

還在為選擇記憶體測試演算法而煩惱嗎？

隨著先進製程的應用越來越廣泛，物聯網(IoT, Internet of Things)晶片、高效能運算(HPC, High Performance Computing)晶片、人工智慧(AI, Artificial Intelligence)晶片、邊緣運算(Edge computing)晶片與影像處理(ISP, Image Signal Processing)晶片等，都需要使用更先進製程。在先進製程與上述各類應用需求的提升下，使用 SRAM (SRAM, Static Random Access Memory) 來儲存複雜的程式的情況越來越多。如何精準找出記憶體缺陷，降低 DPPM(DPPM, Defect Parts Per Million)相當重要。

過去要設計一個記憶體測試演算法，需要晶片設計工程師，根據晶片應用的特性、晶片測試後的診斷報告和製程，選擇適當的演算法。通常晶片設計工程師就會選擇最基本的 March C 或是 March C' 做為晶片量產測試的演算法。

近期，製程的演進快速，讓過去使用成熟製程的晶片開發商，必須選擇較為先進的製程，隨之而來的就是記憶體測試演算法的改變，因為過去使用的記憶體測試演算法，已經無法精準的檢測出在較為先進製程下的記憶體缺陷的型態 (Memory defect type)，造成 DPPM 提高。

然而，過去晶片設計工程師只能利用土法煉鋼的方式，在 RTL(Register-Transfer Level)階段，根據晶片測試後的診斷報告或是製程，選擇更適當的記憶體測試演算法，通常伴隨著就是晶片開發時間的拉長，因為晶片設計工程師無法有效率且精準的設計出更為適當的記憶體測試演算法。如何快速且有效率的完成記憶體測試演算法的變動呢？

芯測科技推出全世界第一套完全以圖形介面為主 (GUI-Based) 的使用者自定義演算法 (UDA, User Defined Algorithm) 微架構。芯測科技將記憶體測試演算法，以元件 (Component) 的形式表達，如表一，然後利用元件間的『重新排列組合』，形成新的記憶體測試演算法。

語法↵	功能↵
UP↵	Address 由 0 開始上數↵
DN↵	Address 由最大值開始下數↵
ADD_INC↵	由 UP 或 DN 決定，address + 1 或 address - 1↵
N↵	不做任何讀寫的行為↵
R(A)↵	Read memory data，括弧的部份為 Read 的 pattern A↵
W(A)↵	Write memory data，括弧的部份為 write 的 pattern A↵
S↵	Testing Sleeping↵
,↵	區隔不同的 operation↵
;↵	完成當下的 element↵

