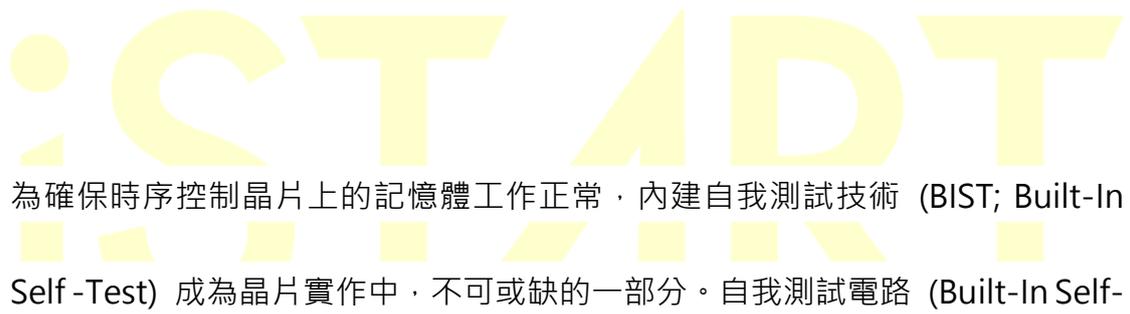


提高整體開發流程的記憶體測試電路開發工具

隨著半導體製程技術的提升，IC 設計規模與時脈愈來愈高，加上目前 IC 設計對於記憶體 (RAM、ROM、Embedded Flash、DRAM、Embedded DRAM) 需求的比重愈來愈大，對於 5G、TV、網通、物聯網、挖礦機、車用電子、智慧語音裝置等新一代系統晶片 (SoC) 內的記憶體需求量也大，就意味著更大的晶片面積，且隨著效能與耗電的要求更加嚴謹，晶片的製程也就愈往高階製程邁進，設計的複雜度越高，開發的時間相對也越高。



為確保時序控制晶片上的記憶體工作正常，內建自我測試技術 (BIST; Built-In Self-Test) 成為晶片實作中，不可或缺的一部分。自我測試電路 (Built-In Self-Test)，可以提高測試的錯誤涵蓋率，縮短設計週期，增加產品可靠度，並加快產品的上市速度。為了迅速處理大量的資訊，記憶體所占相對於晶片面積的比重也將日益增加，除了需要有機制能夠找出記憶體的錯誤，更需要進一步能夠進行記憶體的修復，也需要提高整體開發流程的穩定性與便利性，因此，芯測科技 (iSTART-Tek Inc.) 推出「Bottom-up Flow」，提供使用者系統性的流程，使用者可依其需求選擇應用，提高生產率並加速產品開發的時程。

iSTART 的 Bottom-up 流程專為 IP / Harden 所設計，實現此流程包括兩個主要步驟。首先，使用者需要在 IP 或模組中插入 BIST 電路；最終，將所有的模組整合至 SoC 中，以減少開發的時程。以下將介紹完整的 Bottom-up 流程。

1 Bottom-up Flow 舉例說明

1.1 Test Case 架構圖

圖 1-1 是 Test Case 架構圖，圖中的三個 module design1, design2 與 design4 需要被模組化(hardened)，此 case 需要執行兩次 bottom-up Flow。

