iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

# 在 DFT 設計流程中，利用 START 縮短 AI 晶片量產時程

隨著科技的進步，人工智慧(AI)蓬勃發展，許多的應用在生活中隨處可見，例如: 人臉辨識、車牌辨識、APPLE 的 Siri 等，因應目前疫情，人工智慧也能為防疫出一份力，工研院最近發表了一項技術，利用 AI 的技術與熱顯像儀搭配，透過人臉辨識來避免其他的熱源(熱咖啡…等)造成熱顯像儀判斷錯誤，加速量測體溫的時間與正確性，因此 AI 晶片的需求也急速上升，如果能夠縮短 DFT 的時間，便能加速晶片的生產，先他人一步取得市場先機。

為了縮短 DFT 的設計時間，先從 AI 的組成開始，AI 由硬體、演算法、資料庫所組成，其中資料庫需要使用到大量的記憶體，以人臉辨識為例子，要正確的辨識出人臉需要透過大量的資料來比對，因此存放這些資料的記憶體相對的重要，但是一塊 AI 晶片內，會有成千上萬顆的記憶體，若是人工一顆一顆的設計會浪費大量的時間，因此 iSTART 的產品 START 提供了友善的環境，讓 DFT 的工程師能夠透過 START 輕鬆的設計 BIST (Built-In Self -Test)及 BISR (Built-In Self -Repair)，來縮短設計時間，加速晶片生產。

AI 晶片屬於大型 SoC 設計，DFT 實作上需要耗費較多的時間，透過工具輔助，

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

可以有效地縮短設計時間，介紹三個好用的功能，Bottom-Up、Auto-Clocking gating cell insertion、Multi-Chain，讓 DFT 工程師可以迅速地完成設計。

## Bottom-Up flow

SoC 內通常會有許多相同性質的模組重複呼叫，Bottom-up 功能便能有效地縮短設計 BIST (Built-In Self -Test)及 BISR (Built-In Self -Repair)的時間，由於是將模組封成 block 的設計，在每個 block 下都可以進行記憶體檢測，確保這個 block 的正確性，在重複呼叫 block 時可以縮短驗證模組的時間。

### 1. 測試範例架構圖

圖 1 是測試範例架構圖，圖中的三個模組 design1、 design2 與 design4 需要被模組化，此範例需要執行兩次 Bottom-up 流程，另外 design3 的 BIST(Built-In Self Test)電路會在執行 design1 的 BFL (BIST FEATURE LIST) 流程中插入。

