

Advantest V93000 PS1600 基礎操作簡介

陳泓烈 hlchen@cic.narl.org.tw

前言

本中心自 2012 年 4 月引進最新 Advantest V93000 SoC Series 混合訊號自動測試機台，Model 為 SmartScale C-CLASS，命名為『**Advantest V93000 PS1600**』。

此機台之數位模組部分，計有 512 個 digital channels 可供使用；全部 512 個 channels 的 data rate 均可達 200Mbps；而其中有 256 個 channels 的 data rate 達 533Mbps；再其中有 128 個 channels 可達 1.6Gbps 的 data rate，以上所有 data rate 相關的 channel license 均可視各 DUT 需求任意調整。所有 digital channels 的 pattern memory 均有 64MVectors 的深度。

在類比模組部分，計有 high resolution analog instrument 及 high frequency analog instrument 各 4 組；其中 high resolution analog instrument 部分均為 24bits 解析度，計有 200Ksps 的 Digitizer 及可產生 sine wave frequency **125KHz** 的 Waveform Generator (maximum sampling rate **1Msps**)；而 high frequency analog instrument 部分有 180Msps 的 Digitizer 及可產生 sine wave frequency **80KHz** 的 Waveform Generator (Maximum sampling rate **200Msps**)。

機台另外可提供 device power supply channels 達 32 組，其於大電流模式可提供最大電壓 3V 及最大電流 1.5A，大電壓模式則可提供最大電壓 7V 及最大電流 0.5A；另外亦可支援將數根或全部 channels 的電流連結使用，最可達 48A 電流。

軟體操作介面部分，新機台改行 SmarTest7 以上版本，此版本相較 SmarTest4/5/6 版本而言，捨棄以往著重於工程開發之介面，全面改成強調於量產測試之介面。因此在新版本上我們可以很方便地設定 Testflow 及 TestMethod，並且非常容易設定 Testsuite parameter 及測試環境。

本機台與中心舊款 Agilent 93000 SoC Series 之硬規格比較如下表所示：

<i>Agilent 93000 SoC Series Advantest V93000 PS1600</i>		
<i>Digital Instrument</i>		
Channel Count	320 channels	512 channels
Data Rate	660 Mbps	1.6 Gbps / 533 Mbps / 200Mbps
Vector Memory	28 MVectors	64 MVectors
Edge Placement Accuracy	100 ps	80 ps
<i>Analog Instrument</i>		
High Speed Digitizer	12 bits, 41Msps, 100MHz	16 bits, 180Msps, 30KHz
High Resolution Digitizer	16 bits, 2Msps, 3MHz	24 bits, 200Ksps, 150KHz
High Speed AWG	12bit, 500Msps, 125MHz	16 bits, 200Msps, 80KHz
High Resolution AWG	16bit, 30Msps, 7.5MHz	24 bits, 1Msps, 125KHz
Channel count	4	8
<i>Device Power Supply</i>		
Channel count	8	32

一、簡介

自動測試設備的使用為目前晶片產出之重要工具，其功用在於可以快速且正確地量測晶片之各項參數及功能完整性，其操作訊號更可以達到 GHz 等級以上。量測時，自動測試設備將可自動輸入測試訊號並觀測待測物(Device Under Test, DUT)的輸出響應且比對預測結果，因此可自動判斷其 DUT 之 good(pass)或 bad(fail)；由於其功能皆可自動執行，因此可大大增加晶片之產出，其完整流程如圖 1.所示。

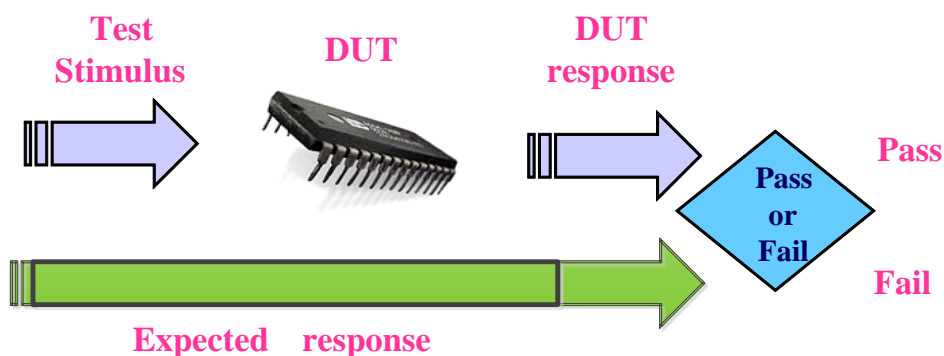


圖 1. Automatic Test Equipment Testing flow

CIC 目前所引進 Advantest V93000 PS1600 自動測試設備其外觀如圖 2.所示，其中 testhead 內部即為所有 channel board 所存放位置，待測試的 IC 則是透過 DUT board 及 DUT interface 與 testhead 內部的 channel board 連接在一起，manipulator 內部放置鉛塊來平衡 testhead 的重量。故 DUT 透過 DUT Interface 與 channel board 連接，channel board 內含許多 DSP 模組，用來產生 DUT 的輸入訊號及分析 DUT 的輸出訊號。

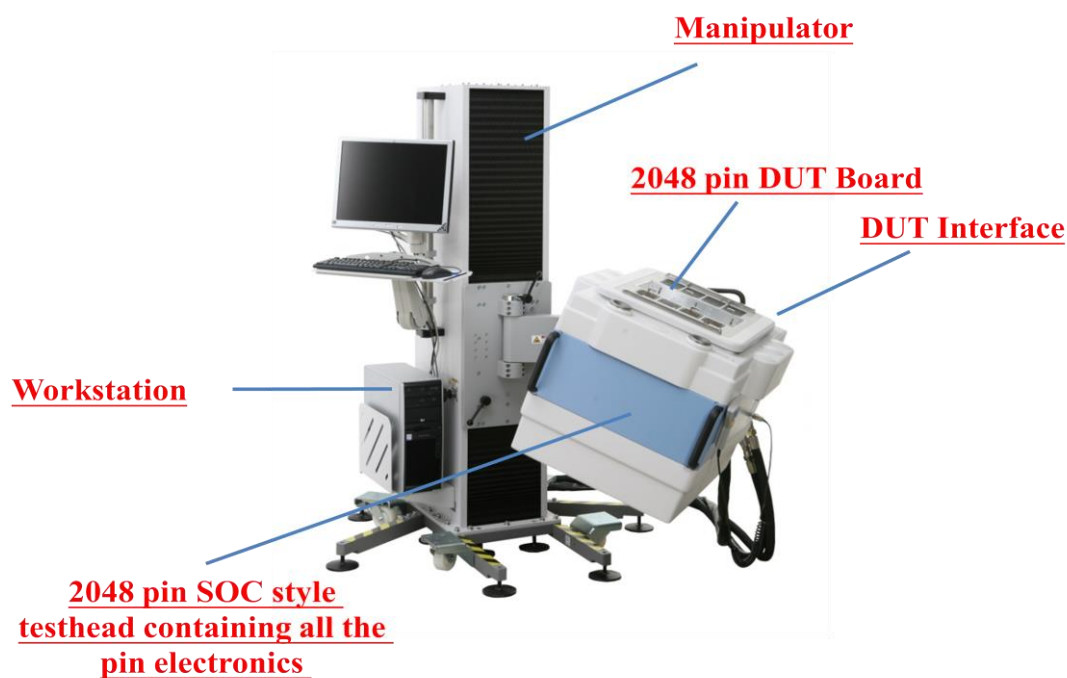


圖 2.Advantest V93000 PS1600 外觀圖

Advantest V93000 PS1600 測試設備的操作軟體名為『SmarTest』，我們可透過 SmarTest 進行量測程式的設定及測試；亦可由軟體上的操作介面控制測試設備內部硬體的動作。使用 SmarTest 及測試機台進行數位 IC 測試的流程如圖 3. 所示，共分成 Test Plan、Design a DUT Board、Pin Configuration、Level Setup、Timing Setup、Vector Setup、Testflow Setup、Testing the Device 以及 Result Analysis Tool 等九大步驟。在本節中我們將分別先對九個步驟作解釋說明。

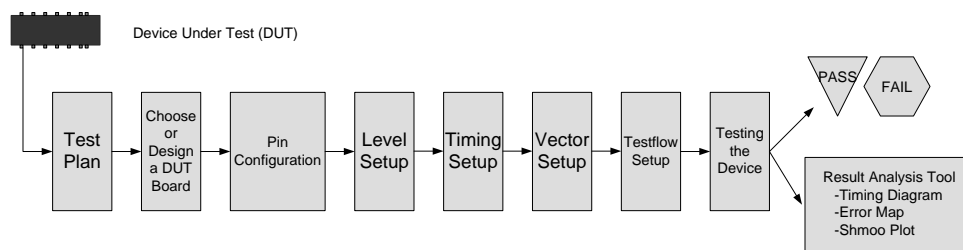


圖 3. 測試開發流程圖

本流程中[1]及[2]步驟為事先規畫及硬體環境準備，步驟[3]之後則是採用 SmarTest 進行軟體操作。

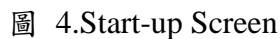
- [1]. Test Plan：此步驟表示量測前需事先規劃想要測試的項目有那些：如 functional 的測試、scan pattern 的測試、power dissipation 量測.....等。
- [2]. Design a DUT Board：本系統需藉由一個 DUT board 將待測 IC 以及 testhead 上的 pogo pin 連接在一起，如此才能進行 IC 測試，CIC 針對目前提供的 8 種 IC package(DIP48、PLCC68、PLCC84、CQFP100、CQFP128、CQFP144、CQFP160、CQFP208) 都有相對應的 DUT board 可以使用。使用者需選擇其適合的 DUT board 後，在 DUT board 中將此 DUT 的 Power/Ground 以跳線方式將其與機台的 DPS (Device Power Supply) 模組連接。
- [3]. Pin Configuration：設定待測 IC 每個 signal pin 的名稱，以及每個 signal pin 連接到哪個 test channel，以及使用那些 device power supply 的模組來供應電源給待測 IC。
- [4]. Level Setup：設定 power supply 的 voltage 大小，current limit 的大小，訊號的 drive voltage 大小(VIL、VIH)，訊號 compare 的 voltage threshold 大小(VOL、VOH)等。
- [5]. Timing Setup：主要設定每個訊號輸入及輸出波形的樣式，以及 system cycle 的週期與頻率。
- [6]. Vector Setup：設定要測試的測試向量，其設定方式為利用 system cycle 來設定，因此可結合 Timing Setup 的設定將每個訊號在每一 system cycle 時的波形建立出來。
- [7]. Testflow Setup：將想要測試的項目及各項目的特殊設定，設定成一個流程，流程中亦可包含分支或次數等各種控制項。這些項目包含軟體內建的 test suite 及使用者自訂的 testmethod，當內建 test suite 無法滿足使用者所需要的測試功能時，使用者就必須自己設計一個 testmethod。由於一般的數位晶片量測項目均可使用內建 test suite，故本文就不再針對 testmethod 多做介紹。
- [8]. Testing the Device：使用已設定完成之 testflow 開始進行測試，機台會根據前面步驟所作的設定產生 input stimulus 給待測 IC，同時也會量測待測 IC 所產生的 response，並且與預期的 response 作比較，若比較相同就是通過測試(PASS)，若不相同就是有錯誤(FAIL)。
- [9]. Result Analysis：分析量測的結果，可使用 timing diagram 來看波形（又可分為 standard mode、High Resolution Mode 及 Scope Mode 等方式），可使用 error map 來發生錯誤的點是在哪個 system cycle，另外亦可使用 shmoo plot 來分析待測 IC 的特性資料。

二、 Software Overview

開啟 SmarTest 軟體的方式可分為 offline mode 與 online mode，online mode 即有連接機台，可以進行實際測試；offline mode 只能純粹進行軟體設定，不能

進入 offline mode 的指令為 ***unix%> HPSmarTest -o &***。

進入 online mode 的指令為 ***unix%> HPSmarTest &***。



在 Eclipse Workcenter (圖 5.) 左側有一連串的設定 icon，整個量測所需設定將會在此處完成，上方中間 (紅色框框處) 有 4 個 tabs 分別是 connect、disconnect、

break 以及 change device, connect 是用來連接測試機台上的 AC relay, disconnect 是用來切斷機台上的 AC relay, break 是用來停止 IC 的測試, change device 則是用來切換測試不同的 IC 或新增一個待測的 IC。