另外 design3 的 BIST(Built-In Self Test)電路會在執行 design1 的 BFL flow 中插入。

iSTART

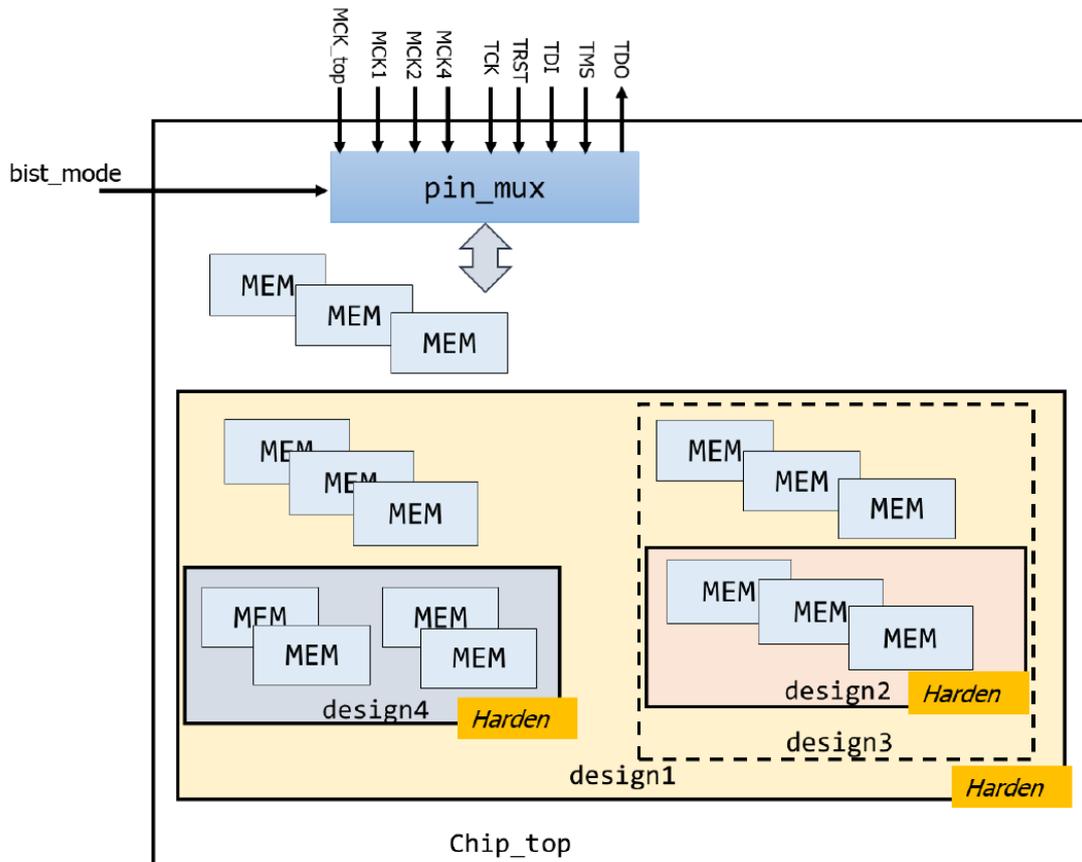


圖 1-1 Test Case for Bottom-up Flow

1.2 Test Case 實例

圖 1-2 是此 Test Case 的實作流程，使用 START 執行 Bottom-up Flow，使用者需要將插好的 BIST 電路整合到最頂層的模組，以此例子，使用者首先將 BIST 電路插入 design2 和 design4，第二步將 BIST 電路插入 design1，這一步同時也將 BIST 電路插入到 design3，再來將 BIST 電路插在頂層的模組 chip_top，最後，整合每一個 BIST 的串聯並使用標準 JTAG 介面進行控制。單一 JTAG 介面的控制方式，可節省晶片頂層的腳位數目，且標準 JTAG 介面，也方便與其它功能整合。