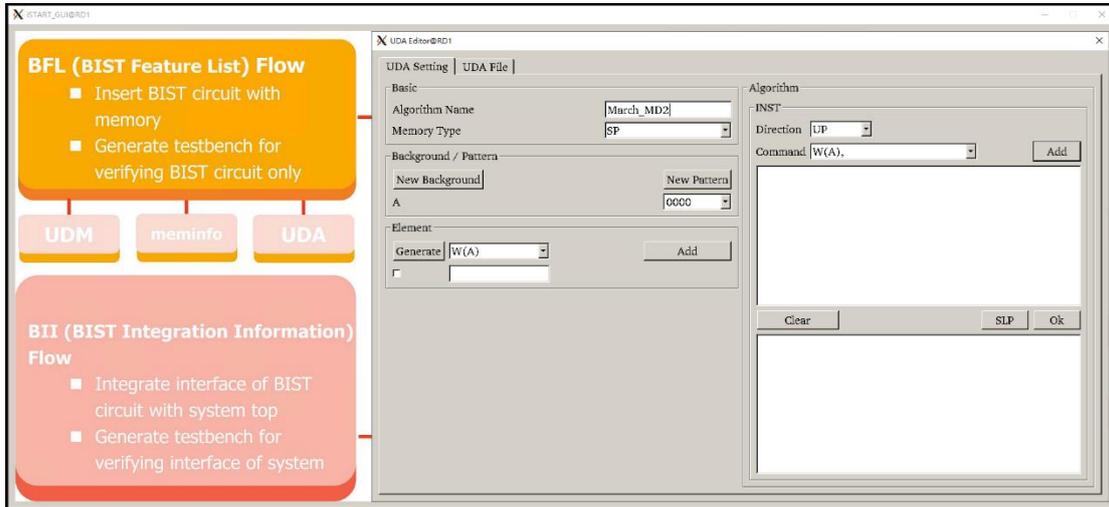
如此一來，任何記憶體測試演算法都可以利用 UDA 在幾分鐘內產生，大幅縮短晶片設計者開發記憶體測試演算法的時間，也可以讓晶片設計者根據記憶體缺陷的資訊，快速的產生『專用的』記憶體測試演算法。此外，透夠元件化的架構，UDA 產生的記憶體測試演算法的電路面積也相當精簡。以下為產生一個複雜度為 70N 的記憶體測試演算法 March MD2(如图一)的操作说明：

```

>(wa)
<(ra,wb,wb,rb,wb,wb,rb,wa,wa,ra,wa,wa,ra,wa,wb,wa,wb)
>(rb,wa,wa,ra,wa,wa,ra,wb,wb,rb,wb,wb,rb,wb,wa,wb,wa)
<(ra,wb,rb,wb,rb,rb,rb,wa,ra,wa,ra,ra,ra,wa,wb,wa,wb)
>(rb,wa,ra,wa,ra,ra,ra,wb,rb,wb,rb,rb,rb,wb,wa,wb,wa)
>(ra)

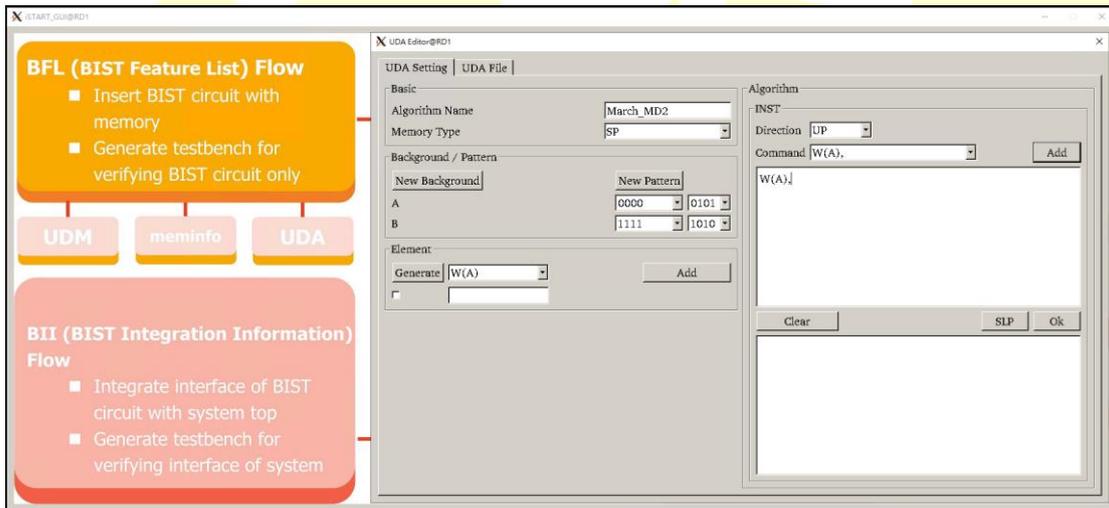
```