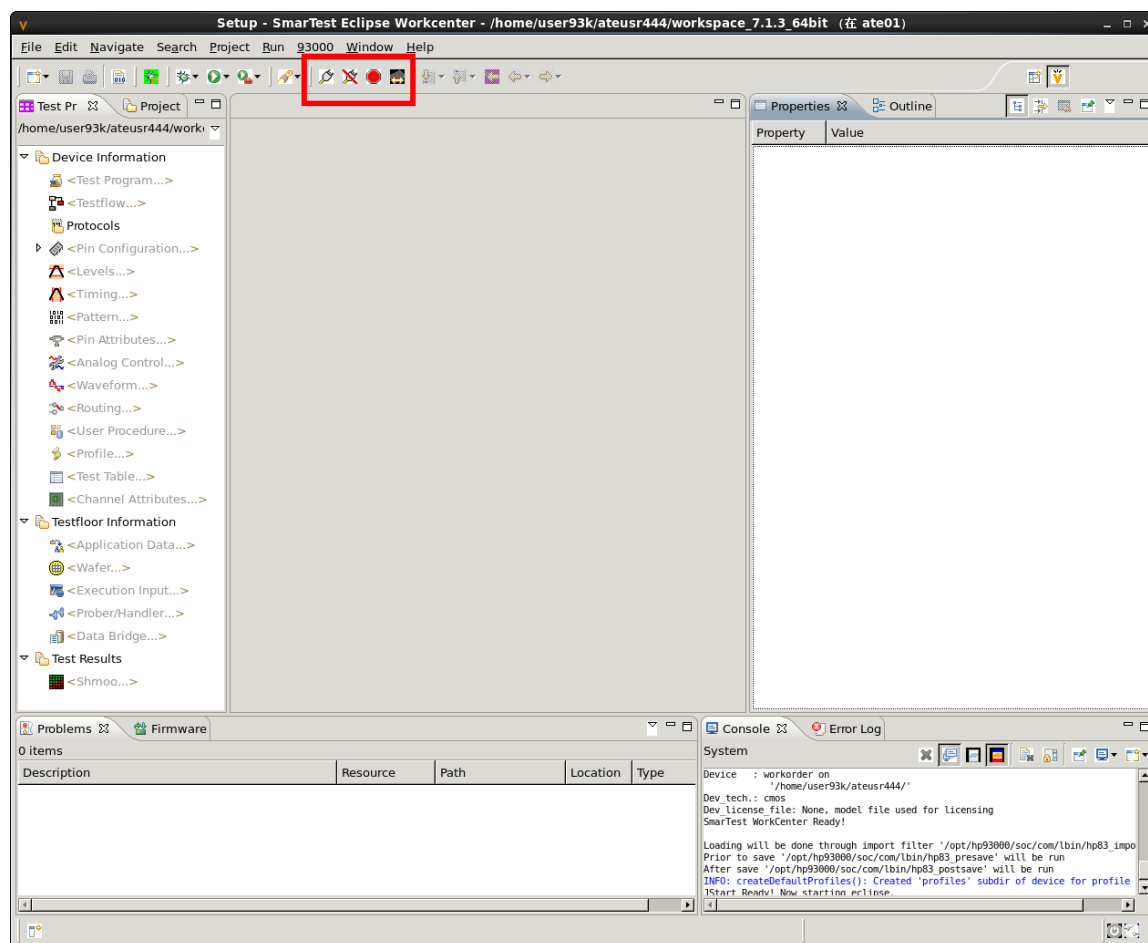


圖 5. Eclipse Workcenter

第一次要開始進行測試程式開發時，SmartTest 會先以 Project wizard 的方式將 Device 建立出。Project wizard 流程首先是 Select device 如下圖 6.所示，此步驟目的在於指定這 Device 資料存放的位置，選擇好位置後，按下 Create device，接著按下確認的 OK 鍵即可。接著則是選擇 Workspace 資料夾，此資料夾將用來當作 SmartTest 程式操作的工作空間及其所有暫存資料，流程如圖 7.所示。接下來 SmartTest 就可建立起 Device。一般來說，我們會用一個 Device 來代表一個晶片的 Test program。之後的操作步驟則將在這個被建立起的 Device 中完成。

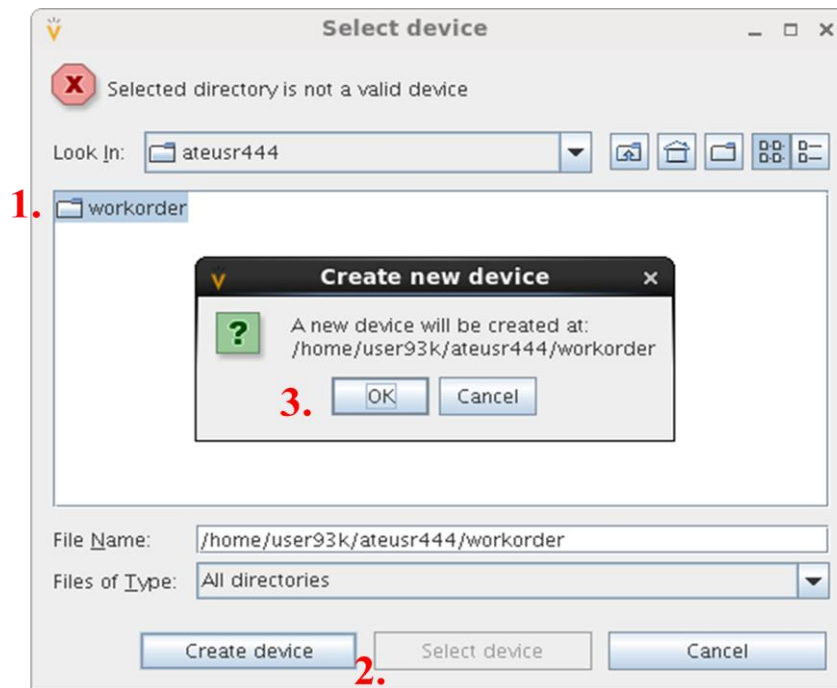


圖 6. Select the Device

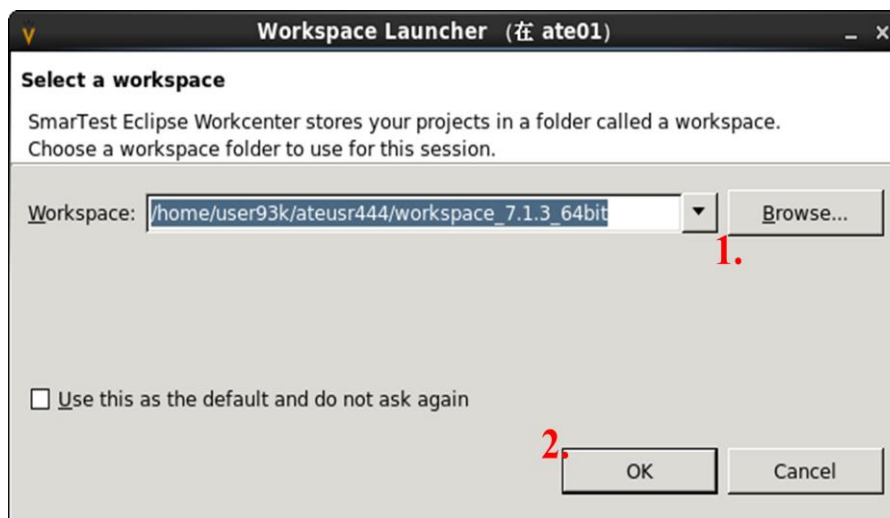


圖 7. Workspace Launcher

三、 Pin Configuration

接著要進行 pin configuration (Channel configuration) 設定，必須在 Eclipse Workcenter 的左手邊，右鍵點選 Pin Config Icon 如圖 8.所示，接著選擇 Load，根據每顆 IC 所使用的封裝 (package) 種類，選擇一個 template 檔案，由於 CIC 所提供的 DUT board 上，每個 package 上的每根 pin，對應連接到測試機台的哪個 test channel 已經固定，因此 CIC 事先提供每種封裝的 template 檔案，如此使用者只需讀取這 template 檔案後，修改檔案中訊號的名稱，以及訊號的種類是輸入或輸出即可，不用去查詢此訊號是連接到機台上的那個 test channel。本 template 檔

案可在『/home/user93k/ ateur/ate_env_file/configuration』中找到。

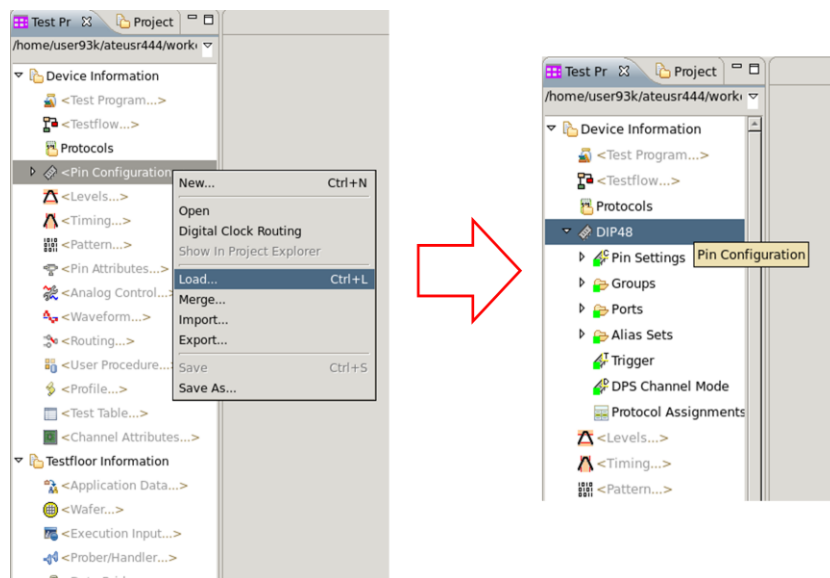


圖 8. Load the Pin Configuration

讀取之後，可以連點兩下打開，接著需設定每根 pin 的定義。首先選擇到『Digital Pins』，可以見到內容清楚的定義著每根 pin 的 Name、Type 及 Test Channel 等資訊。使用者必須將 pin name 及 pin type 修改成符合這顆 Device 的名稱及型式，這是個有點類似 Excel 的表格，因此直接點擊後修改名稱即可，如圖 9.所示。pin type 可以使用 i (Input Pad)、o (Output Pad)、io (Bidirectional Pad)、off (Turn Off) 等選項；而如果想知道 input pad 的機台輸入波形變化，可以將其設定為 io 亦無妨，但是在之後的設定中就必須對這根 input pad 使用如同 bidirectional pad 的設定方式。而由於 CIC 所提供的 loadboard 上，每個 package 上的每根 pin 對於連接到測試機台的 test channel 已經固定，因此使用者不需查詢 pin 與 test channel 的連接對應，當然也不用修改這部分的內容。另外對於 power/ground pin 的處理方式，由於在 CIC 使用本機台前，使用者必須以接線方式將 power/ground pin 以跳線方式連接到 loadboard 的 DPS connector 上，因此使用者需在 Pin Configuration 設定上將 power/ground pin 刪除，方式為按下表格最左邊的號碼，然後點右鍵選擇 Delete 即可。對於沒有接到任何 DUT pins 的 package 腳位也一併刪除。

一般我們在 IC 設計流程上，大多習慣對 Bus 進行設計，例如：Data[7:0]。但對晶片量測來講，每根 pin 都是獨立的個體，因此我們可以使用 Pin Group 方式將某些 pins 集成 group。而另外為了 Level、Timing 及 Vector 設定上的方便，我們也可以將某些將擁有同樣設定內容的 pins 設定成 group，如此只要針對 group 進行設定也就等同於對 group 內的所有 pins 進行設定。其方式為在 Pin Setting > Groups 按下滑鼠右鍵選擇 New，接著輸入 Group Name，然後從 Group Editor 右邊的 Available Pin List 中選擇要加入的 pins，然後按下 Add 將之複製到右邊即可，

其排序為上面的 pin 為 MSB，下面的 pin 則為 LSB，同樣的方式也可以利用 Remove 把 group pin 進行刪除。步驟流程如下圖 10.所示。