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

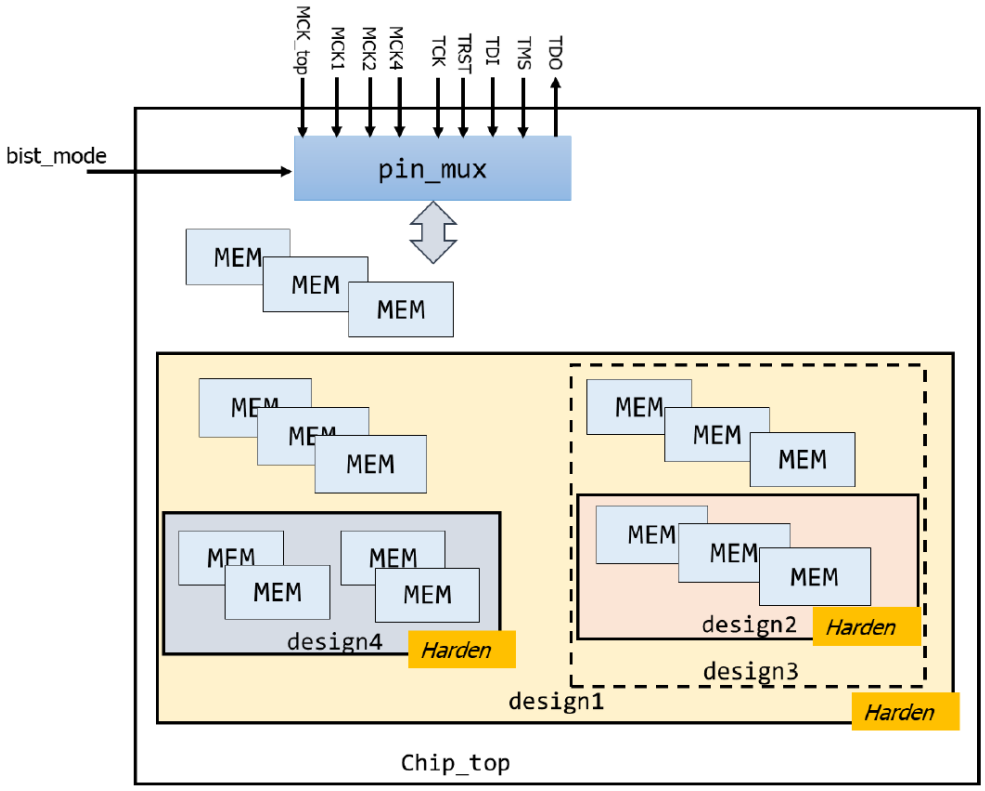


圖 1. Bottom-up Flow 測試範例架構圖

## 2. 測試範例實作

圖 2 是此測試範例的實作流程，使用 START 執行 Bottom-up 流程，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 desing2 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。