一、透過 GUI 選擇 UDA 的使用，如圖二。

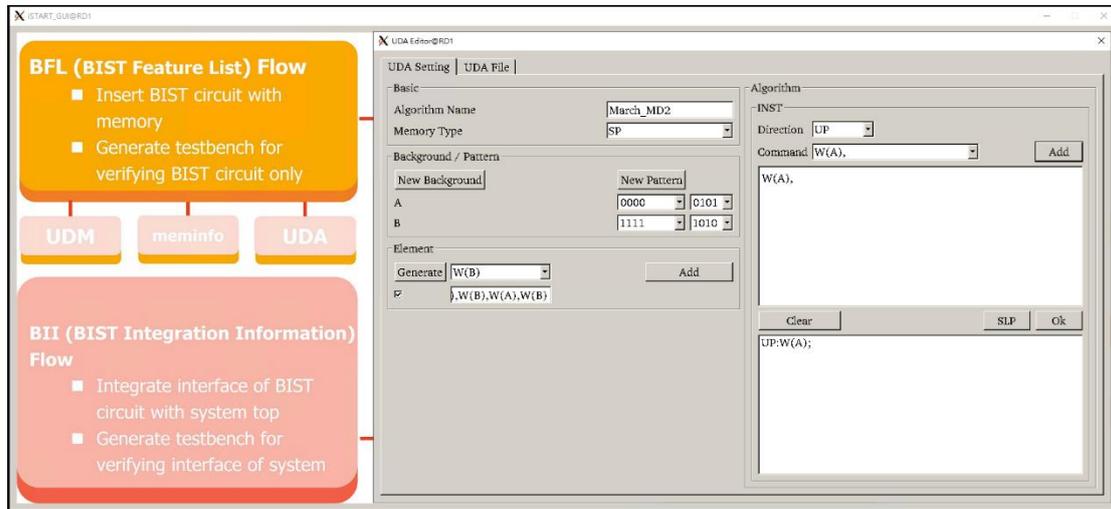


圖二

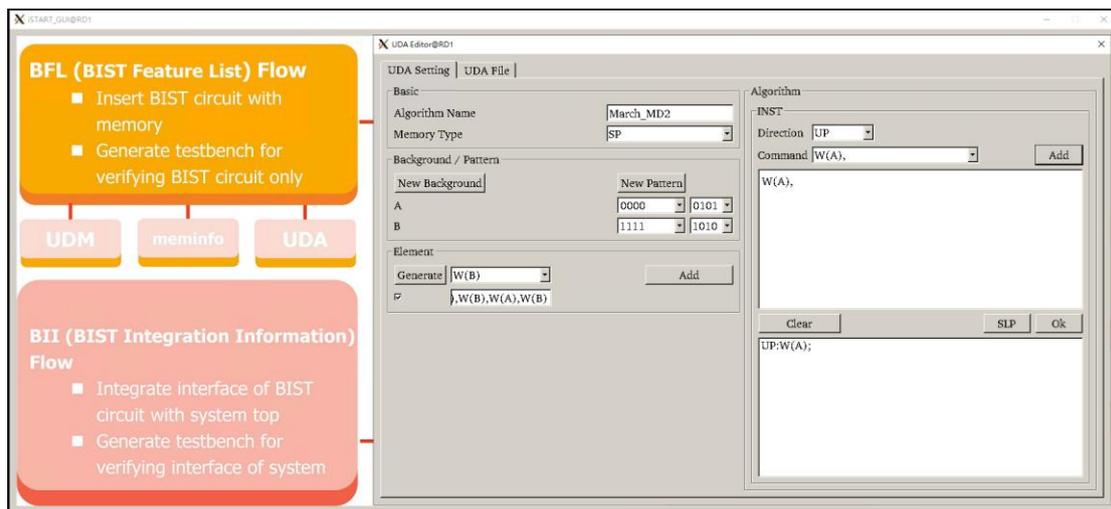
二、透過 GUI 根據 March MD2 的演算法依序完成記憶體測試演算法的實現，如圖三、圖四、圖五、圖六、圖七、圖八、圖九、圖十、圖十一、圖十二、圖十三、圖十四、圖十五、圖十六。



