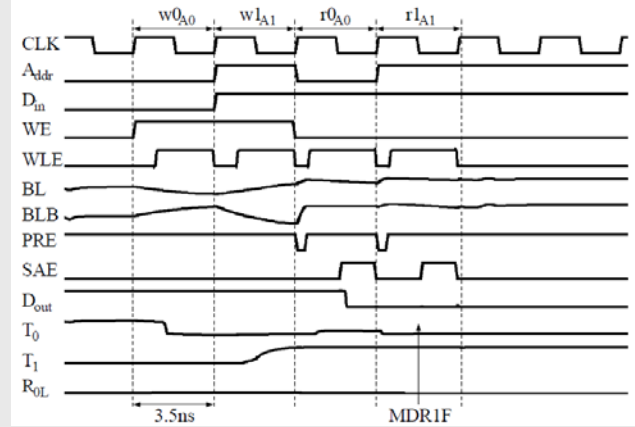


圖5 MDRF之模擬波型樣本

4 · 類行軍式測試

4.1 測試演算法

由於Rnv8T SRAM由6電晶體SRAM及憶阻器組成，故在測試時，SRAM相關之錯誤及所定義之錯誤都需要列入考慮。因此，我們結合了所定義之錯誤之測試需求以及現存測試SRAM相關錯誤之類型軍式演算法，提出了一個類型軍式測試March CM來涵蓋SRAM相關的錯誤以及定義之憶阻器相關錯誤。

$$\left\{ \begin{array}{l} (\uparrow_{even}(w0); \uparrow_{odd}(w1); (store); \\ \uparrow_{even}(w1); \uparrow_{odd}(w0); (store);)^n \\ \uparrow_{even}(w0); \uparrow(r0, w1); \uparrow_{even}(r1, w0); (store); \\ \uparrow_{odd}(r1, w0); \uparrow(r0); (poff); \\ (restore); \uparrow_{odd}(r1); \uparrow_{even}(r0); \\ \uparrow_{odd}(w0); \downarrow(r0, w1); \downarrow_{odd}(r1, w0); (store); \\ \downarrow_{even}(r1, w0); \uparrow(r0); (poff); \\ (restore); \uparrow_{even}(r1); \uparrow_{odd}(r0); \end{array} \right\}$$

在此rk以及wk表示一個read-k及write-k動作且k=0或1。Store、restore及poff動作為一次執行於所有的細胞之動作。 $\uparrow$ 、 $\downarrow$ 及 $\uparrow_{even}$ 表示位址序列為遞增順序、遞減順序及皆可。 $\uparrow_{even}$ 表示於偶數位址之遞

增順序(0, 2, 4, 6, ..., N-2) · $\uparrow_{\text{odd}}$ 表示於奇數為址之遞增順序(1, 3, 5, 7, ..., N-1)。相反的, $\downarrow_{\text{even}}$ 及 $\downarrow_{\text{odd}}$ 表示於偶數及奇數位址之遞減順序。

March CM由自我組成(self-forming)序列開始, 此n次的寫入及儲存動作可解決過度組成(over-forming)問題並且可將憶阻器之組值調整在工作區間內[23], 而n之次數需經由憶阻器之物理特性來決定。其中的測試元素(test element) $\uparrow(r0, w1)$ 、 $\uparrow_{\text{even}}(r1, w0)$ 、 $\uparrow_{\text{odd}}(r1, w0)$ 、 $\downarrow(r0, w1)$ 、 $\downarrow_{\text{odd}}(r1, w0)$ 及 $\downarrow_{\text{even}}(r1, w0)$ 用於觸發等冪耦合錯誤(Idempotent coupling fault)、反向耦合錯誤(Inversion coupling fault)及狀態耦合錯誤(State coupling fault)之測試元素。在奇數及偶數位址之測試元素可產生棋盤式(checkerboard)測試向量(test pattern)在此用於觸發及偵測MDRF。在 $\uparrow(r0, w1)$ 中之w1及 $\uparrow_{\text{even}}(r1, w0)$ 之w0動作可將SRAM之初始值設至棋盤式向量(010101...)而接續之(store)動作可以觸發位於偶數位址之MSA1F、SS0F及SD1F與奇數位址之MSA0F、SS1F及SD0F。接下來的(poff)及(restore)動作可將憶阻器之資料回復到SRAM細胞中。因此, $\downarrow_{\text{odd}}(r1)$ 可觀測到位於奇數位址之MSA0F、SS1F、SD0F及MDR1F的錯誤現象(fault effect)。 $\downarrow_{\text{even}}(r0)$ 可以觀測到位於偶數位址之MSA1F、SS0F、SD1F及MDR0F的錯誤現象。相同的, 位於偶數位址之MSA0F、SS1F、SD0F及MDR1F與位於奇數位址之MSA1F、SS0F、SD1F及MDR0F可經由 $\downarrow(r0, w1)$ 、 $\downarrow_{\text{odd}}(r1, w0)$ 、(store)、(poff)、(restore)、 $\downarrow_{\text{even}}(r1)$ 及 $\downarrow_{\text{odd}}(r0)$ 測試元素來偵測。