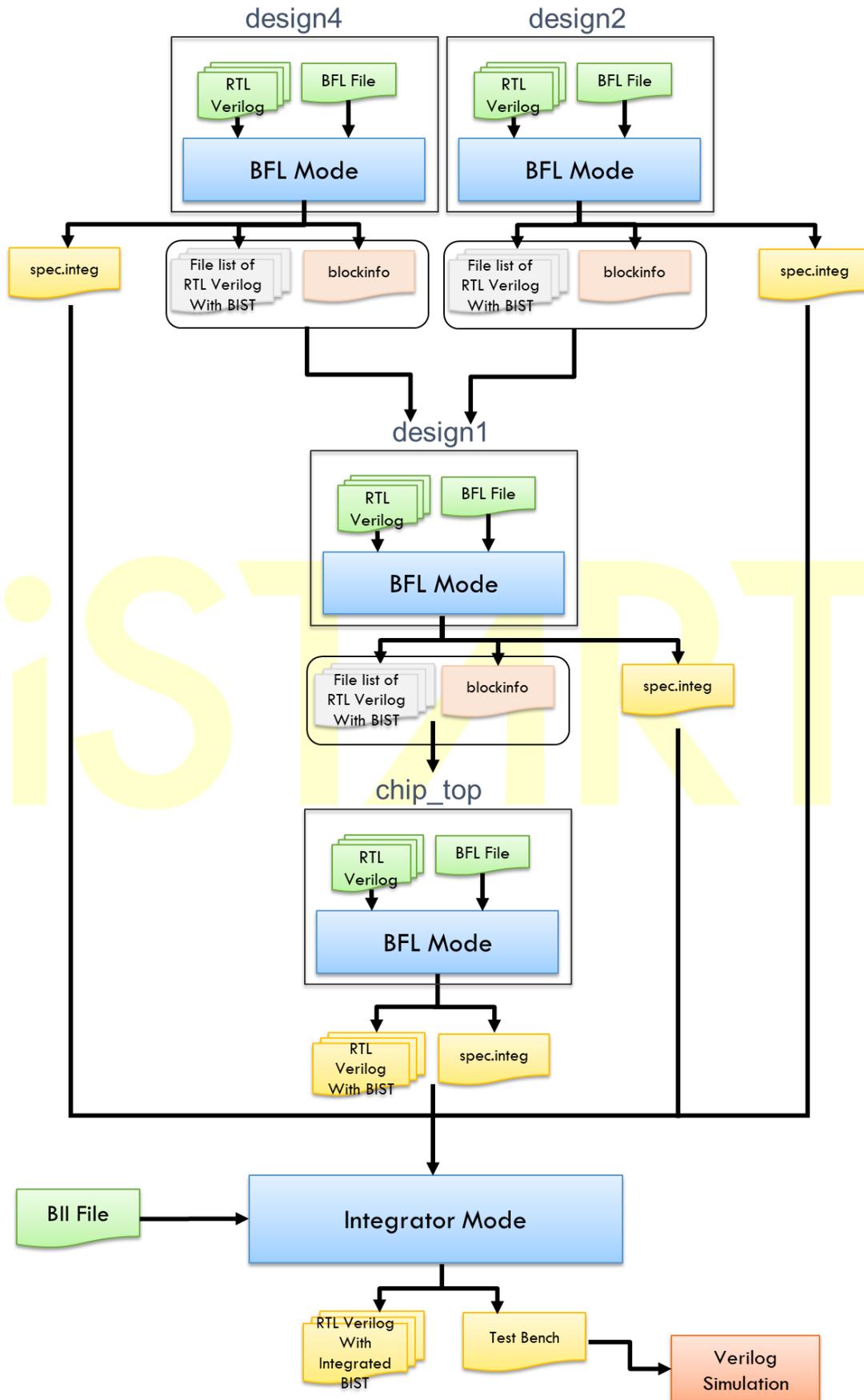


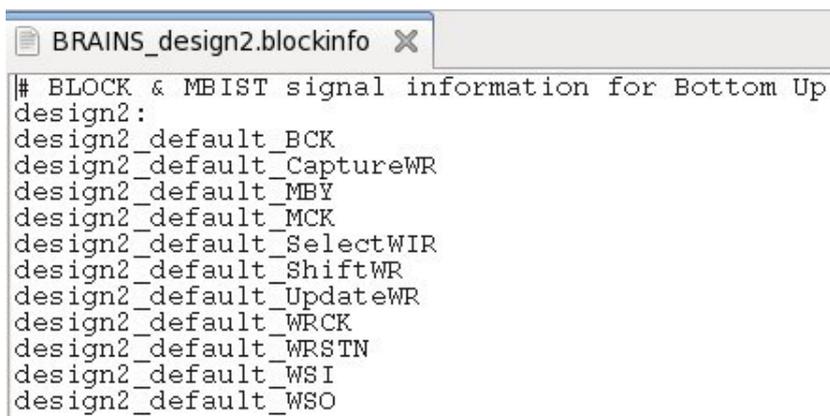
圖 1-2 Test Case 執行流程

1.2.1 在 design2 和 design4 插入 BIST 電路

使用我們 BLF 設定檔範本，可以從中填寫每一個子模組的設定。在此 Test Case 中，需要注意某些設定，由於此 case 將與 JTAG 介面整合，因此子模塊的控制接口應設置為 IEEE 1500 介面。此外，在這種情況下，子模塊將會模組化，因此為了整合 BIST 電路，使用者需要在頂層產生接口。所以，修改每個子模塊的 BFL 設定檔應遵循以下兩個設定：