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

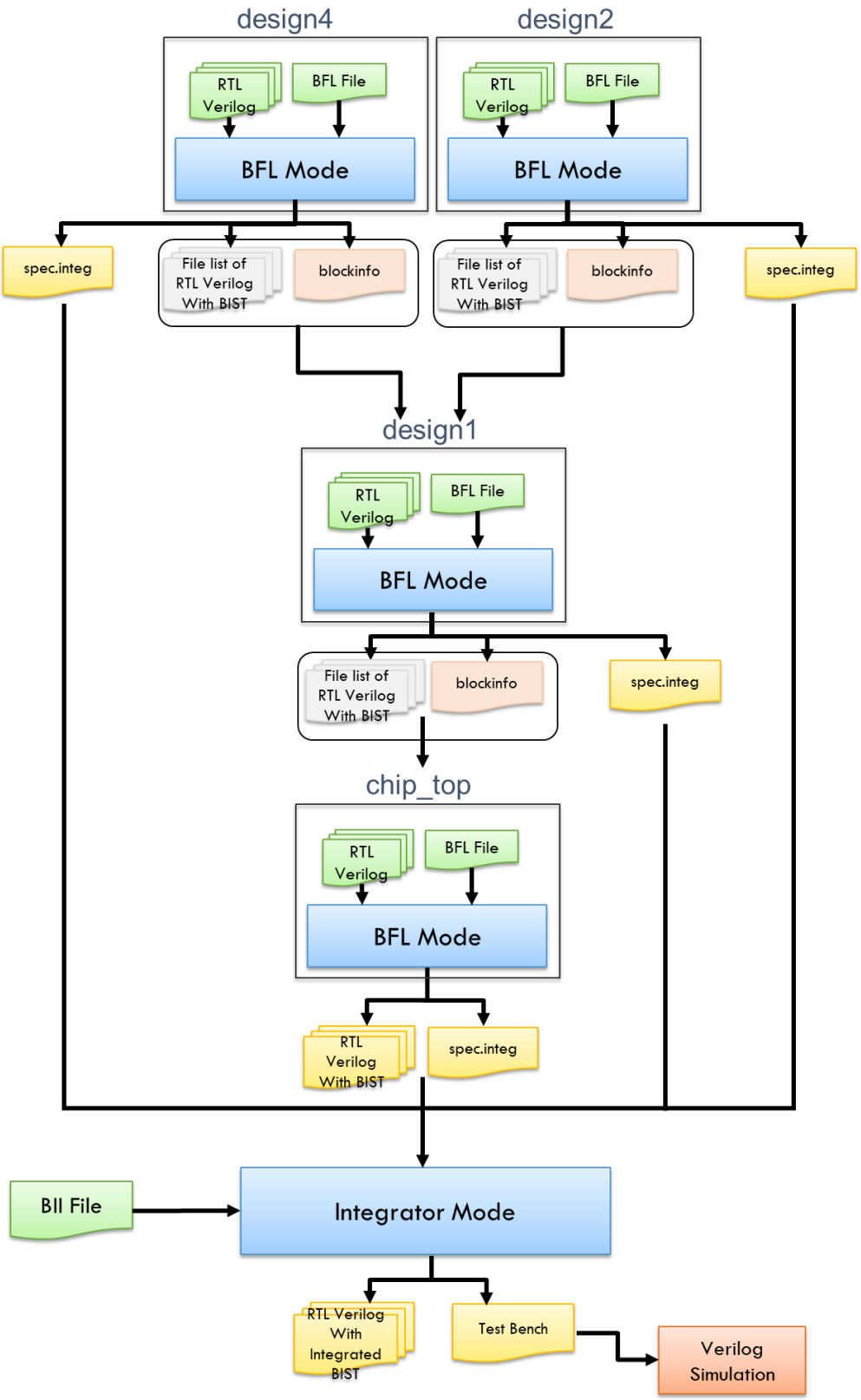


圖 2. 測試範例執行流程

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

### 2.1 在 design2 和 design4 插入 BIST 電路

使用我們 BFL (BIST FEATURE LIST)設定檔範本，可以從中填寫每一個子模組的設定。在此測試範例中，需要注意某些設定，由於此範例將與 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 (BIST FEATURE LIST)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，在這個範例，我們會得到 START_design2.blockinfo 和 START_design4.blockinfo，參考 圖 3 的 START_design2.blockinfo，這兩份檔案將會使用在步驟 2-2。