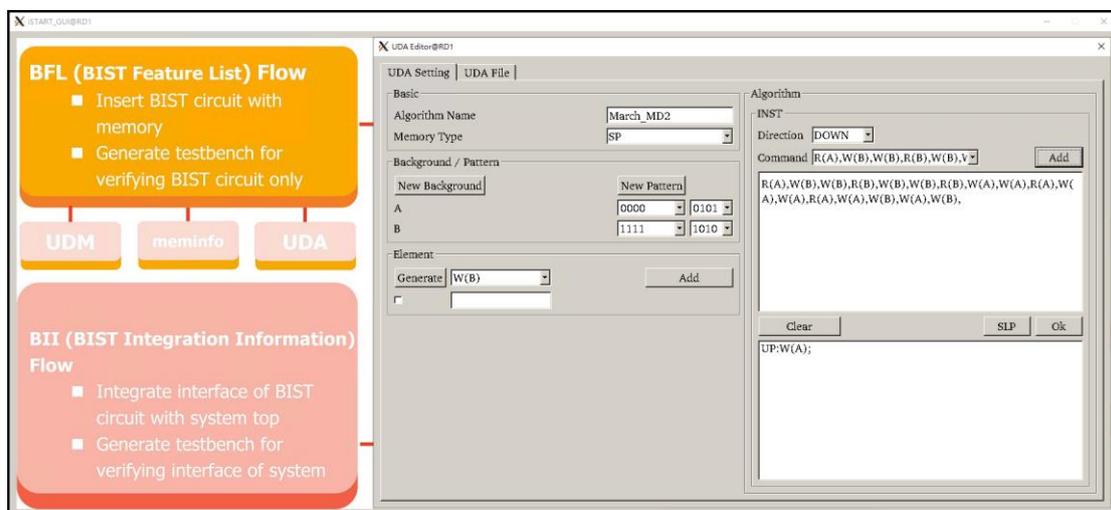
圖三



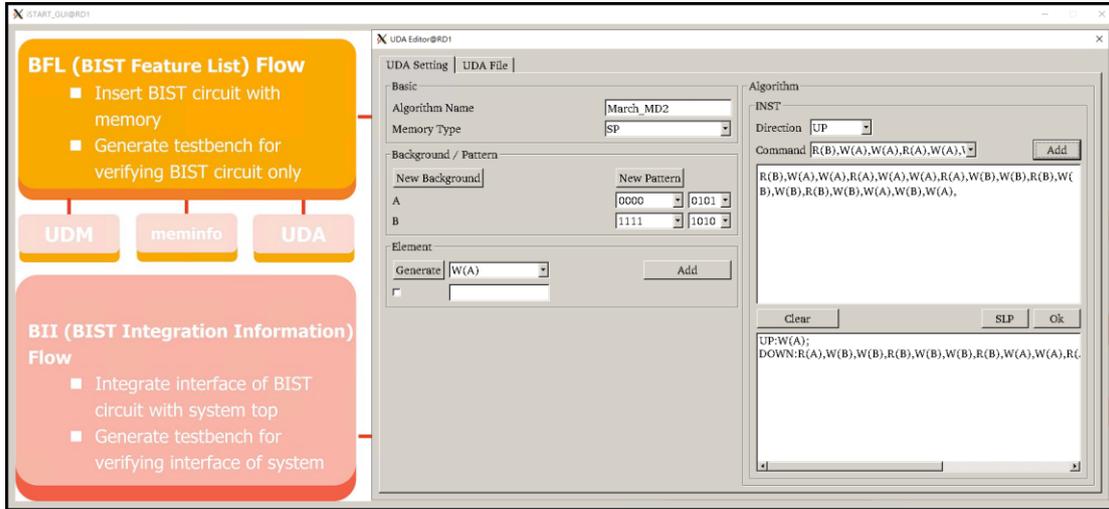
圖四



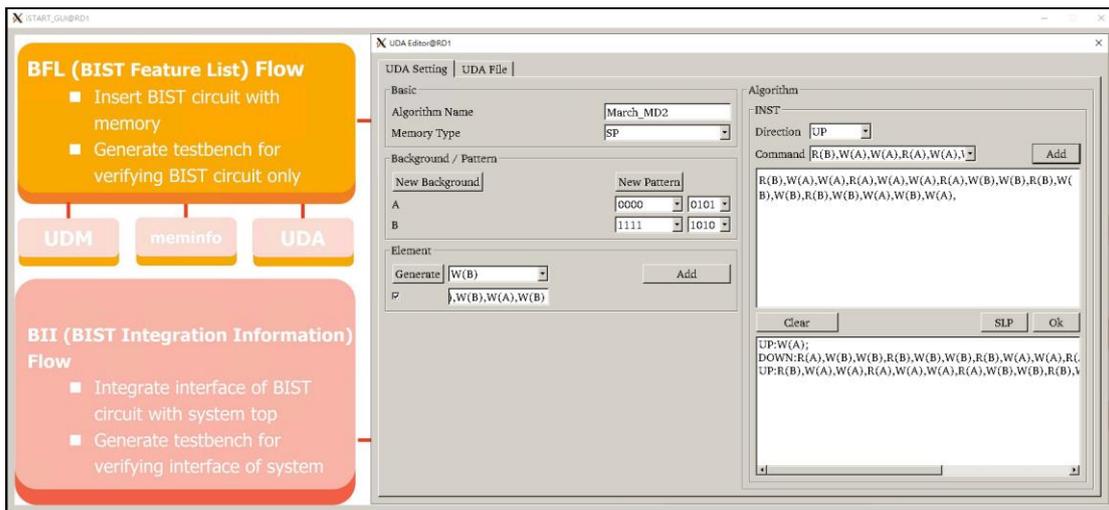
圖五



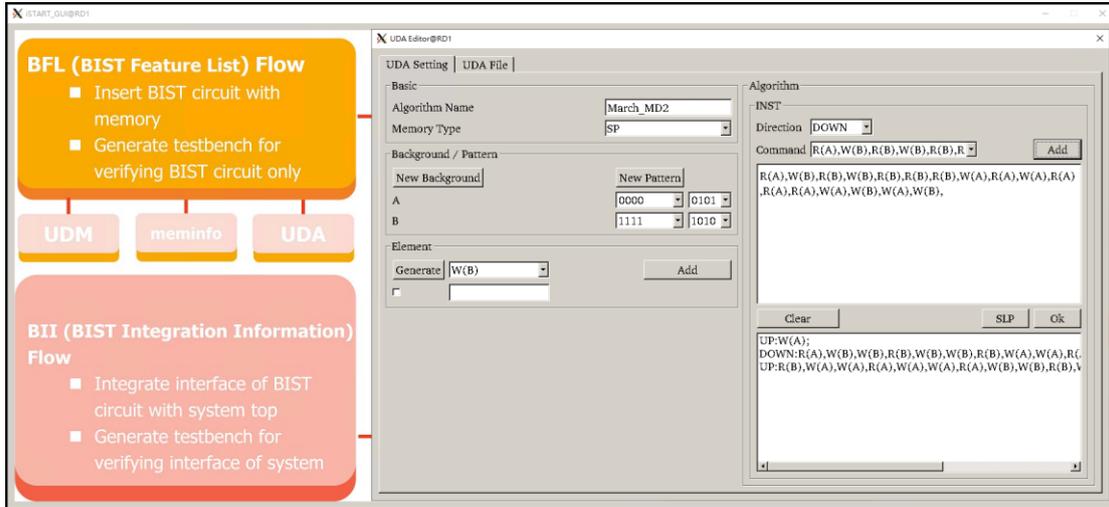