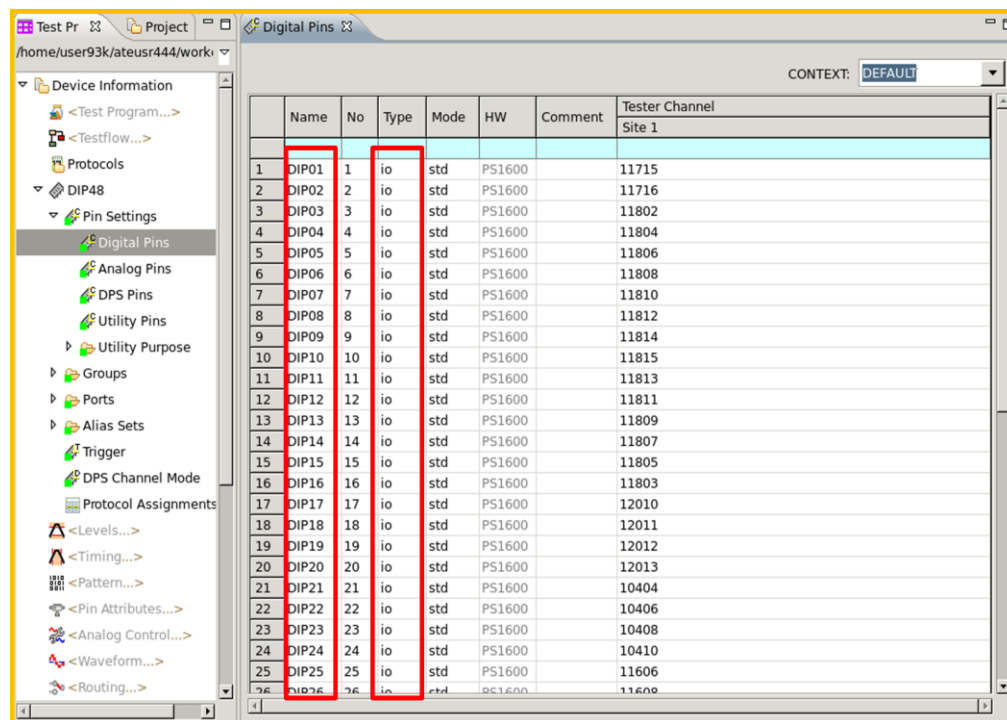


圖 9. Digital Pin Settings

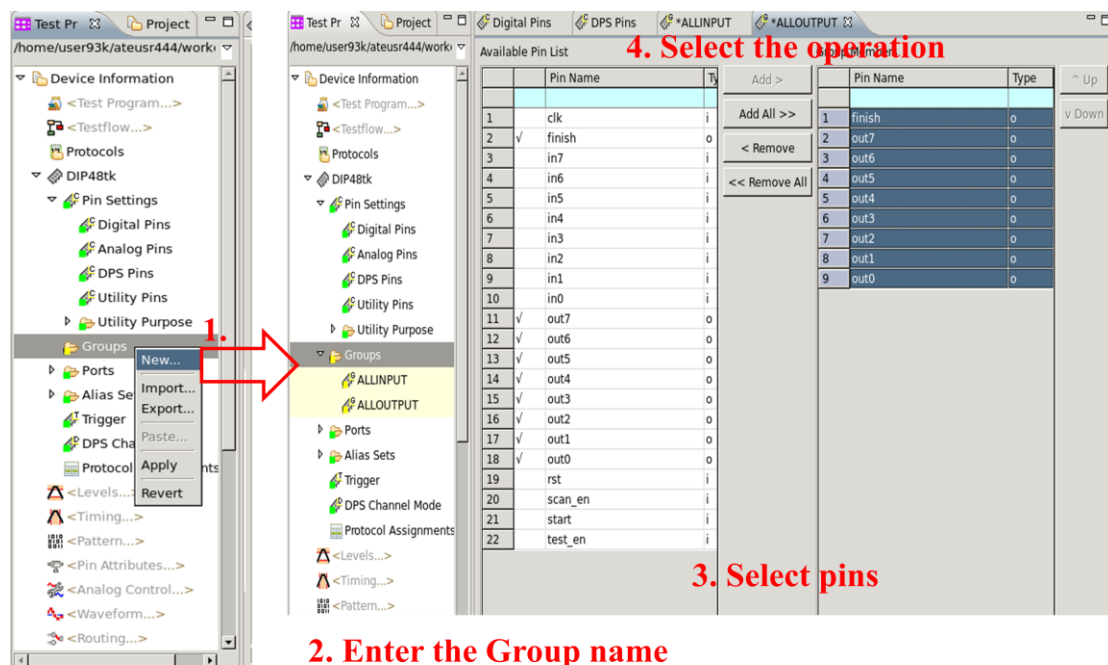


圖 10. Group Setting

而對於 DPS pins 的處理上，我們使用的單心線將之跳線，並於 Digital Pins

設定中將之刪除，接下來我們需正確的設定其 DPS 組態內容。設定上需點選 Pin Settings > DPS Pins 後打開 DPS Pins Editor，接著修改其 Test Channel（如圖 12. 所示），這步驟需視 loadboard 跳線上所連接的位置來設定，而實體跳線與設定內容可參考圖 11. 所示。例如使用 DPS13 跳線時，則 Test Channel 設定為 22505。Ground pin 則不需額外設定，另外若有 multi power domain 的話，可以直接在此表格左手邊上按下右鑑後選擇新增一行後，再依照其他 DPS pins 設定內容將資料填入，但 TestChannel 必不會重複。

Channel mapping table

DPS32 channel	MSDPS channel
22501	11
22502	15
22503	12
22504	16
22505	13
22506	17
22507	14
22508	18
22509	21
22510	25
22511	22
22512	26
22513	23
22514	27
22515	24
22516	28
42501	31
42502	35
42503	32
42504	36
42505	33
42506	37
42507	34
42508	38
42509	81
42510	85
42511	82
42512	86
42513	83
42514	87
42515	84
42516	88

圖 11. DPS32 Mapping Table

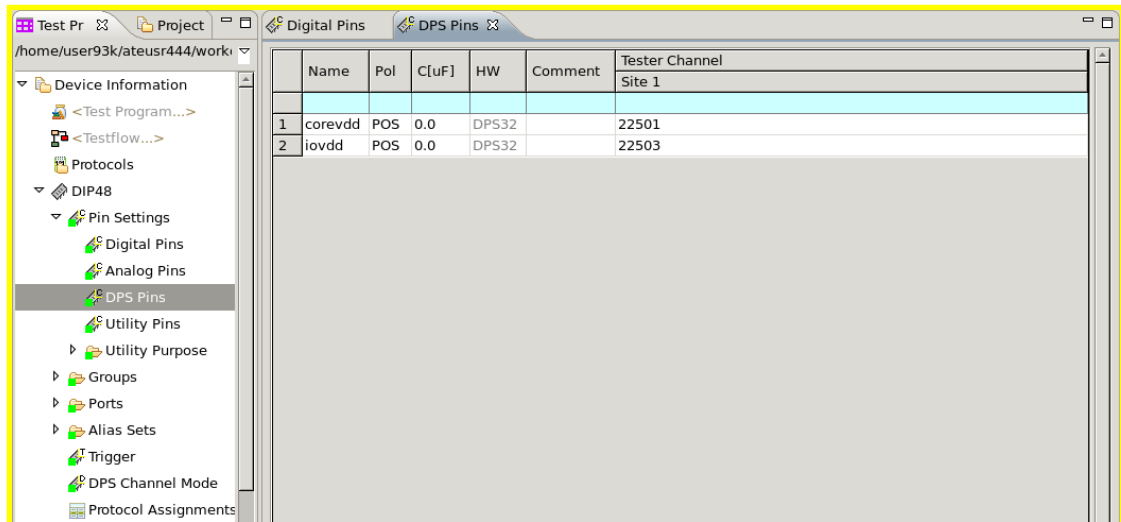


圖 12. DPS Pins Setting

進行完以上設定後，使用者必須將設定內容 Apply 到機台上，其方式為在 Pin Settings 圖示上按下右鍵選擇 Apply 即可（如圖 13.所示）；完成 Apply 後，此設定內容才會進入機台內。

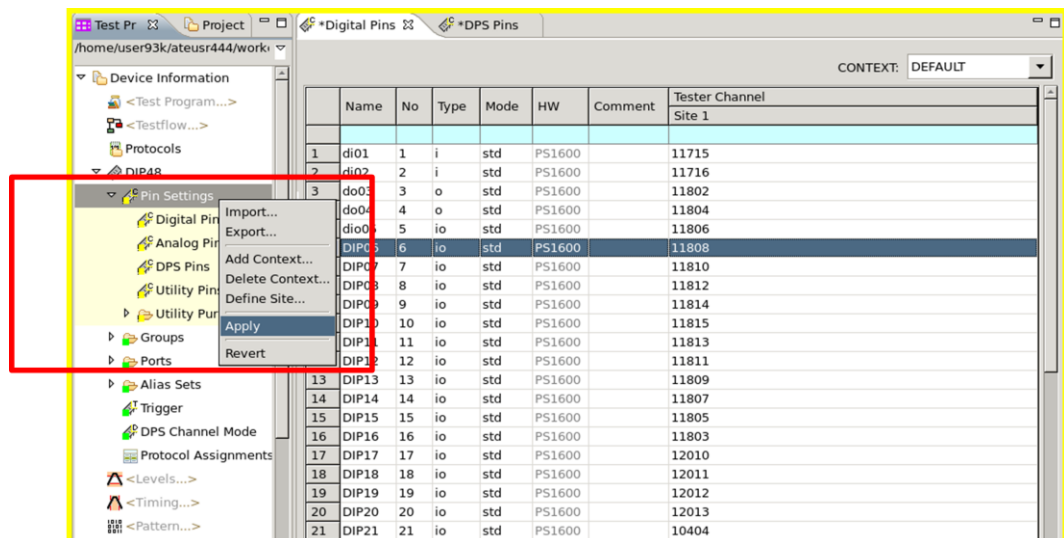


圖 13. Apply the Setting

完成以上步驟後，接著就是存檔。其方式為在 Pin Configuration 圖示上按下右鍵選擇 Save 即可，如圖 14.所示。存檔完成後就完成了 Pin Configuration 的設定步驟。其他諸如 Levels、Timing 及 vector 設定之存檔方式亦如此步驟。

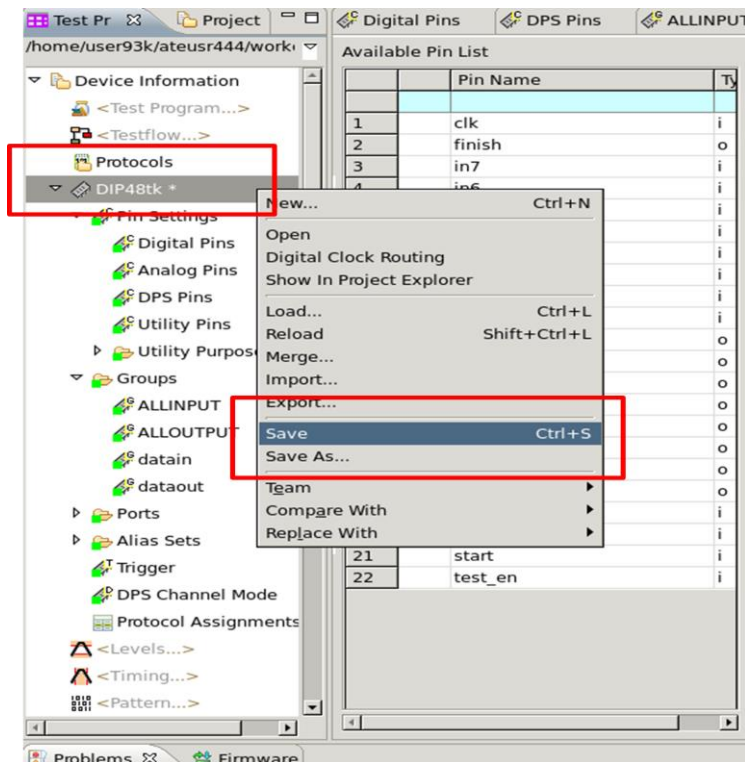


圖 14. Finish the Configuration

四、 Defining the Levels

Level Setup 是設定訊號的 drive voltage 大小(VIL、VIH)如圖 15.所示，訊號 compare 的 voltage threshold 大小(VOL、VOH)如圖 16.所示，以及 power supply 的 voltage 輸出腳 (vout)，current limit 的大小 (ilimit) 等多項參數，如圖 17.所示。設定的方式是在 Eclipse Workcenter 的左手邊，點選 Levels Icon 如圖 18.所示，右鍵選擇 New 後，輸入此 Level Setup 檔案名稱，接著連點兩下此 icon 後就會出現 Level Setup 的視窗，便可以開始進行設定，其設定的架構如圖 19.所示，可以利用 Level Equation Editor 來設定 level equation set，每個 level equation set 中可以利用 SPECS 以及 EQUATIONS 來定義一些變數以及數值運算。另外在同一個 Device 中可以設定多組 Level Equation Set 及多組 Level Spec，我們可在 testflow 中搭配選擇使用不同的 Equation 及 Spec。