4.2 診斷演算法

針對憶阻器相關之錯誤(MSAF、SSF、SDF及MDRF)之診斷演算法表示於下方。

$$\left\{ \begin{array}{l} (\downarrow_{\text{even}}(w0); \downarrow_{\text{odd}}(w1); (store); \\ \downarrow_{\text{even}}(w1); \downarrow_{\text{odd}}(w0); (store);)^n \\ \downarrow_{\text{even}}(w0); \downarrow_{\text{odd}}(w1); (store); \downarrow_{\text{even}}(r0); \\ \downarrow_{\text{odd}}(r1); (poff); (restore); \downarrow_{\text{even}}(r0); \\ \downarrow_{\text{odd}}(r1); \downarrow_{\text{even}}(w0); (poff); \\ (restore); \downarrow_{\text{even}}(r0); \downarrow_{\text{odd}}(r1); \\ (store); (poff); (restore); \downarrow_{\text{even}}(r0); \downarrow_{\text{odd}}(r1); \\ \downarrow_{\text{even}}(w1); \downarrow_{\text{odd}}(w0); (store); \downarrow_{\text{even}}(r1); \\ \downarrow_{\text{odd}}(r0); (poff); (restore); \downarrow_{\text{even}}(r1); \\ \downarrow_{\text{odd}}(r0); \downarrow_{\text{even}}(w1); (poff); \\ (restore); \downarrow_{\text{even}}(r1); \downarrow_{\text{odd}}(r0); \\ (store); (poff); (restore); \downarrow_{\text{even}}(r1); \downarrow_{\text{odd}}(r0); \end{array} \right\}$$

表4表示為針對憶阻器相關錯誤的診斷演算法之錯誤辭典, 表內之e及o表示在偶數位址及奇數位址之錯誤現象可經由診斷演算法之read動作觀察, E_0 至 E_{15} 表示於診斷演算法中的各個read動作(E_0 表示第1個read動作, E_1 表示第2個read動作, 依此類推)。

為了要區分MASF及SSF, 對於相同資料之儲存動作須執行兩次。SS0F及SS1F會造成在執行第一次儲存動作時的資料固定於原始之值並可經由 E_2 、 E_3 、 E_{10} 及 E_{11} 觀察錯誤現象。在第二次儲存動作執行之後, 發生SS0F及SS1F之錯誤細胞之憶阻器可被操作至正確之狀態而發生MSAF之資料依舊被固定於錯誤之狀態。SDF部分可經由儲存動作之後之read動作(E_0 、 E_1 、 E_8 及 E_9)來偵測。在MDRF部分, 經由棋盤式資料之所有read動作皆可觀察到MDRF之現象。

表4 診斷演算法之錯誤辭典 (e: even; o: odd)

錯誤型態	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}
MSA0F	0	0	0	o	0	0	0	o	0	0	e	0	0	0	e	0
MSA1F	0	0	e	0	0	0	e	0	0	0	0	o	0	0	0	o
SS0F	0	0	e	0	0	0	0	0	0	0	0	o	0	0	0	0
SS1F	0	0	0	o	0	0	0	0	0	0	e	0	0	0	0	0
SD0F	0	o	0	o	0	0	0	0	e	0	e	0	0	0	e	0
SD1F	e	0	e	0	0	0	e	0	0	o	0	o	0	0	0	o
MDR0F	e	0	e	0	e	0	e	0	0	o	0	o	0	o	0	o
MDR1F	0	o	0	o	0	o	0	o	e	0	e	0	e	0	e	0