- set bist_interface = ieee1500
- set integrator_mode = no

如果 integrator_mode 設定 no, START 會產出*.blockinfo 文件，在這個 case, 我們 會 得 BRAINS_design2.blockinfo 和 BRAINS_design4.blockinfo, 參考 圖 1-3 BRAINS_design2.blockinfo, 這兩份檔案將會使用在步驟.2.2.



```
# BLOCK & MBIST signal information for Bottom Up
design2:
design2_default_BCK
design2_default_CaptureWR
design2_default_MBY
design2_default_MCK
design2_default_SelectWIR
design2_default_ShiftWR
design2_default_UpdateWR
design2_default_WRCK
design2_default_WRSTN
design2_default_WSI
design2_default_WSO
```

圖 1-3 BRAINS_design2.blockinfo for Bottom Up Flow

當 BLF 檔案設定完成後，即可以輸入指令去完成插入 BIST 電路流程。

1.2.2 在 design1 插入 BIST 電路

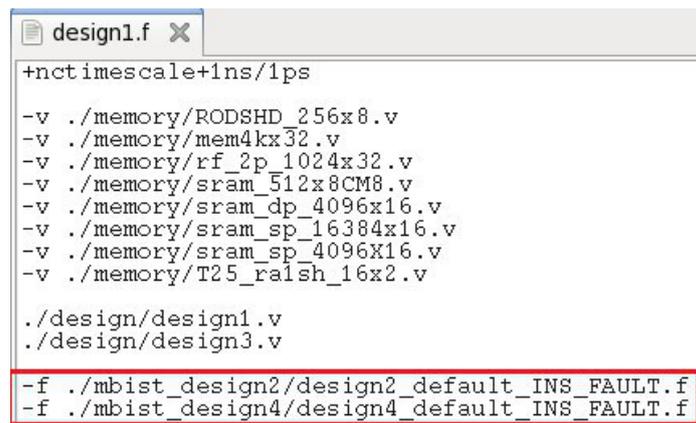
基本上操作與 design2/design4 流程一樣，只是要在 OPTION function 多設定 block_path 選項，和手動修改 file list, design1 的 BLF 檔案設定如下：

- set bist_interface = ieee1500
- set integrator_mode = no
- set block_path = ./mbist_design2/BRAINS_design2.blockinfo
| ./mbist_design4/BRAINS_design4.blockinfo

如果使用者也想將 design3 模組化，請遵循步驟 1.2.1 插入 BIST 電路到 design3，design2 成為 design3 的子模組，接著設定 block_path 如下：

- set block_path = ./mbist_design3/ BRAINS_design3.blockinfo
| ./mbist_design4/BRAINS_design4.blockinfo

接著，請將 design2 和 design4 插入 BIST 的 file-list 檔案加入 design1.f, 如圖 1-4，接著輸入指令透過 BFL 檔案去完成插入 BIST 電路流程。



```
design1.f x
+nctimescale+1ns/1ps
-v ./memory/RODSHD_256x8.v
-v ./memory/mem4kx32.v
-v ./memory/rf_2p_1024x32.v
-v ./memory/sram_512x8CM8.v
-v ./memory/sram_dp_4096x16.v
-v ./memory/sram_sp_16384x16.v
-v ./memory/sram_sp_4096x16.v
-v ./memory/T25_ralsh_16x2.v

./design/design1.v
./design/design3.v
-f ./mbist_design2/design2_default_INS_FAULT.f
-f ./mbist_design4/design4_default_INS_FAULT.f
```