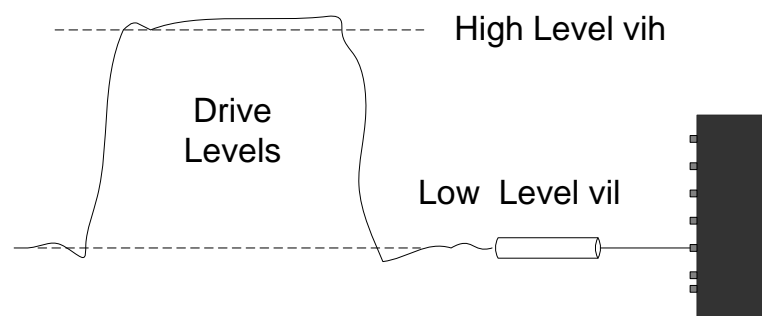


圖 15. Drive Levels

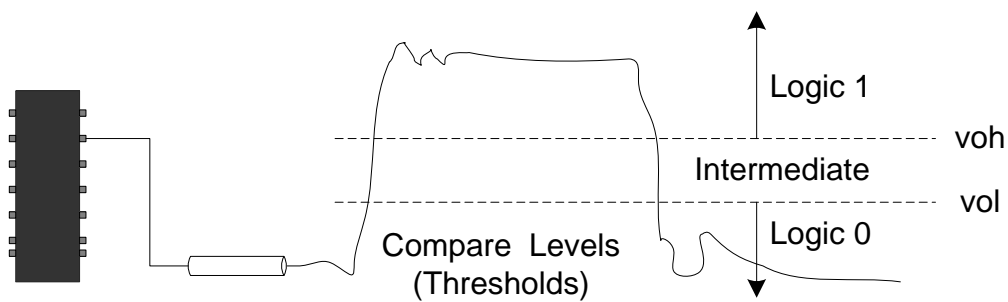


圖 16. Compare Levels

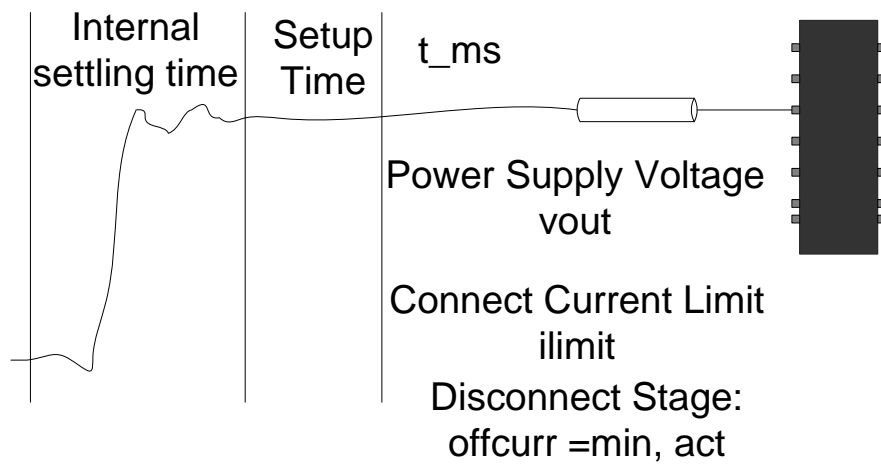


圖 17. DPS Voltage and Current Limit

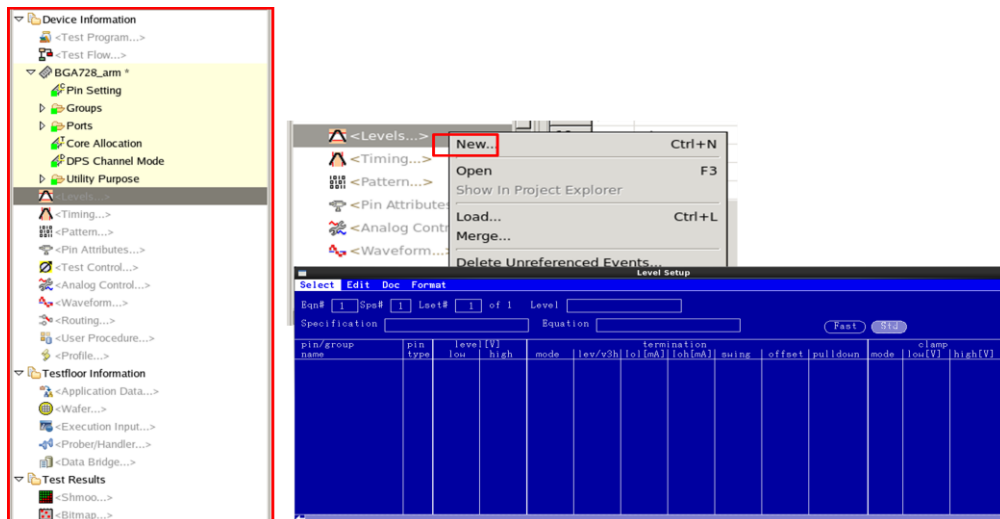


圖 18. Level Setup

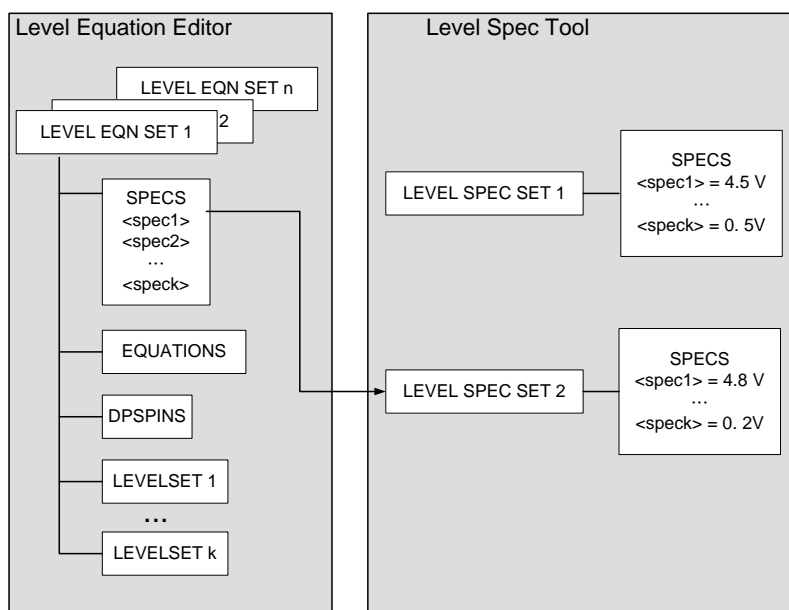


圖 19. Level Setup Architecture

在 Level Setup 設定步驟中，使用者必須先在 Level Setup 視窗中選擇 Select > Edit Equations 開啟 Level Editor 以進行 Level Equation Set 設定。在這步驟中，使用者可以直接拿 CIC 事先準備好的 example 檔案內容來進行編輯修改，編輯完成之後在 Editor 中選擇 Shell > Download 就可以將設定內容下載到機台上(此步驟有點類似 Pin Configuration Setup 的 Apply 步驟)，如圖 20.所示。CIC 事先所準備好的 level template 檔案存放路徑為

『/home/user93k/ateusr/ate_env_file/level_template』文件。

Level Equation Set 內容如圖 20.範例所示，在 SPECS 中設定了 corevdd、iovd、VIL、VIH、VOL 及 VOH 等六個 SPECS，這些 SPECS 變數將可以在 Level Specification 指定其數值，每個 EQUATIONS 可以指定多個的 SPECS，另外亦可利用 EQUATIONS 再將這些 SPECS 參數做一些計算而產生更多參數來方便後續設定使用。DPSPINS 是用來定義 Device Power Supply 的相關參數，其中 vout 為指定此組設定對應到 Pin Configurations 設定上的哪一根 DPS 腳位，ilimit 設定為限制最大可供給電流，t_ms 為設定電源供給的 setup time，offcurr 設定為 act 表示 DUT 不動作時允許系統電容可以 discharge。LEVELSET 則是用來定義每一訊號的 vih、vil、voh、vol 等參數，通常 input pad 只需要設定 vih 及 vil 表示輸入訊號的高低準位，而 output pad 只需要設定 voh 及 vol 表示判別輸出訊號為 H 或 L 的 thresholds，bidirectional pad 則是這四個參數都需要設定；在此步驟中，我們可以利用 Pin Configurations 的 Groups 來同時設定許多 pins 為同一種設定內容。

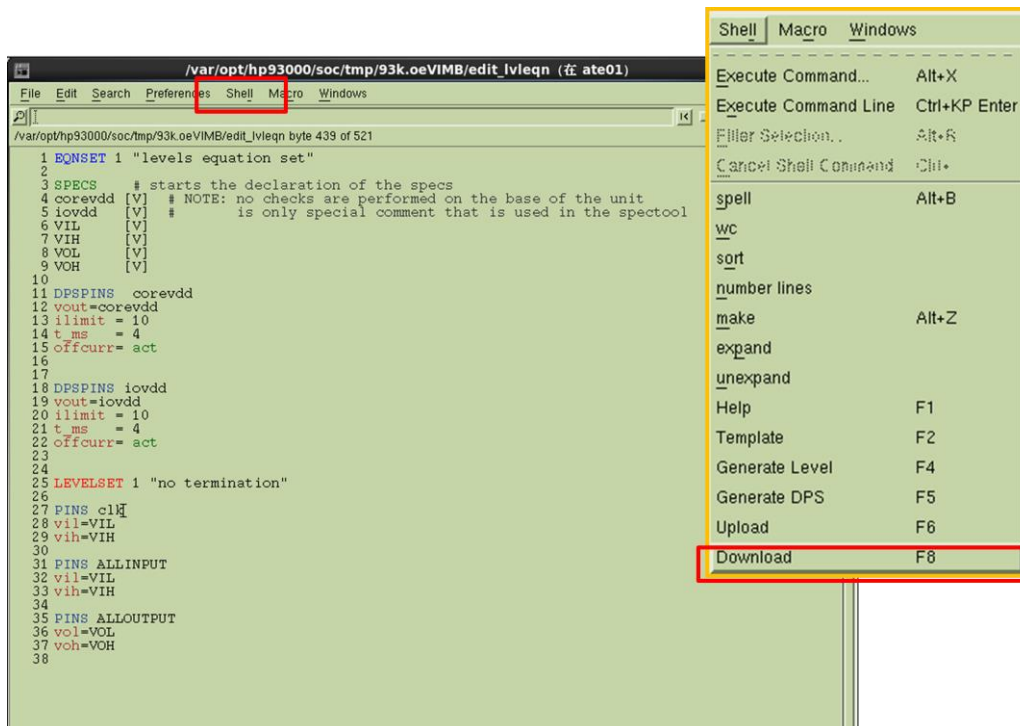


圖 20. Download the Level Equation Setup

設定好 Level Equation Set 之後必須針對所有的 SPECS 變數指定其數值，指定的方式必須透過 Spec Tool 來進行，開啟 Spec Tool 的方式之一是在 Level Setup 視窗中選擇 `Select > Edit Specifications`，然後會出現如圖 21.的視窗，選擇 level equation set 之後點選 `create` 按鈕，會出現 `Create level spec set` 的視窗如圖 22.，在 `Create level spec set` 的視窗選擇 `Set number` 後，在 `Description` 欄位填入名稱，然後點選 `create` 按鈕，便會出現 `Spec Tool` 的視窗(圖 23.)，在 `Spec Tool` 的視窗中，便會出現所有的 SPECS 變數，可以在 `Actual` 的欄位填入變數的數值，輸入完成之後選擇左上方向下的箭頭（如圖 23.所示）即執行 `download` 指令。`download` 之後以同樣方式在 `Eclipse Workcenter` 中 `Level Icon` 進行存檔即完成設定。