5. 實驗結果與分析

一般而言，March C-演算法常被用來測試簡單SRAM之錯誤，然而，此演算法在Rnv8T SRAM中僅可涵蓋小部分憶阻器相關之錯誤，因此我們比較了可偵測憶阻器相關錯誤之演算法March CM以及[18]所提出之演算法。表5彙整了此兩個演算法於定義之錯誤及簡單SRAM之錯誤之錯誤涵蓋率。[18]之演算法包含 $\uparrow(r0, w1)$ 及 $\downarrow(r1, w0)$ ，僅可涵蓋半數之inversion coupling faults (CF_{in}) 及 idempotent coupling faults (CF_{id})。另外，[18]之演算法並未輸入棋盤式資料，故MDRF無法被完全偵測。

表5 測試演算法之錯誤涵蓋率比較

錯誤	測試演算法	
	[18]	March CM
SAF	100%	100%
CF_{st}	100%	100%
CF_{id}	50%	100%
CF_{in}	50%	100%
MSAF	100%	100%
SSF	100%	100%
SDF	100%	100%
MDRF	96.875%	100%

考慮16K-位元、32K-位元及64K-位元之待測Rnv8T SRAM。圖6所示為[18]、March C-及March CM之測試時間比較結果。在此我們假設read/write、儲存、回復、及power off之時間為3.5ns、31.5ns、5ns、及20ns。在此self-forming序列之時間不包含於測試時間的計算之內。如圖所示，[18]演算法之測試時間接近於March C-之測試時間。然而，[18]之演算法僅能涵蓋部分之憶阻器相關錯誤及SRAM相關錯誤。而March CM需要比較長的測試時間，但可提供100%之憶阻器相關錯誤及簡單SRAM錯誤之錯誤涵蓋率。然而，提高錯誤涵蓋率可以較低的成本間接的提高測試品質。

圖7為提出之診斷演算法在不同記憶體配置下之測試時間，由圖可以看出所需之測試時間隨著記憶體之大小成長。

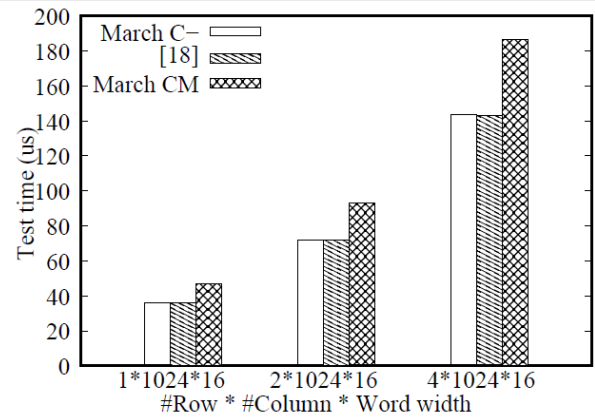


圖6 不同Rnv8T SRAM組態之測試時間比較

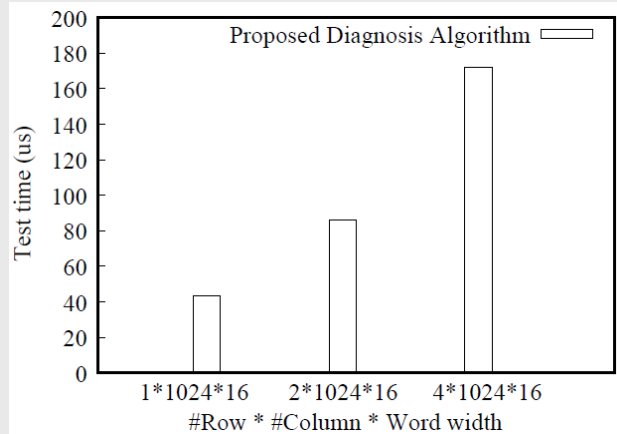


圖7 不同Rnv8T SRAM組態之診斷時間比較

5. 結論

在本文中，我們針對Rnv8T SRAM定義了憶阻器相關的錯誤模型其中包含新的憶阻器相關錯誤模型，同時再考慮了電氣瑕疵使用電路模擬工具HSPICE驗證定義的錯誤模型之行為。我們亦提出測試演算法以涵蓋所定義的憶阻器相關錯誤與簡單的靜態隨機存取記憶體的錯誤。在16kb之待測記憶體下，所提出的測試與現有的技術相比雖需要25.9%的額外測試時間，但在這些探討的錯誤模型下可提供100%的錯誤涵蓋率。

參考文獻

- [1] P.-F. Chiu, M.-F. Chang, C.-W. Wu, C.-H. Chuang, S.-S. Sheu, Y.-S. Chen, and M.-J. Tsai, "Low store energy, low VDDmin,