START_design2.blockinfo

```
# 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 3. Bottom Up 流程中的 START_design2.blockinfo

當 BFL (BIST FEATURE LIST)檔案設定完成後，即可以輸入指令去完成插入 BIST 電路流程。

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

### 2.2 在 design1 插入 BIST 電路

基本上操作與 design2、design 4 流程一樣，只是要在 OPTION function 多設定 block_path 選項，和手動修改檔案清單，design1 的 BFL (BIST FEATURE LIST)檔案設定如下：

- set bist_interface = ieee1500
- set integrator_mode = no
- set block_path = ./mbist_design2/START_design2.blockinfo | ./mbist_design4/ START_ design4.blockinfo

如果使用者也想將 design3 模組化，請遵循步驟 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 的檔案清單加入 design1.f，接著輸入指令透過 BFL (BIST FEATURE LIST)檔案去完成插入 BIST 電路流程。

### 2.3 在 Top 模組 (chip_top)插入 BIST 電路

Top 模組插入 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/START_design1.blockinfo

請將 design1 插入 BIST 電路的 file-list 檔案放入 chip_top.f，接著輸入指令透過 BFL (BIST FEATURE LIST)檔案去完成插入 BIST 電路流程。

### 2.4 在 top 階層整合四個 BIST 電路

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

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 BII (BIST INTEGRATION INFORMATION)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 (BIST INTEGRATION INFORMATION)檔案來實現 BIST 電路的串聯與 wrapper 的套用。

這邊需額外注意，在 Bottom-up 流程中，START 會在插入 BIST 電路後產出 *.postfixinfo 檔案，必須在 BII (BIST INTEGRATION INFORMATION)檔案中設定*.postfixinfo 檔案如下：