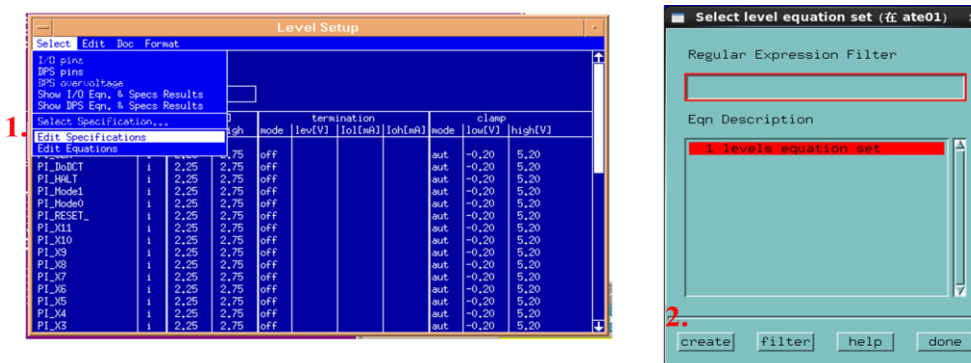


圖 21. Create Level Spec Sets I

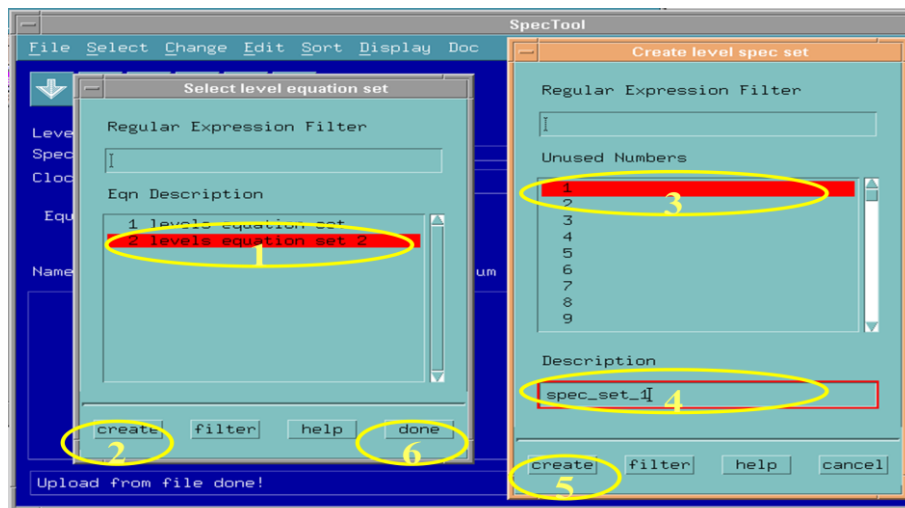


圖 22. Create Level Spec Sets II

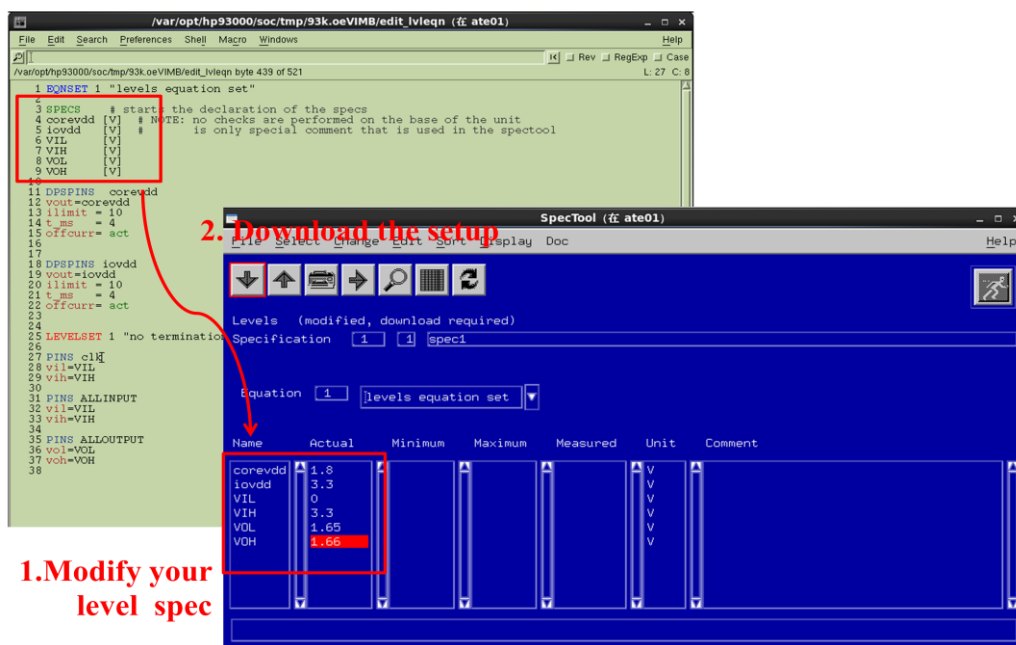


圖 23. Spectool

五、Timing and Vector Setup

在講到 Timing and Vector Setup 之前，必須先談到本測試機台的 license 概念。本測試機台俱備 512 根可達 200Mbps 的 digital channels；其中又有 256 根可達 533Mbps；在其中又有 128 根可達 1.6Gbps。由於這些 digital channel license 非固定於其指定 channels 上，而是可視使用狀況來調整的，因此使用者在開啟 SmartTest 之前必須先編輯測試機台 model 設定，使其 digital channel operation frequency 可適用於待測 Device 上。其流程為使用 superuser 登入後將『/etc/opt/hp93000/soc/model』檔案複製回原登入帳號之家目錄下，接著對應每個不同 package 的 test channels 將 model file 內容中關於 digital channels 的部分做

修改。欲修改部分為 model file 內容如下圖 24.位置，其格式為：

Test Channels number : HW=PS1600, speed={ max-rate,533Mbps,200Mbps}, smem=64M, diff

使用者可觀察其 Device 的 Pin configuration 設定上的 Test Channel number 後，將之寫在 model file 內並宣告其適當的 speed 即可，而 HW=PS1600 指的是本測試機台內模組型號，因此不需改變。smem 指的是 sequence memory 深度，這也是跟機台模組有關，因此也不需改變。最後一個 diff 則是指此 test channels 可用於 differential test。而修改完 model file 並覆蓋為原路徑檔案後，需重新開啟 SmarTest 方可生效。

```
IOCHANNEL
# Smart_Scale System
10101-10816:HW = PS1600, speed = max-rate, smem = 64M, diff
10901-11616:HW = PS1600, speed = 533 Mbps, smem = 64M, diff
11701-13216:HW = PS1600, speed = 200 Mbps, smem = 64M, diff
```

圖 24. IOCHANNEL configuration

接著要進行的設定便是 Timing 與 Vector，圖 25.顯示了 Timing 與 Vector 的設定，與機台產生訊號給待測晶片之間的關係。Timing Setup 定義了每個 Test Cycle 的週期時間是多長，不同的 Vector 所對應的波形，以及每個波形中 Edge 的時間點，以圖 25.的例子可以看出來，我們定義了 16 種不同的波形，其 Physical Waveform Indices 為 0、1、2、3.....f，並且每個 Physical Waveform Indices 均可能代表相同或不同的波形，這 Physical Waveform Indices 即是在 Vector Setup 中所要設定的 state character。接著我們在 Vector Setup 中，指定了一個 pattern 順序，最後機台便會根據 Level Setup 的設定對應 Physical Waveform Indices 的波形，產生圖 25.下方的訊號給待測晶片，因此只要針對每個訊號進行類似的設定，便可產生完整的訊號給待測晶片。對於待測晶片的 response 的量測，我們也是採用相同的方法，不同在於指定的是 real response 與 expected response 所比對的時間點。

一般在進行測試時，訊號的波形是非常複雜同時測試向量的數量也都是非常龐大，因此必須另外利用其他的軟體工具來幫助我們進行轉換，在數位晶片量測流程上，使用者可利用 ASCII Interface 幫助進行轉換。要執行 ASCII Interface 必須先準備好三個資料，如圖 26.所示，其中 ASCII timing file (*.dvc)用來定義 ASCII vector file(*.avc)中每個 state character 所代表的波形與以及每個波形中 edge 的時間點，還有每個 test cycle 的週期時間是多長及 real response 與 expected response 比對的時間點。ASCII configuration file(*.aic)用來指定檔案放置的目錄，以及有那些 vector 要進行轉換，ASCII vector file 則就是 pattern 資訊。以上三個檔案的資訊結合後，機台便可由 ASCII vector file 得知每 cycle 要輸入哪些訊號進到 DUT 及預期 DUT 的輸出訊號為何，而這些訊號的長相就是由 ASCII timing file 所定義。以上轉換完成後會產生 timing setup file 與 binary vector file 二個檔案，

這兩個檔案就是機台所要讀取的資料。

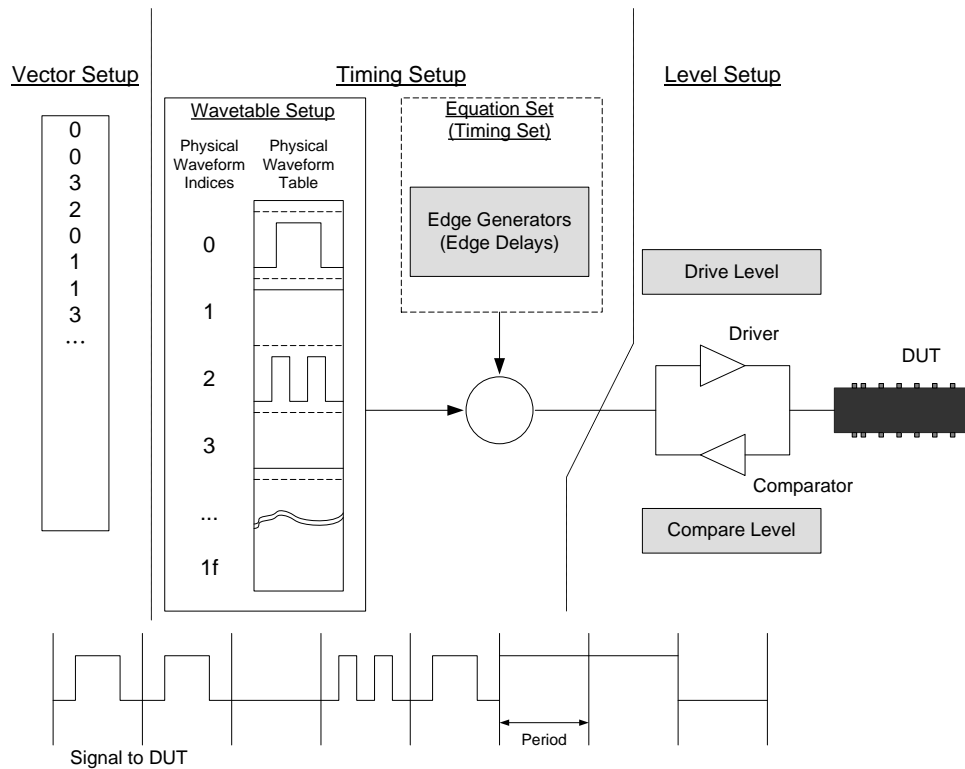


圖 25. Timing and Vector Setup

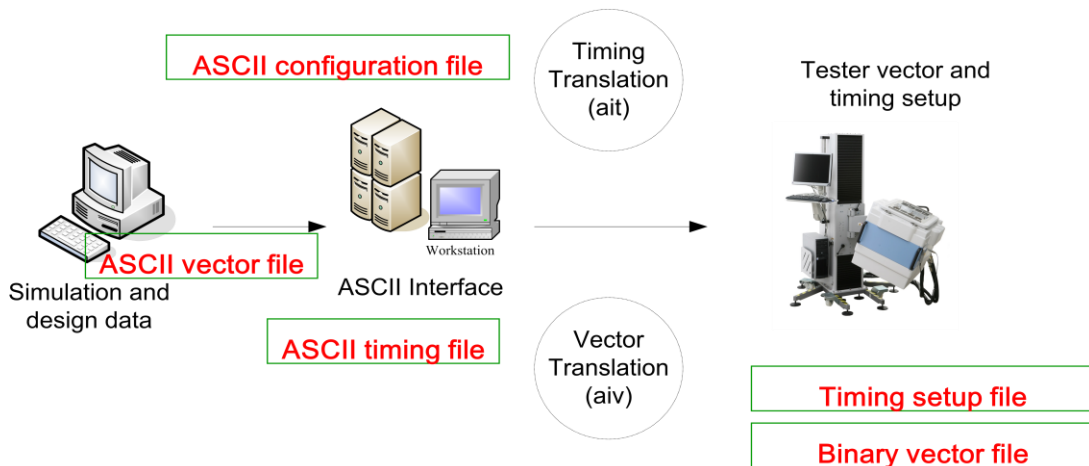


圖 26. ASCII Interface

對於 ASCII vector file 內容可由以下得知。圖 27.是一個 up down counter 的測試訊號示意圖，其中 q 的輸出是 8 個位元，圖中只有列出 2 個位元，由圖中可知，當 reset 信號為邏輯 0 時計數器會輸出 0，若 reset 信號為邏輯 1，則每次 clk 信號的 rise edge 發生時，updown 信號若為邏輯 1 則計數器會向上數，若 updown

信號為邏輯 0 則計數器會向下數，要將這個驗證電路的 pattern，從 simulation 的 testbench 轉換成 ASCII vector file，我們就要先定義出每個訊號在 test period 中所要呈現的波形，以圖 27.的例子來看，我們可以定義出如圖 28.這樣的一個結果，同時產生對應的 ASCII vector file。要將圖 27.的信號，轉換成 ASCII vector file 的表現方式，就必須針對每個信號，定義出在每個基本的 test cycle 中其波形的型式與其對應的 state character，以圖 27.的例子，我們定出了 clk 訊號的波形，以及對應的 state character “1”，而針對 reset 與 updown，我們定出了二種波形，與對應的 state character “0” 與 “1”，針對輸出 q，我們定出預期的訊號若是邏輯 0，則其 state character 為” L”，預期的訊號若是邏輯 1，則其 state character 為” H”。

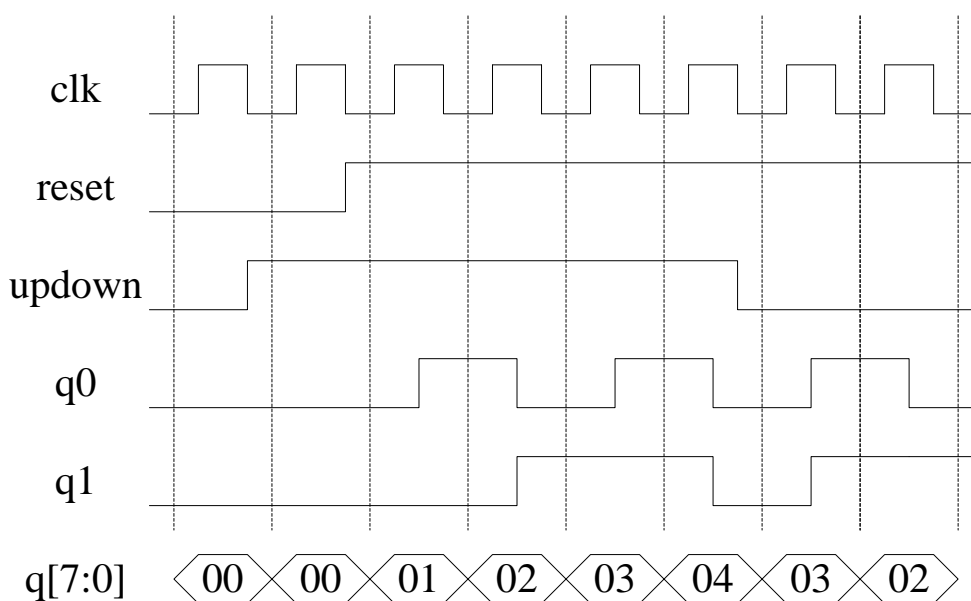


圖 27. Up/Down Counter Waveform

```

Vector.avc
FORMAT   clk,  reset,  updown,  q7,q6,q5,q4,q3,q2,q1,q0;

R1  std   100  LLLLLLLL;
R1  std   101  LLLLLLLL;
R1  std   111  LLLLLLLH;
R1  std   111  LLLLLLHL;
R1  std   111  LLLLLLHH;
R1  std   111  LLLLLLHL;
R1  std   110  LLLLLLHH;
R1  std   110  LLLLLLHL;

```

clock		1
reset/updown		1
reset/updown		0
q		H
q		L

input signal的位置
 輸入0 or 1...等Drive vector

Output signal的位置填入L or H or X...等compare vector

若是bi-direction signal，則在動作為input的cycle填上Drive vector；動作為output的cycle填上compare vector!!