圖六



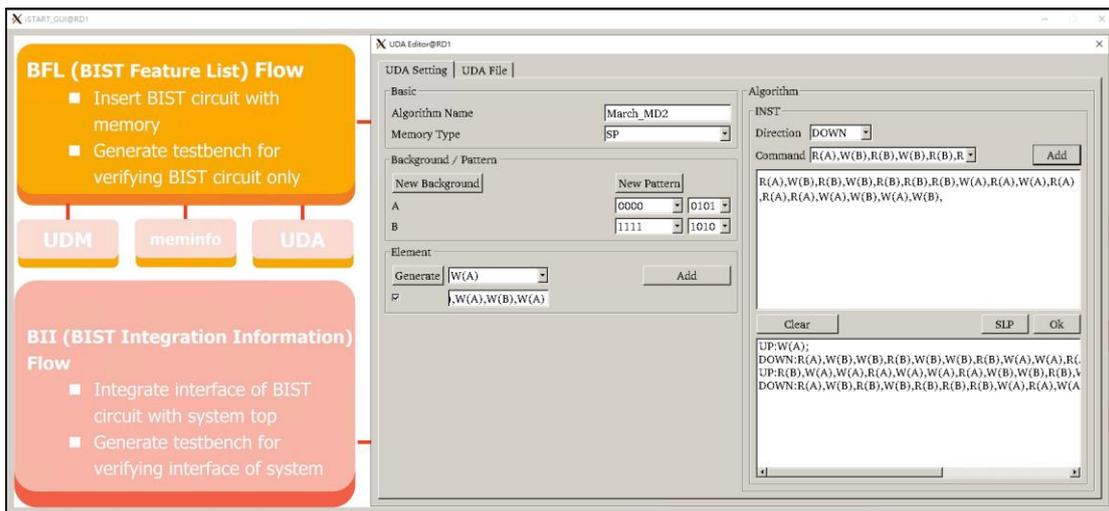
圖九



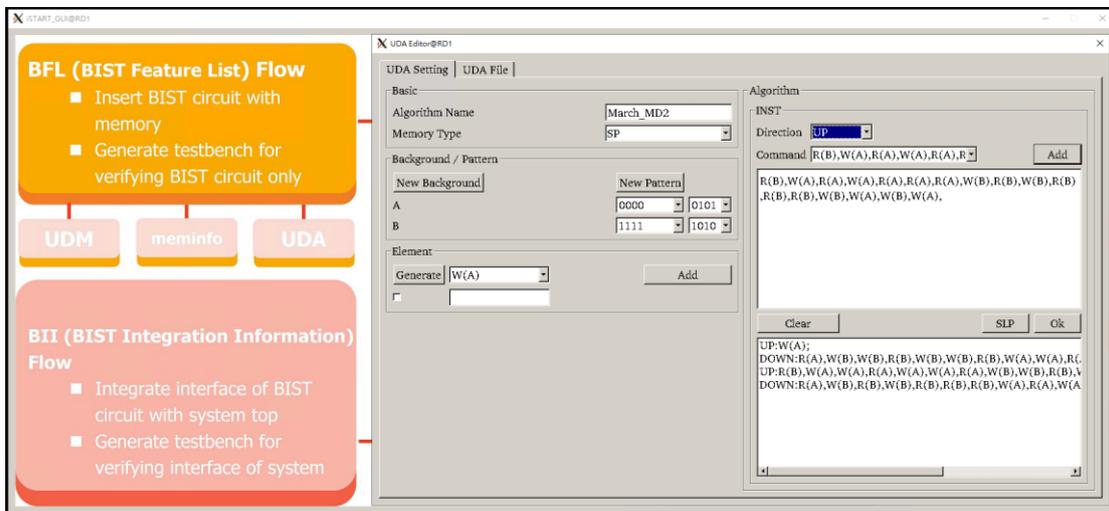
圖十



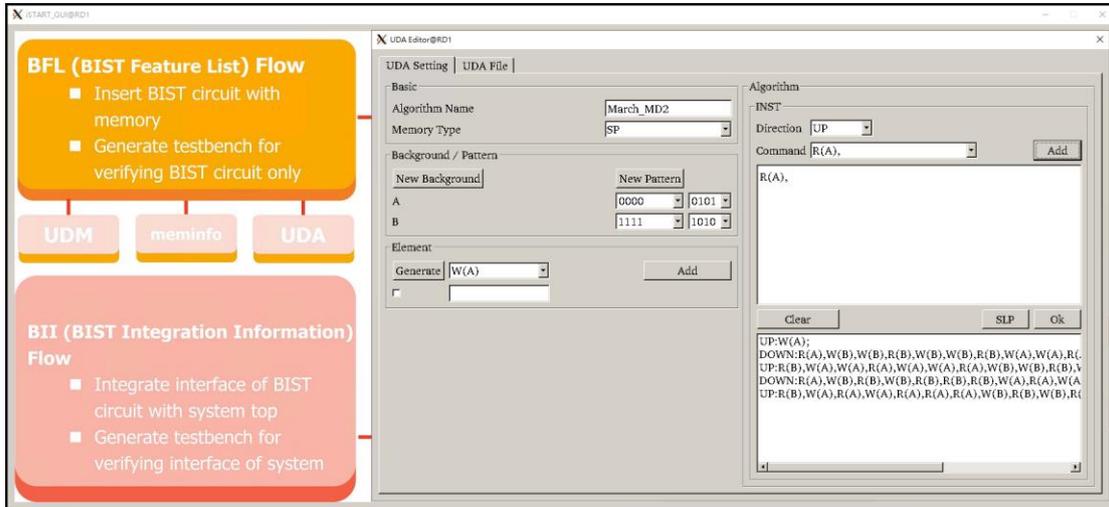