圖 1-4 File List of design1 in Test Case

1.2.3 在 Top Module (chip_top)插入 BIST 電路

Top Module 插入 BIST 電路做法跟 design1 幾乎一樣，除了 integrator_mode 選項需要改成 yes，其餘選項可以依照前面步驟設定，

注意以下選項需設定如下：

- set bist_interface = ieee1500
- set integrator_mode = yes
- set block_path = ./mbist_design1/ BRAINS_design1.blockinfo

請將 design1 插入 BIST 電路的 file-list 檔案放入 chip_top.f，如圖 1-5，

接著輸入指令透過 BFL 檔案去完成插入 BIST 電路流程。



```
chip_top.f X
+nctimescale+1ns/1ps
-v ./memory/RODSHD_256x8.v
-v ./memory/mem4kx32.v
-v ./memory/rf_2p_1024x32.v
-v ./memory/sram_512x8CM8.v
-v ./memory/sram_dp_4096x16.v
-v ./memory/sram_sp_16384x16.v
-v ./memory/sram_sp_4096x16.v
-v ./memory/T25_ralsh_16x2.v
./design/chip_top.v
-f ./mbist_design1/design1_default INS_FAULT.f
```

圖 1-5 File List of design1 in Test Case

1.2.4 在 top level 整合四個 BIST 電路

使用者可以透過 START 的 integration 流程將各個 BIST 進行串聯，執

行 integrator 前需產生 BIST Integration Information (BII) 檔案，透過

設定該檔案來提供 IEEE1149.1 規格、BIST 規格記錄檔(*_spec.integ)等

資訊後，即可讀取該 BII 檔案來實現 BIST 電路的串聯與 wrapper 的套用。

這邊需額外注意如果使用者將模組化的 module 例化(instantiate)多次，則必須在 BII 檔案中設定*.postfixinfo 檔案如下，START 會在插入 BIST 電路後產出*.postfixinfo 檔案

- set postfixinfo = ./mbist_top/BRAINS_top.postfixinfo

1.3 執行完後的架構圖

圖 1-6 為 Test Case 執行完後的架構，依 BII 檔案內容設定來完成 BIST 的串接動作與 wrapper 的套用，該整合模組用於整合 BIST 的 WSI 與 WSO 訊號；以及 BISR 的 WSI、WSO、MRSI 與 MRSO 訊號，幫助提供使用者將 IEEE1149.1 的輸出入埠及 BIST clock 映射至系統電路內部指定的 wire 上，所以亦包含了一個 IEEE1149.1 TAP 的引用模組來實現 IEEE1149.1 與 IEEE1500 之間的訊號轉換。