- set postfixinfo = ./mbist_chip_top/START_chip_top.postfixinfo

## 2.5 執行完後的架構圖

圖 4 為測試範例執行完後的架構，依 BII (BIST INTEGRATION INFORMATION)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。

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

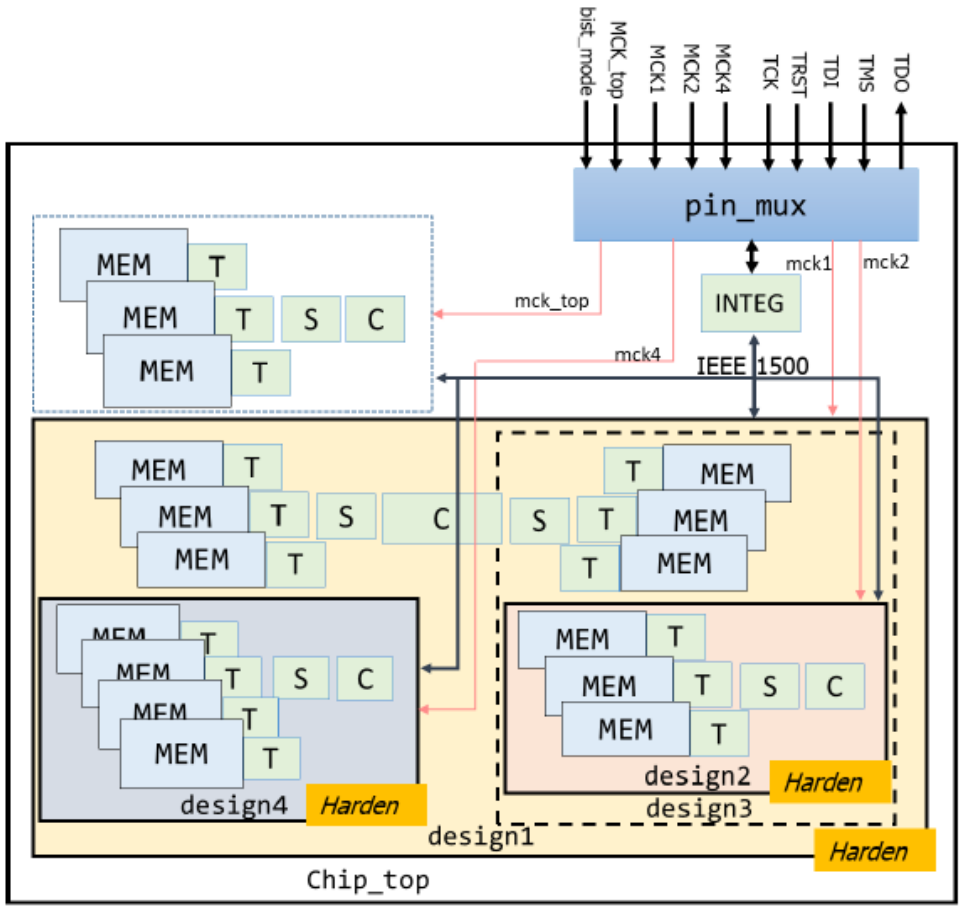


圖 4. 執行後架構圖

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

# Auto-Clocking gating cell insertion

由於 DFT 設計時，記憶體測試不需要無時無刻都有 clock，為了省電會在 BIST 及 BISR 的 clock 點加入 gating cell 以及 mux，以人工的方式去搜尋 clock，會耗費大量的時間，因此使用 START 來自動串接，填寫相關的檔案即可自動串接，省下人工的搜尋時間。

## 1. 測試範例架構圖

圖 5 為測試範例架構圖，圖中的 MCK 為使用者設計的 clock 要接至 BIST 電路，在 BII 流程中，可以在 clock 接至 BIST 電路前依照需求插入 gating cell。

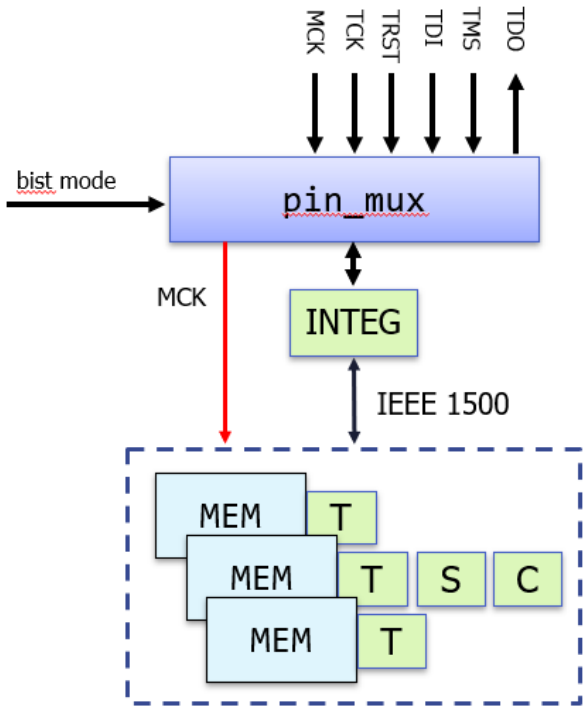


圖 5. Auto-Clocking gating cell insertion 測試範例架構圖

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

## 2. 測試範例實作

使用 START 執行 Auto-Clocking gating cell insertion 時，使用者需撰寫 gate_cell.info 將指定的 library 填入，若需要在前一級產生 MUX，則須多撰寫 mux_cell.info，並將這兩個 file 的路徑填入 BII file。

### 2.1 gate_cell.info 及 mux_cell.info 撰寫

圖 6 為實際 gate_cell.info 的截圖，其中格式為固定的，port 的順序也是固定，需依照此格式依序填入。

**gate_cell.info 格式：[CellName] xx [IO Description] x, y, s, z**

- xx -> 填入所要使用的 cell module name
- x -> 填入 clock input 的 port name
- y -> 填入 enable input 的 port name
- s -> 填入 clock enable 的 port name
- z -> 填入 clock output 的 port name