 Ex: R1 std 00110100;
 R1 std 01001010;
 R1 std LLHLLLH;
 R1 std 11010100;
 R1 std HLHLHLH;

圖 28. ASCII Vector Setup File(*.avc)

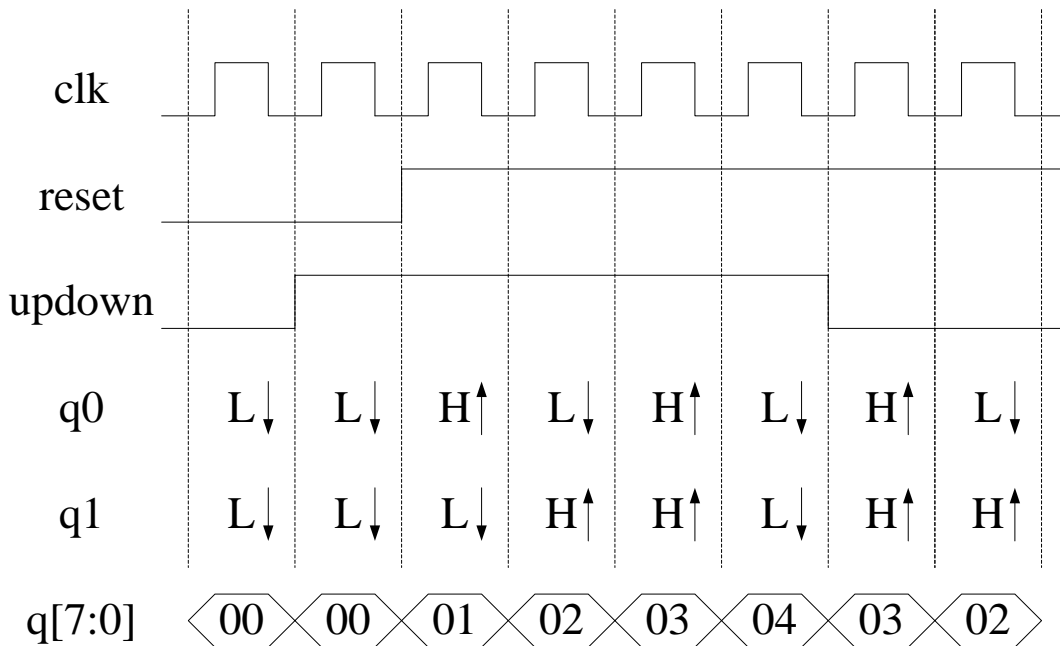


圖 29. The Input Stimulus and Output Response

透過所定義的波形，可以將圖 27.的測試訊號，利用 ASCII vector file 的內容來描述如圖 28.所示。在 ASCII vector file 中的第一行是列出每個訊號名稱，描述的方式先利用 FORMAT 這個關鍵字，後面加上每個訊號的名稱，每個訊號用逗號隔開，結束時加上分號。第二行開始便可以描述所要產生的測試訊號，開始的關鍵字 R1 指的是重覆這個 vector 一個 cycle，後面接上 std 是 device cycle name，這個 device cycle name 會在 ASCII timing file 中使用到，最後接上的就是每個訊號所要產生的波形，描述的方法就是透過 state character，一般使用上的習慣都是用” 0” 與” 1” 代表輸入邏輯 0 與邏輯 1，而使用” L”、” H”、與” X” 來表示輸出的預期結果是邏輯 0、邏輯 1 或 don’t care。每個訊號則透過所定義的波形與其 state character 連結在一起，描述完一個 vector 之後，結束時加上分號，接著可再下一行描述新的 vector。因此利用圖 28.中所列的 ASCII vector file 所產生的訊號就如圖 29.所示，ASCII vector file 中第一個 vector，R1 std 100LLLLLLLL 就會產生圖 29.的 cycle 1 這個波形，其中第 1 個 state character “1” 代表 clk 這個訊號，第二個 state character “0” 代表 reset 這個訊號，第三個 state character “0” 代表 updown 這個訊號，第 4 個 state character “L” 代表輸出 q7，之後的以此類推。第二個 vector，R1 std 101LLLLLLLL 就會產生 cycle 2 的波形，第三個之後的 vector 也是同樣的方式產生。另外，當我們描述 ASCII vector file 的時候，對於訊號的名稱亦可以使用 Pin Group 名稱；當然前提是必須先將所要描述的訊號定義成 Pin Group。

所有的輸出訊號部份，機台在測試時便會自動將待測晶片的 response，於所

定義的比較時間點，與 Level Setup 中所設定的 VOL 與 VOH 的電壓值進行比較，來判斷是否與預期的邏輯值相同，因此 Vector 內對於輸出訊號的設定就是 expected pattern。

ASCII vector file 的產生方式有二種，若是波形很簡單，可以使用文字編輯器來產生；若是波形複雜，一般會在進行電路模擬的同時，讓電路模擬軟體來產生，以 Verilog 語法來說，通常可使用 \$fdisplay 與 \$fwrite 等 system task 來產生。

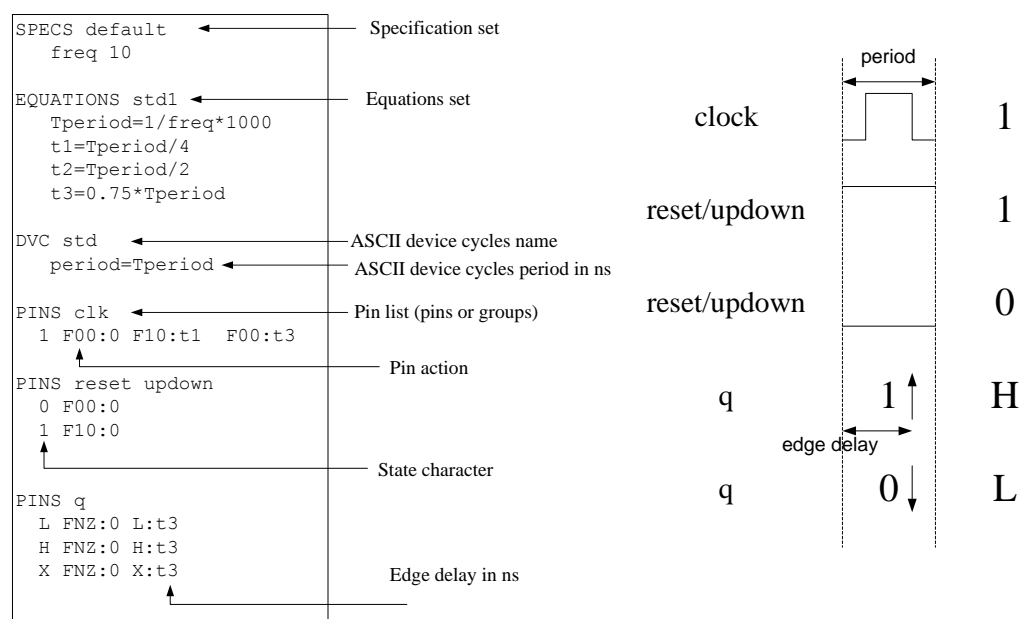


圖 30. ASCII Timing Setup File(*.dvc)

ASCII timing file 則是用來定義每個訊號，其 state character 所對應的波形，以及每個波形中的 edge delay，還有 test cycle 的週期長度，圖 30.便是將所定義的波形，利用 ASCII timing file 來描述。一開始的 SPECS 是關鍵字，可以用來指定 specification set，也就是定義一些變數，以這個例子，我們定義了一個變數 freq 代表信號的頻率，一開始設定其值為 10MHz。第二個部份 EQUATIONS，用來指定一些運算式，以便得到所需的數值，範例中運算了從 10MHz 的頻率，轉換成 100ns 的週期，以及 t1、t2 和 t3 三個時間點，在描述 edge delay 時會使用，這些時間點可視需求自行增減定義。第三個部份是定義 ASCII device cycle name 與其週期，使用 DVC 這個關鍵字，後面加上在 ASCII vector file 中的 device cycle name，由於我們使用了 freq 的變數，並且使用 equation 來運算得到週期時間 Tperiod，因此將 Tperiod 變數指定給 period 關鍵字來設定 device cycle period，這個 period 指的就是測試時的單位時間週期。第四個部份便是描述每個訊號的波形外觀，以及每個 edge 的 delay 時間是多少，首先用 PINS 這個關鍵字來定義訊號，在 PINS 後面加上所要定義的訊號名稱，如 PINS clk 代表要定義 clk 這個信號，由於我們所定義的 clk 這個信號，其 state character 只有” 1” 這種，其波形如圖 30.的右方所示，定義的方式為 1 F00:0 F10:t1 F00:t3，一開始的” 1” 代表

的是 ASCII vector file 中， clk 這個信號的 state character “1”，後面的 F00:0 是用來描述 pin action，代表的是在每個 test cycle 的起點開始算 0ns 時要 force 一個邏輯 0，F10:t1 指的是要在每個 test cycle 的起點開始算 t1=25ns 時要 force 一個邏輯 1，因此 clk 訊號的 rising edge 與 test cycle 一開始的原點，相差了 25ns，F00:t3 代表在 t3=75ns 的時候，要 force 一個邏輯 0，因此 clk 訊號的 falling edge 與 test cycle 一開始的原點相差了 75ns 的時間，因此 clk 的波形就如圖 30.所示的 clock 訊號波形。reset 與 updown 這二個訊號其 state character 有” 0” 與” 1” 二種，其波形分別是整個 test cycle 都是邏輯 0 與邏輯 1，在 PINS 後面可以把相同設定的訊號一起寫入，然後分別定義” 0” 與” 1” 二種 state character 其波形。輸出的部份，q7 至 q0 因為在 pin configuration 有設定一個群組 q，所以在 PINS 後面可以指定群組名稱，接著指定每個 state character 與對應的動作，FNZ:0 L:t3 其意義為在 test cycle 一開始先開啟 tri-state，而在 t3=75ns 時，進行比較，看看待測晶片所產生的 response 是否為邏輯 0，也就是在 t3 這個時間點，待測晶片所產生的 response，其電壓是否有比 VOL 的電壓還要低，至於 state character “H” 也是相同的比較方式，只是 H:t3 代表預期的邏輯值為 1，也就是電壓值要比 VOH 的電壓還高，而 state character “X” 所定義的 X:t3 代表 mask 也就是不用比較待測晶片所產生的 response。

圖 31.列出了所有可供 drive action 的 pin action 選項，也就是在描述每個訊號波形時，可以使用的動作，使用 F<D><T>是正規的方式，有的也可以用簡寫，如 F10 可以簡寫為 1，例如寫 F10 就是代表 drive High 及 tri-state off。圖 32.列出了所有可供 compare action 的 pin action 選項，比較時可分成 edge compare 與 window compare，edge compare 指的是在特定的時間點去比較 response，而 window compare 指的是持續一段時間的比較，因此使用 window compare 時必須在結束的地方加上 close window 的描述。當 vector 寫到了 compare action 時，SmarTest 便會針對 real response 做比對的動作，而 drive action 則會做輸入訊號；因此 bidirectional pins 的 vector 會同時出現這兩種 action，對於 bidirectional pins 的設定可參考圖 28.的說明。