- 8T2R nonvolatile latch and SRAM with vertical-stacked resistive memory (memristor) devices for low power mobile applications,” *IEEE Jour. of Solid-State Circuits*, vol. 47, no. 6, pp. 1483–1496, June 2012.
- [2] C. E. Herdt, “Nonvolatile SRAM—the next generation,” in *Nonvolatile Memory Technology Review*, 1993, pp. 28–31.
 - [3] H.-S. P. Wong, H.-F. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F.-T. Chen, and M.-J. Tsai, “Metal-oxide RRAM,” *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, May 2012.
 - [4] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie, “Design implications of memristor-based RRAM cross-point structures,” in *Proc. Conf. Design, Automation, and Test in Europe (DATE)*, 2011, pp. 1–6.
 - [5] Y.-Y. Liauw, Z. Zhang, W. Kim, A. E. Gamal, and S. S. Wong, “Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory,” in *Proc. IEEE Int’l Solid-State Cir. Conf. (ISSCC)*, 2012, pp. 406–408.
 - [6] W. Fei, H. Yu, W. Zhang, and K.-S. Yeo, “Design exploration of hybrid CMOS and memristor circuit by new modified nodal analysis,” *IEEE Trans. on VLSI Systems*, vol. 20, no. 6, pp. 1012–1025, May 2011.
 - [7] S.-S. Sheu, M.-F. Chang, K.-F. Lin, C.-W. Wu, Y.-S. Chen, P.-F. Chiu, C.-C. Kuo, Y.-S. Yang, P.-C. Chiang, W.-P. Lin, C.-H. Lin, H.-Y. Lee, P.-Y. Gu, S.-M. W. an F.-T. Chen, K.-L. Su, C.-H. Lien, K.-H. Cheng, H.-T. Wu, T.-K. Ku, M.-J. Kao, and M.-J. Tsai, “A 4Mb embedded SLC resistive-RAM macro with 7.2ns read-write random-access time and 160ns MLC-access capability,” in *Proc. IEEE Int’l Solid-State Cir. Conf. (ISSCC)*, 2011, pp. 200–202.
 - [8] A. Chen, M. Meneghini, D. V. Blerkom, F. Schanovsky, and T. Shaw, “Mechanisms and performance of metal oxide resistive RAM (RRAM),” in *Proc. IEEE Int’l Integrated Reliability Workshop (IRW)*, 2013, pp. 187–189.
 - [9] B. Mohammad, D. Homouz, and H. Elgabra, “Robust hybrid memristor-CMOS memory: modeling and design,” *IEEE Trans. on VLSI Systems*, vol. 21, no. 11, pp. 2069–2079, June 2013.
 - [10] P.-F. Chiu, M.-F. Chang, S.-S. Sheu, K.-F. Lin, P.-C. Chiang, C.-W. Wu, W.-P. Lin, C.-H. Lin, C.-C. Hsu, F.-T. Chen, K.-L. Su, M.-J. Kao, and M.-J. Tsai, “A low store energy, low VDDmin, nonvolatile 8T2R SRAM with 3D stacked RRAM devices for low power mobile applications,” in *Proc. IEEE Symp. VLSI Circuits (VLSIC)*, 2010, pp. 229–230.
 - [11] W. Wei, K. Namba, J. Han, and F. Lombardi, “Design of a nonvolatile 7T1R SRAM cell for instant-on operation,” *IEEE Transactions on Nanotechnology*, vol. 13, no. 5, pp. 905–916, Sept. 2014.
 - [12] S. Yamamoto, Y. Shuto, and S. Sugahara, “Nonvolatile SRAM (NVS RAM) using functional MOSFET merged with resistive switching devices,” in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, 2009, pp. 531–534.
 - [13] S.-S. Sheu, C.-C. Kuo, M.-F. Chang, P.-L. Tseng, C.-S. Lin, M.-C. Wang, C.-H. Lin, W.-P. Lin, T.-K. Chien, S.-H. Lee, S.-C. Liu, H.-Y. Lee, P.-S. Chen, Y.-S. Chen, C.-C. Hsu, F.-T. Chen, K.-L. Su, T.-K. Ku, M.-J. Tsai, and M.-J. Kao, “A ReRAM integrated 7T2R non-volatile SRAM for normally-off computing application,” in *Proc. IEEE Asian Solid-State Circuits Conference (A-SSCC)*, 2013, pp. 245–248.
 - [14] S. Hamdioui and A. J. van de Goor, “An experimental analysis of spot defects in SRAMs: realistic fault models and tests,” in *IEEE Asian Test Symp. (ATS)*, 2000, pp. 131–138.
 - [15] N. Z. Haron and S. Hamdioui, “On defect-oriented testing for hybrid CMOS/memristor memory,” in *IEEE Asian Test Symp. (ATS)*, 2011, pp. 353–358.
 - [16] —, “DfT schemes for resistive open defects in RRAMs,” in *Proc. Conf. Design, Automation, and Test in Europe (DATE)*, 2012, pp. 799–804.
 - [17] S. Kannan, J. Rajendran, R. Karri, and O. Sinaoglu, “Sneak-path testing of crossbar-based nonvolatile random access memories,” *IEEE Trans. on Nanotech.*, vol. 12, no. 3, pp. 413–426, May 2013.
 - [18] B.-C. Bai, K.-L. Luo, C.-A. Chen, Y.-W. Chen, H. W. M, C.-L. Hsu, L.-C. Cheng, and J. C.-M. Li, “Back-end-of-line-defect analysis for Rnv8T nonvolatile SRAM,” in *IEEE Asian Test Symp. (ATS)*, 2013, pp. 123–127.
 - [19] S. Kannan, R. Karri, and O. Sinaoglu, “Sneak path testing and fault modeling for multilevel memristor-based memories,” in *Proc. Int’l Conf. Comput. Des. (ICCD)*, 2013, pp. 215–220.
 - [20] L. O. Chua, “Memristor—the missing circuit