```
#Format: [CellName] xx [IO Description] x, y, s, z port name x(cl
[CellName] CKLHQD12BWP35P140ULVT [IO Description] CPN, E, TE, Q
```

圖 6. gate_cell.info 檔案截圖

圖 7 為實際 mux_cell.info 的截圖，其中格式為固定的，port 的順序也是固定，需依照此格式依序填入。

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

**mux_cell.info 格式：[CellName] xxx [IO Description] x, y, s, z**

- xxx -> 填入所要使用的 mux module name
- x -> 填入 clock1 input 的 port name
- y -> 填入 clock2 input 的 port name
- s -> 填入 clock select 的 port name
- z -> 填入 clock output 的 port name

```
#Format: [CellName] xxx [IO Description] x, y, s, z port name
[CellName] GMUX2D2BWP30P140ULVT [IO Description] I0, I1, S, Z
```

圖 7. mux_cell.info 檔案截圖

### 2.2 BII file 設定

撰寫完上述兩個檔案後，將兩個檔案的路徑填入 BII file，如圖 8 所示，gate_cell.info 填入 gate_cell_lib，mux_cell.info 填入 gate_cell_with_mux，完成設定，最後在執行 START 的 BII，tool 就會自動將所填的資訊插入電路中。

```
define{Integrator}[INTEG]
    set port_alias              = no            # yes, no
    set create_pio              = yes           # yes, no
    set integrator_interface    = mux-out       # all-out / mux-out
    set group_order             = BISTGRP0
    set top_module_name         = top             # design top
    set TAP_hierarchy           = top             # design_top sub_module_instance_name ...
    set verilog_path            = ./mbist/RP_default_INS_FAULT.f            # /absolute path/co
    set work_path               = ./integ             # ./work
    set bist_integ_path         = ./mbist/RP_default_spec.integ | ./mbist/top_default_spec.integ
    set parsing_mode            = RTL_only        # hybrid, RTL_only, Netlist_only
    set gate_cell_lib             = ./gate_cell.info # gate clock standard library
    set gate_cell_with_mux        = ./mux_cell.info    # mux standard library
```

圖 8. BII 設定

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

## 2.3 執行結果

圖 9 為執行完後架構圖，圖 9 的右圖為未使用 MUX 的架構圖，圖 9 的左圖為有使用 MUX 的架構圖，依使用者需求設計，cell 插入後，我們會用 define 做區隔，電路結果如圖 10。

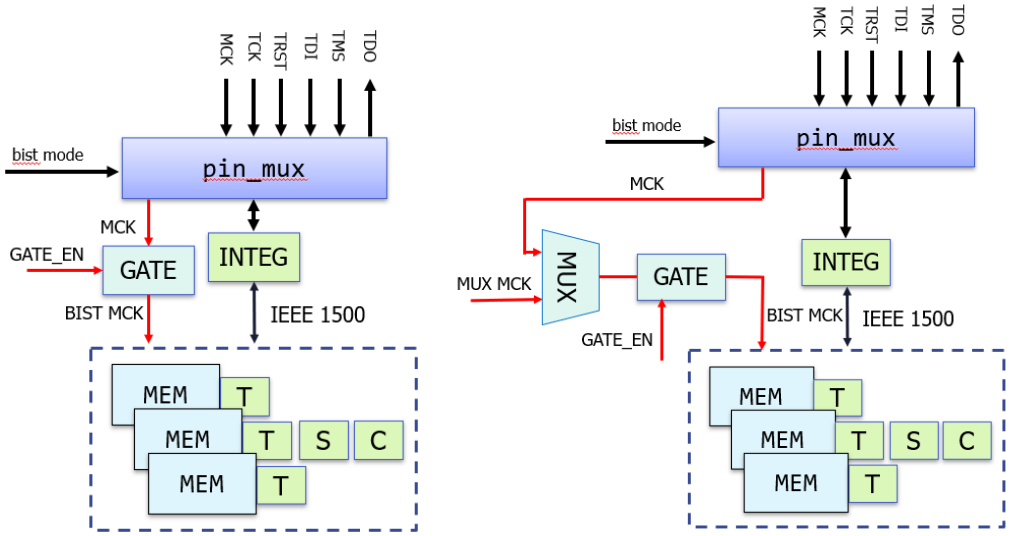


圖 9. 執行完後架構圖