List of Drive Actions

N	or	.(hold)
F0N		
F1N		
FN0	or	!Z
FNZ	or	Z
F00	or	0
F0Z		
F10	or	1
F1Z		

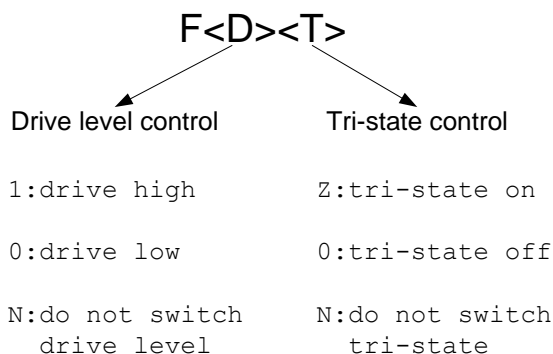


圖 31. Drive Actions in Pin Action

List of Edge Compare Actions

N	or	.(hold)
L		compare to low
H		compare to high
M		compare to intermediate
X		don' t care (mask)

List of Window Compare Actions

WL	compare to low
WH	compare to high
WM	compare to intermediate
WX	don' t care (mask)
WU	compare to unstable
WC	close Window

圖 32. Compare Actions in Pin Action

ASCII configuration file 是用來指定進行 ASCII interface 時，來源的 ASCII vector file 有那些，以及 ASCII vector file 所存放的目錄，還有轉換完成的 Time 與 Vector 設定檔存放目錄位置，圖 33.列出了CIC所設定的 ASCII configuration file，一般要修改的部份有 pinconfig_file 這行，改成 Pin Configuration 的設定檔存放位置，avc_dir 說明 Vector files 就放置在這資料夾下，PATTERNS 下方則列出所有要進行轉換的 ASCII vector file 的主檔名(如 updown.avc，就列出 updown)，type 的欄位是用來決定 pattern 是 main pattern 還是一個 subroutine，v2b_options 用來指定此測試機台為 sequence memory mode，ctim 的欄位是用來決定是否要改變 timing 的設定，xfact 是用來決定 X-mode，也就是在一個 test cycle 中，是否要放入 2 個以上的 vector，ascii_dvc 欄位是用來定義 ASCII timing file 中的 device cycle name，而 vec_ascii_dvc 欄位是用來定義 ASCII vector file 中的 device cycle name。

對於 xfact 的設定部分，由於本測試機台所使用的操作頻率是以倍頻方式產生，而 xfact 共可設定 1~4 等四種倍頻。以 1.6Gbps 來說，若以 xfact=1 來轉檔的話，則此 Device 的 Timing 只可定義最高 400Mbps (1600/4)；若設定 xfact=2 則是可達 800Mbps，故最高速為 xfact=4 可達 1.6Gbps 最高操作頻率。

```
AI_DIR_FILE
tmp_dir    ./tmp
tmf_dir    ./timing_mapping_files
vbc_dir    ./
avc_dir    ./ascii_vectors/
allvec_file ./all_vectors/all
pinconfig_file  /user93k/hlchen/workorder/configuration/pins
single_binary_pattern_dir ./single_vectors/

PATTERNS  name  v2b_options  type  ctim  xfact  { vec_ascii_dvc  ascii_dvc };
pattern1  -k      MAIN    NCT    1      { std  std };
pattern2  -k      MAIN    NCT    1      { std  std };
pattern3  -k      MAIN    NCT    1      { std  std };
```

圖 33. ASCII Configuration Setup File (*.aic)

準備好修改完成的 ASCII configuration file (sample.aic)、ASCII timing file (sample.dvc)與 ASCII vector file (updown.avc)之後，利用

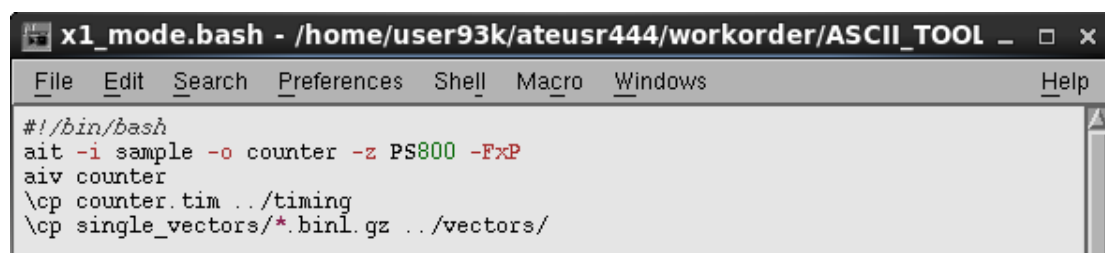
```
ait -i sample -o timing_setup_file -z PS800 -FxP
```

來進行 timing 設定檔的轉換，其中-i 之後接的是 ASCII configuration file 及 ASCII timing file 設定檔的檔名，-o 之後寫上欲輸出檔案名稱，-z 後接測試機台模組名稱，而 FxP 則是針對 sequence memory 架構所下的宣告。轉換完成之後會產生 timing_setup_file.tim 這個檔案，將它拷貝至 device 目錄下的 timing 目錄，則在 data manager 中的 timing icon 即可以讀取。再使用

```
aiv timing_setup_file
```

這個指令來轉換 vector，其指令是在 aiv 後接 ait 指令上所宣告的輸出檔案名稱即可。轉換成功之後，會產生 updown.binl.gz 這個壓縮檔（本名稱會依使用者所使用的 pattern 名稱而異），將它拷貝至 device 目錄下的 vectors 目錄，則在 Eclipse Workcenter 中的 Vectors icon 即可以讀取。檔案讀取的方式為先點選 Eclipse Workcenter 中的 Timing icon，然後右鍵選擇 Load 就會出現 Timing setup file 的選擇視窗，選擇所要的檔案後，點選 Load 按鈕，Vector 的讀取方式類似，如此就可讀取轉換成功的檔案。

而以上的 ASCII Interface 相關檔案，CIC 都已準備好 template file 於『/home/user93k/ateusr/ate_env_file/ASCII_TOOLS』資料夾，使用者只需把資料夾複製回 Device 下即可使用。關於 ait 及 aiv 等轉換程式及複製等動作，亦可執行此 ASCII_TOOLS 資料夾下的./x1_mode.bash 完成，關於./x1_mode.bash 檔案內容如下圖 34 所示。



```
x1_mode.bash - /home/user93k/ateusr444/workorder/ASCII_TOOL
File Edit Search Preferences Shell Macro Windows Help
#!/bin/bash
ait -i sample -o counter -z PS800 -FxP
aiv counter
\cp counter.tim ../timing
\cp single_vectors/*.binl.gz ../vectors/
```

圖 34. ASCII Interface transfer bash

六、SmarTest Testflow 簡介

當以上步驟完成後，基本上就完成了 SmarTest 基礎設定操作，接下來要進行的步驟是將待測項目整合成醫晶片測試流程，此測試流程名為 Testflow。其內容包含測試項目（例如：訊號連線測試、供電測試、功能測試等諸多測試項目）、每測試項目所要使用的 Level 及 Timing 的 Equation 及 Spec、待測 Patterns、測試結果顯示、迴圈、計數及分支等多項功能。一簡單的 Testflow 完成圖如圖 35.所示。由於其功能繁複，因此將另外撰文討論其產生方式。

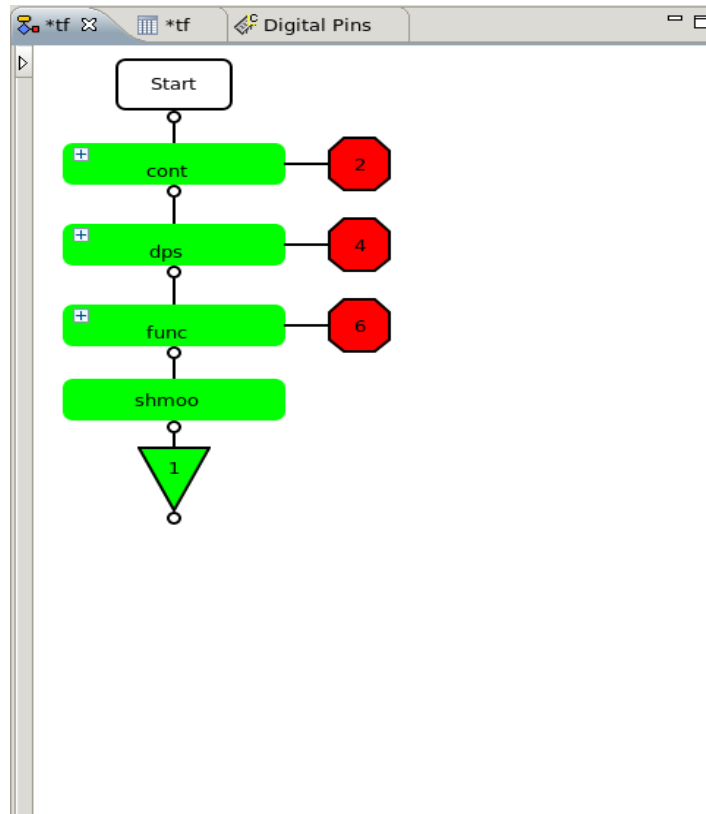


圖 35. A Simple Testflow

七、結論

本文簡介了 CIC 新引進的 Advantest V93000 PS1600 硬體規格，並介紹了 SmarTest 軟體基礎操作步驟（本文內容適用於 SmarTest 7 之後的版本）。簡單來說，Pin Configuration 設定待測晶片與測試機台的連接方式，Level 設定待測晶片的供電電壓、輸入及輸出訊號準位，Timing 則是設定輸入訊號及輸出訊號的波形變化，Vector 則是設定輸入 stimulus 及預期的輸出 pattern。希望讀者可藉由本文對 CIC 的混合訊號自動測試機台有初步的認識。

八、參考資料

- [1] Agilent Technologies, "Agilent 93000 SOC Series User Training Part 1" Dec. 2002.
- [2] Agilent Technologies, "Agilent 93000 SOC Series User Training Part 1" Oct. 2004.
- [3] Agilent Technologies, "Agilent 93000 System Training ASCII Interface Training" May 2001.

前言

本中心 Advantest V93000 PS1600 自動測試機台提供強大的測試功能。使用者使用本機台時，需搭配一測試程式，方可使用測試程式來控制機台，並進而達到晶片量測的目的。前文『Advantest V93000 PS1600 基礎操作簡介』中，已敘述了由 SmarTest 開啟到四個基礎設定完成，接下來便要進入測試程式開發的階段。

一、Testflow 簡介

欲使用 SmarTest 進行晶片測試前，需先建立 testflow。其 testflow 顧名思義即是測試流程，內容包含加入測試項目，測試流程控制，流程參數設定……等多個項目。以下將說明如何在 SmarTest 上建立 testflow，而此範例 testflow 為參考本中心近年來對於學術界量測數位晶片之流程所建立，因此本範例 testflow 將適用於絕大多數研究用的學術界所設計之數位晶片。

二、測試項目說明

一般於本中心量測數位晶片時，通常會量測以下幾個測試項目：

「continuity」、「standby current」、「functional test」、「shmoo」、「operation current」等測試項目，完成之後的操作介面如圖 1.所示。以下將針對各測試項目作捍簡單的介紹。

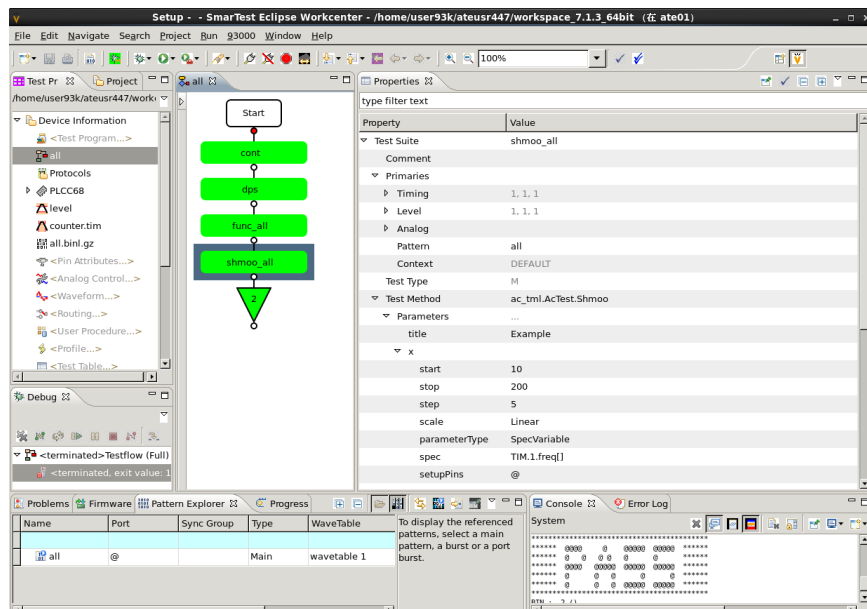


圖 1. SmarTest 操作介面與 testflow

A. Continuity

首先談到「continuity」。這個測試項主要用來量測 ATE 與晶片之間的連線檢查，當我們在進行 pin configuration 設定時，會對晶片的每個 pad 進行 test channel 設定，代表晶片的某一 pad 將連接到 ATE 的某一 test channel。量測時，ATE 將在每個 DUT 的 channel 上輸入或抽出一測試電流，而由於 DUT 的每一 pad 上均內建有一組 ESD 保護 diode（此 diode 有可能連接 signal-power 或 signal-ground 或 power-signal-ground 等三種可能路徑），因此測試電流將在 signal-power 或 ground 路徑上通過此 ESD diode 並產生約 200mV~700mV 之間的壓降，而 ATE 只要在這個 pad 的相對 test channel 上利用一 voltage measurer 檢查是否有此一壓降存在便可知道 ATE 與 DUT 的 pad 是否有異常連接，如圖 2.所示。