圖十一



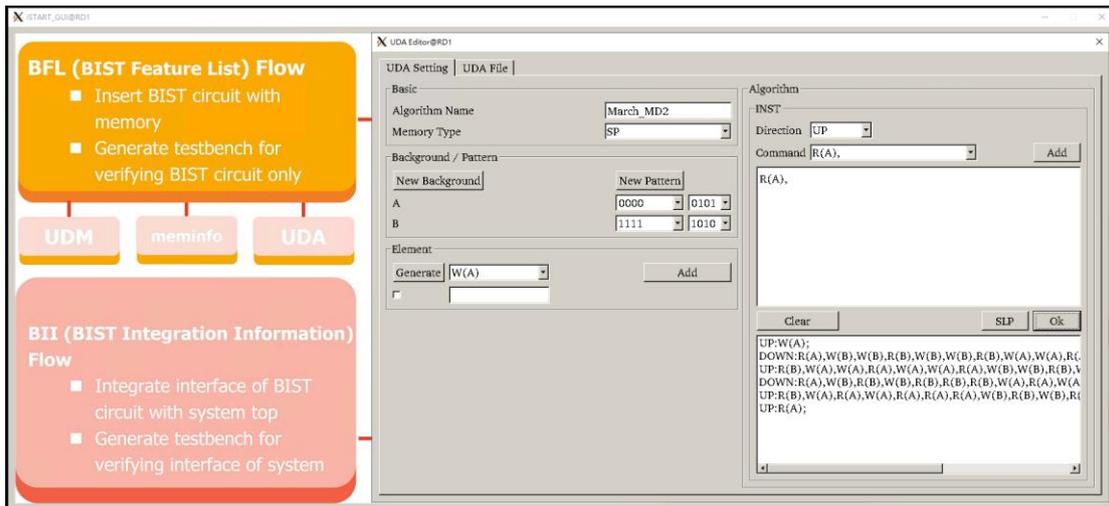
圖十二



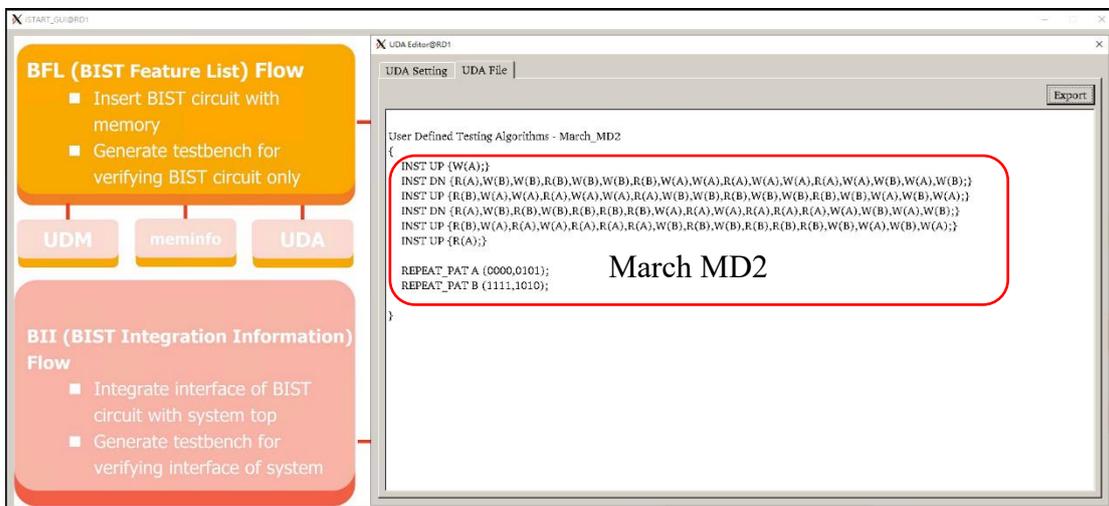
圖十三



圖十四



圖十五



圖十六

透過上述幾個步驟，晶片設計工程師，可以利用 UDA 微架構設計出複雜度很高的記憶體測試演算法。透過 UDA 的使用，讓記憶體測試演算法的設計更具彈性化也更多元。

先進製程的演進帶動記憶體需求的增加，相對的各種記憶體缺陷的型態也會因應而生，開發出複雜度夠高的記憶體測試演算法，才可精準的檢測出各類的記憶體失效情況。

芯測科技的 UDA 微架構，可望為晶片設計者帶來快速、省成本且高效能的服務。

iSTART