element,” *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sept. 1971.

- [21] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.
- [22] D. Biolek, Z. Biolek, and V. Biolkova, “SPICE modeling of memristive, memcapacitive and meminductive systems,” in *Prof. European Conference on Circuit Theory and Design (ECCTD)*, 2009, pp. 249–252.
- [23] D. Biolek, M. D. Ventra, and Y. V. Pershin, “Reliable SPICE simulations of memristors, memcapacitors and meminductors,” *Radioengineering*, vol. 22, no. 4, pp. 945–968, 2013.
- [24] H.-C. Shih, C.-Y. Chen, C.-W. Wu, C.-H. Lin, and S.-S. Sheu, “Training-based forming process for RRAM yield improvement,” in *Proc. IEEE VLSI Test Symp. (VTS)*, May 2011, pp. 146–151.

作者簡介

李昱霆



國立中央大學/電機工程學系/碩士生。研究專長為非揮發性靜態隨機存取式記憶體測試及診斷。

[E-mail:y865507@hotmail.com](mailto:y865507@hotmail.com)

陳詠孝



國立中央大學/電機工程學系/博士生。研究專長新興非揮發性記憶體測試與三維積體電路測試。

[E-mail: 995401003@cc.ncu.edu.tw](mailto:995401003@cc.ncu.edu.tw)

吳冠德



資訊與通訊研究所 / 設計自動化技術組 / 實習生。國立中央大學電機系博士生。研究專長為三維記憶體之測試與修復。

[E-mail:itri451596@itri.org.tw](mailto:itri451596@itri.org.tw)

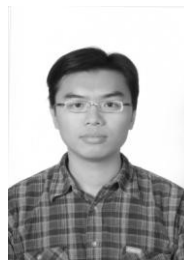
李進福



國立中央大學/電機工程學系/教授兼系主任。國立清華大學電機系博士。研究專長數位積體電路與記憶體設計與測試。

[E-mail:jfli@ee.ncu.edu.tw](mailto:jfli@ee.ncu.edu.tw)

駱致彥



資訊與通訊研究所 / 設計自動化技術組 / 工程師。國立清華大學電機系博士。研究專長為三維堆疊式系統晶片之可測試電路設計與記憶體測試與修復。

[E-mail:chihyenlo@itri.org.tw](mailto:chihyenlo@itri.org.tw)

蒯定明



資訊與通訊研究所 / 設計自動化技術組 / 技術組長。加州大學聖塔芭芭拉分校電機與計算機工程學系博士。研究專長為三維整合、平行處理與記憶體設計。

[E-mail: dmkwai@itri.org.tw](mailto:dmkwai@itri.org.tw)

周永發



資訊與通訊研究所 / 設計自動化技術組 / 技術組長。國立清華大學電機系博士。研究專長為記憶體與數位積體電路的設計與測試。

[E-mail: yfchou@itri.org.tw](mailto:yfchou@itri.org.tw)