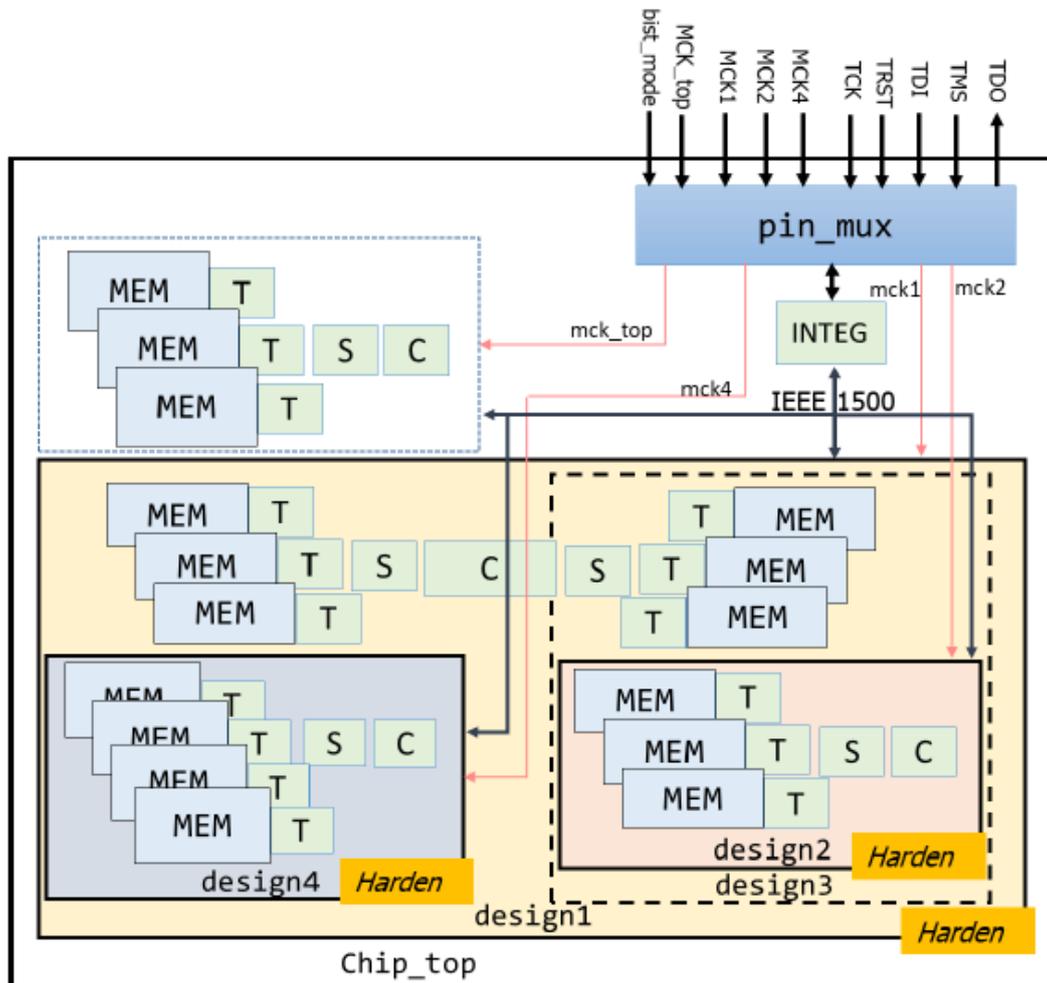


圖 1-6 Architecture after Implementation

因應人工智慧(AI)、車用電子(Auto)世代來臨，時序控制晶片內含之記憶體數量勢必愈來愈多，此時，記憶體測試解決方案亦成為晶片設計中不可或缺的一環。藉由芯測科技的記憶體測試與修復的整合性開發環境-START 的新功能「Bottom-up Flow」，對使用者來講，不需太過繁複的設定過程，即可完成記憶體測試解決方案的實作，以此幫助使用者提高生產率並有效降低產品開發的時程，提升產品可靠度並有效的延長產品的使用壽命。

關於芯測科技

隨著半導體先進製程演進的快速腳步，加上現今各種電子產品功能日趨複雜，系統晶片設計不僅變得更加困難，同時對於記憶體的需求更是日益增加。因而在追求如何提升產品效能降低功耗等課題外，如何加入適當的設計驗證電路來維持晶片的品質，提升可靠度以及控制成本，更是決勝的關鍵點。芯測科技透過創新的可程式化暨管線式架構記憶體測試技術與特有架構的記憶體修復技術，加上客制化以及即時的技術支援服務，提供客戶完整的記憶體測試與修復解決方案，來滿足不同製程及應用的需求。

- 檢測與修復結合的 SoC Memory 測試與修復解決方案：[START™](#)
- 便捷版記憶體測試開發平臺：[EZ-BIST](#)
- 非揮發性記憶體測試與修復矽智財：[NVM Test and Repair IP](#)
- 各類記憶體客制化測試與修復解決方案

展望競爭激烈的電子系統產品市場，芯測科技憑藉其完整的設計驗證解決方案，能夠協助你的團隊有效的完成相關設計，以控制產品品質與良率，在市場上取得領先地位。

以上技術文章，懇請惠予刊登;如有任何疑問，敬請不吝請教。

本篇作者：

[李韋宏](#)

高級工程師

新聞聯絡人：

[蔡羽涵](#)

t: +886-3-560-1667 #316

e: y.h.tsai@istart-tek.com