```
`ifdef SYNTHESIS
    GMUX2D2BWP30P140ULVT INTEG_RP_default_MCK_MUX(
        .I0(INTEG_RP_default_GCK_MUX),
        .I1(INTEG_MUX_CLK), .S(INTEG_MUX_SEL),
        .Z(INTEG_RP_default_GCK));
    CKLHQD12BWP35P140ULVT INTEG_RP_default_MCK_GATE(
        .CPN(INTEG_RP_default_GCK),
        .E(INTEG_GATE_EN), .TE(1'b0),
        .Q(INTEG_RP_default_MCK));
    assign      INTEG_RP_default_GCK_MUX = MCK;
    GMUX2D2BWP30P140ULVT INTEG_top_default_MCK_MUX(
        .I0(INTEG_top_default_GCK_MUX),
        .I1(INTEG_MUX_CLK),
        .S(INTEG_MUX_SEL),
        .Z(INTEG_top_default_GCK));
    CKLHQD12BWP35P140ULVT INTEG_top_default_MCK_GATE(
        .CPN(INTEG_top_default_GCK),
        .E(INTEG_GATE_EN),
        .TE(1'b0),
        .Q(INTEG_top_default_MCK));
`else
    assign      INTEG_RP_default_MCK = MCK;
    assign      INTEG_top_default_MCK = INTEG_top_default_GCK_MUX;
`endif
```

圖 10. 執行完後電路

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

# Multi-Chain

為了符合 Low Power Design，START 推出了 Multi-Chain 的功能，可以提供使用者依電路需求分群，不用像過往一樣全部同時測試，耗費較多的電，可以依照分群來個別測試，當前不需動作的群組，則將 BISR 斷電，省下該群的功耗，達到 Low Power Design。

## 1. 測試範例架構圖

圖 11 為測試範例架構圖，圖中皆為 repair controller，BISR 在 BII 流程中會將所以 repair controller 串接起來，Multi-chain 的功能讓使用者可以依照需求，將 repair controller 分群，個別測試。

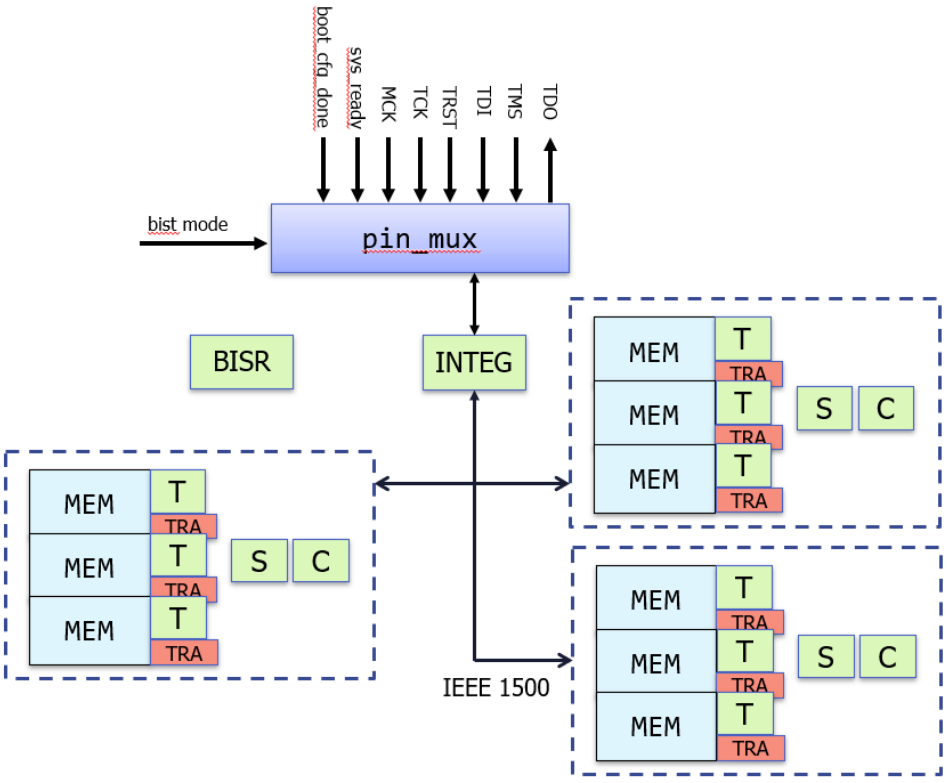


圖 11. Multi-Chain 測試範例架構圖

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

## 2. 測試範例實作

使用該功能，需在 BII 設定檔中，將所要分群的 repair controller 填入，並且需要 hookup 每個群的 sys_ready，START 便會依照所填入的資訊分群。

### 2.1 BII file 設定

如圖 12 所示，將分好的 controller name 填入 chain_order 中，由於每一群會有各自的 sys_ready 讓使用者告知 BISR 電路已經可以動作，因此需要 hookup 每一群的 sys_ready，如圖 13 所示。

```
define{powerchain}[pd1]
  set chain_order           = RP_chip_default, RP_chip_default_1
end_define{powerchain}

define{powerchain}[pd2]
  set chain_order           = RP_s_default
end_define{powerchain}
```

圖 12. BII 設定

```
define{hookup}[integ_sys_ready_pd1]
    set dedicate_port       =               # EXAMPLE_PORT
    set mapping_port        = chip spw_rxd_p            # top example_SYS_READY
end_define{hookup}

define{hookup}[integ_sys_ready_pd2]
    set dedicate_port       =               # EXAMPLE_PORT
    set mapping_port        = chip spw_rxd_p            # top example_SYS_READY
end_define{hookup}
```

圖 13. BII 設定(Hookup)

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

## 2.2 執行結果

由圖 14 所示，執行完後，會依照設定的 pd1、pd2 與 BISR 串接起來，等待 sys_ready 送信號進去便開始 work。

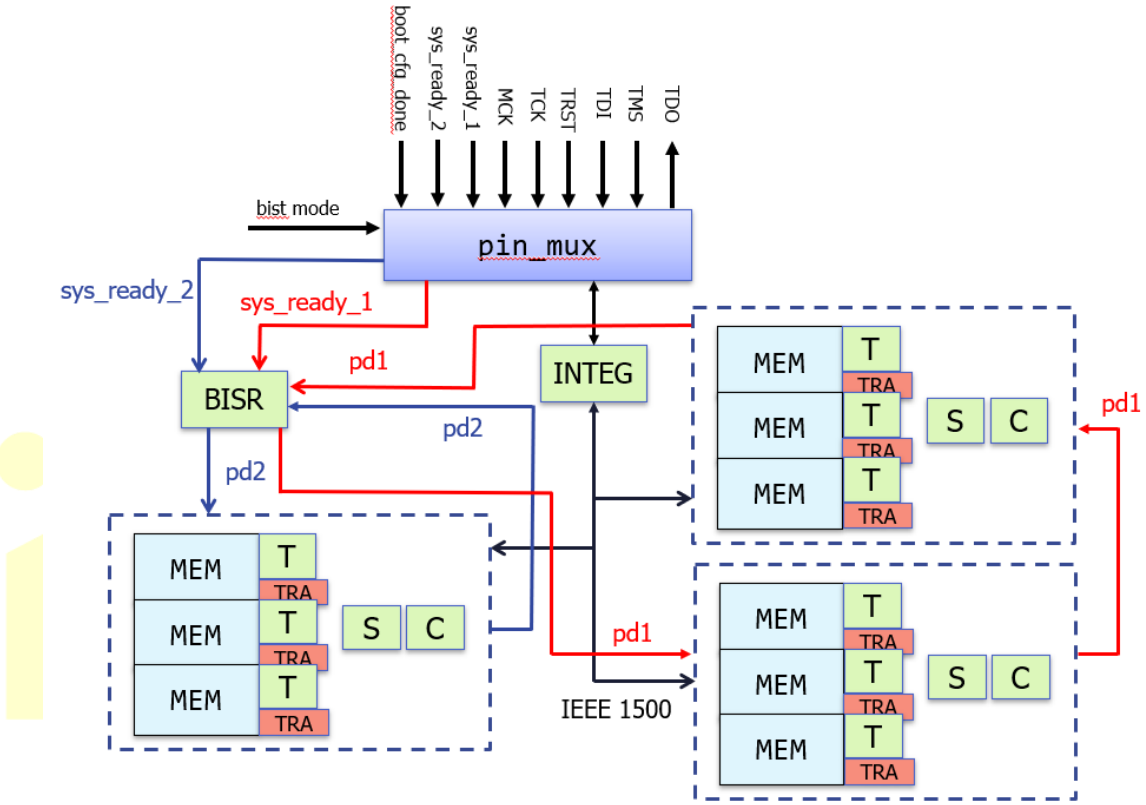


圖 14. 執行完後架構圖

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

## 總結

AI 晶片內會使用到大量的記憶體，因此設計檢測記憶體電路相當重要，此時 START 就能有效地縮短 DFT 工程師的時間，SoC 內通常會有許多相同性質的模組重複呼叫，Bottom-up 功能便能有效地縮短設計 BIST (Built-In Self -Test)及 BISR (Built-In Self -Repair)的時間，除了縮短設計時間，START 採用多層 block 的設計，在每個 block 下都能做記憶體檢測，確保每個 block 的正確性，也能縮短重複呼叫 block 的驗證時間，現今 Low Power Design 越來越重要，Auto-Clocking gating cell insertion 及 Multi-Chain 就相當符合這一趨勢，由於記憶體檢測電路不需要無時無刻都作用，因此透過 Auto-Clocking gating cell insertion 可以在 BIST 及 BISR 的 clock 前加入 gating cell 及 mux，來減少功耗，在使用 Multi-Chain 來分群，不需要每次都全部測試，可以依照需求選擇群組測試，省去不需要的群組所造成的功耗，達到 Low Power Design 的目的。

比起過往人工設計相關測試電路，使用工具自動產生可以減少設計的時間，在瞬息萬變的市場中，能夠先人一步便能搶得先機，因此透過 START 能夠縮短 DFT 設計的時間，進而加速晶片生產的時間，搶先推出新的 AI 晶片，讓客戶在這場 AI 晶片的市場中，搶得先機。

iSTART
芯測科技(股)公司
台灣新竹縣竹北市台元一街 6 號 7 樓之 5
T:+886-3-560-1667
www.istart-tek.com

## 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:**

顏瑞穎

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