由於此測試檢查點實為利用 pad 內建的 ESD diode 進行量測，因此若待測 DUT 無內建 ESD diode 的話，則此測試項將無法正常運作。另外，晶片內部的連線問題也無法使用此測試項得到解答，例如：若 pad 本身無法將訊號由 pad 外傳到晶片內部的問題、或 pad 到 core 的繞線有 open 或 short 等現象。

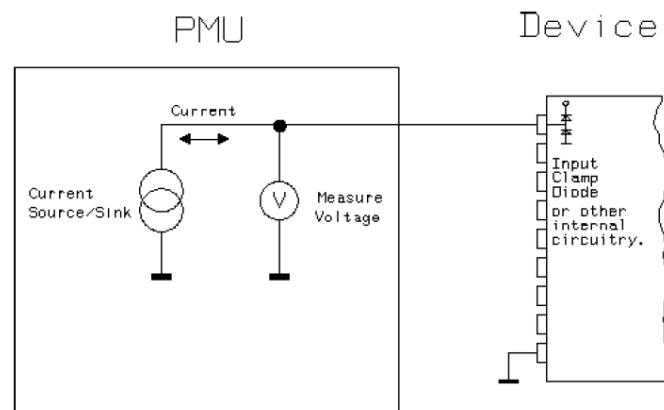


圖 2. continuity 測試項

B. Standby current

接著談到「standby current」，本測試項目在量測時將對 DUT 的 power/ground pad 連上 ATE DPS module channel 的 VDD 及 GND，並且在不輸入測試訊號的情況下，如圖 3.所示。因此此時 DUT 應是處在 standby 狀態下，故理論上在此狀態下的 DUT 將僅僅只會消耗極小的靜態功率（static power），因此若由 DPS module channel 量測消耗電流時，應該僅僅只會量測到非常微小的 standby current，若量測所得的此 standby current 太大則表示 DUT 有異常情況。例如：若 power 與 ground 短路的話，則會造成非常大的短路電流消耗。

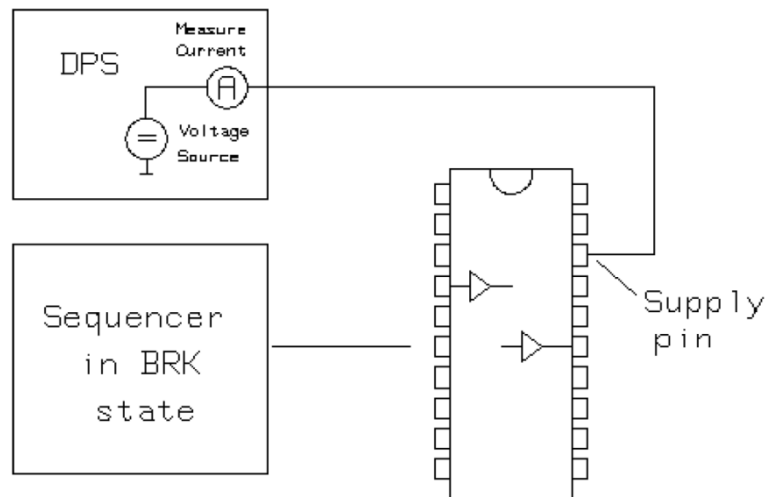


圖 3. standby current 測試項

C. Functional test

由以上兩個測試項目可知，對任一 DUT 而言，「continuity」可檢查 DUT signal pad 與 ATE test channels 的連接；「standby current」可檢查 DUT power/ground 與 ATE DPS channels 的連接。因此當以上兩個測試項目都通過後，就表示可以開始進行功能性測試了。這邊就會用到「functional test」這個測試項目。在 ATE 與 DUT 的 power/ground 正常動作的環境下，此測試項目將使用 Level Setup 及 Timing Setup 的內容，再搭配 Vector Setup 的資料產生 driver data 後，送入 DUT 內，並將由 DUT 輸出的訊號 real response data 與事先於 Vector Setup 所設定好的 expected data 進行 cycle-based 比對，若所有 cycle 的 pattern 都相符的話，表示功能測試通過；若有任一 pattern 為不相同的話，表示測試失敗，流程如圖 4.所示。

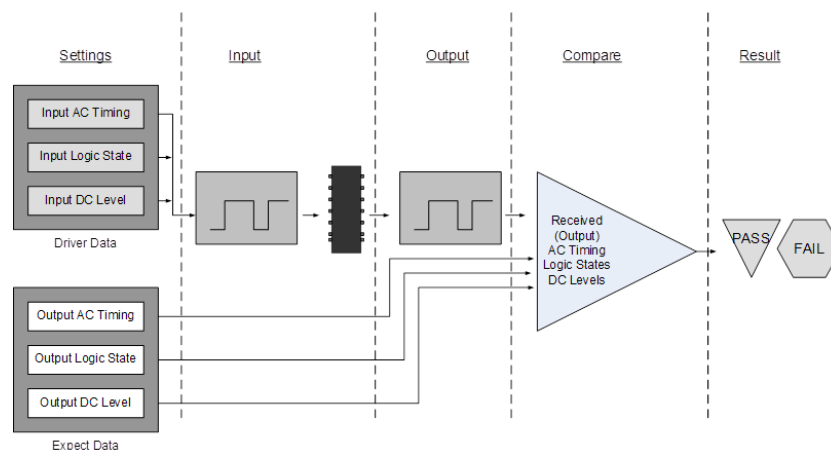


圖 4. functional test 測試項

D. Shmoo

當功能性測試通過後，接著就可以進行 shmoo 的繪製。「shmoo」繪圖

為使用任意兩個 test environment parameter 的組合工作區間內進行功能性測試，一般較常使用 supply voltage 搭配 frequency，因此在 shmoo plot 結果上就可以看出低電壓+低速、低電壓+高速、高電壓+低速及高電壓+高速等多種組合下的功能性測試是否正常，以及電壓及操作速度改變時的晶片的功能是否有任何問題。在 SmarTest GUI shmoo plot 上，綠色結果代表此 condition 下的功能性測試 pass，而紅色則代表 fail。下圖 5.為 shmoo plot 說明示意圖，shmoo plot 進行測試時，會先將 x 軸方向的 parameter 進行一 step-width 數值改變；其方向為由 start 到 stop，待 x 軸方向的數值改變結束後，接著進行 y 軸方向的 parameter 數值由 start 到 stop 進行 step-width 變化，每變化一 step 的 y 軸變化，就要再進行一次完整 x 軸的變化，直到 x 軸與 y 軸的變化都達到 stop 點始完成一次的 shmoo plot 操作。

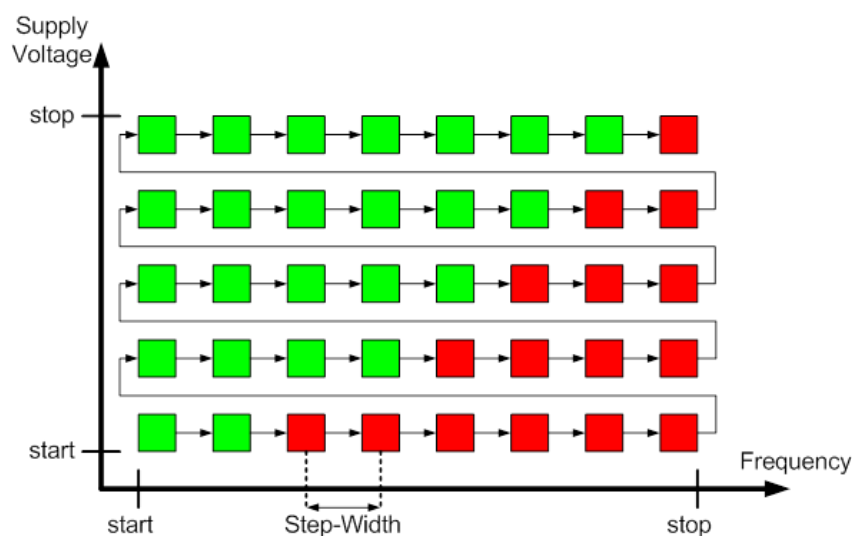


圖 5. shmoo plot 測試項

E. Operation current

最後是「operation current」測試項，此測試項就是 ATE 會在輸入 supply voltage 且進行功能性測試時，一併進行消耗電流量測，如圖 6.所示。由於 operation current 所量測的數值實為平均電流，因此需要讓 functional test 持續進行才可以量到穩定的平均電流，因此此步驟需要將 stimulus patterns 進行 loop 模式，在 loop 模式下 DUT 的 stimulus patterns 將可以持續且重複的輸入到 DUT。

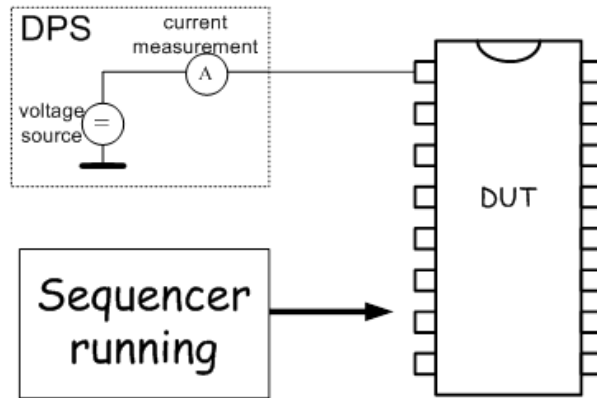


圖 6. operation current 測試項

以上五個是學術界師生使用本中心 Advantest V93000 PS1600 進行晶片測試時最常使用到的測試項目，以下段落我們將說明如何導入測試項目於 SmarTest testflow 中。

三、測試項目及 testflow

首先我們先由 SmarTest GUI 左手邊的 testflow icon 按下右鍵，選擇 [New]，接著在 New Testflow 視窗中自訂輸入此 testflow 的名字，最後按下 [Finish] 即可建立一 testflow 設定，其過程如下圖 7.所示。



圖 7. Start Up The Testflow

接著連點兩下 testflow icon 後就可開啟此 testflow 設定頁面，在頁面的 [start] 按鈕下的圓圈點按下滑鼠右鍵，選擇 [Insert] → [Run Test]，如圖 8. 所示。這邊可以選擇 [Run Test]、[Run and Branch]、[Good Bin] 等多種測試選擇及控制選擇。其中 [Run Test] 指的是測試項僅有測試功能，無法做任何分支操作；而 [Run and Branch] 則是代表當測試成功時可以繼續向下執行其他測試項目，但若測試失敗時則可以進行分支的測試項目或加入一些控制選項。由於一般學術界的數位晶片測試過程都不會太過複雜，所以其實使用 [Run Test] 就足夠了。

按下插入[Run Test]或[Run and Branch]之後，便會跳出如圖 9.的 Node Wizard。每一個 Node Wizard 所建立的就是一個測試項目，也就是一個 test suite。接著就是輸入此 Test Suite Name 後，再按下 Test Method 最右邊的選擇按鈕，如圖 10.所示。在此按鈕中，我們可以選擇任一個 test method 來當作此 test suite 的測試項目，首先展開 dc_tml 這個 test method group 後選擇 DcTest 下面的 continuity 測試項，然後按下[OK]後就完成了 test method 選擇。

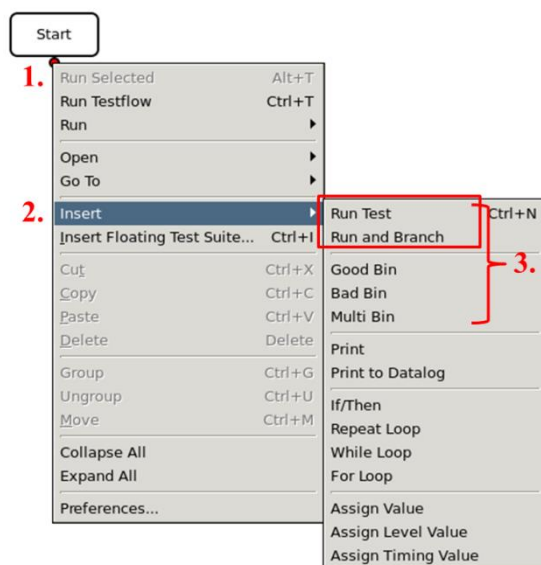


圖 8. Insert The Test Suite

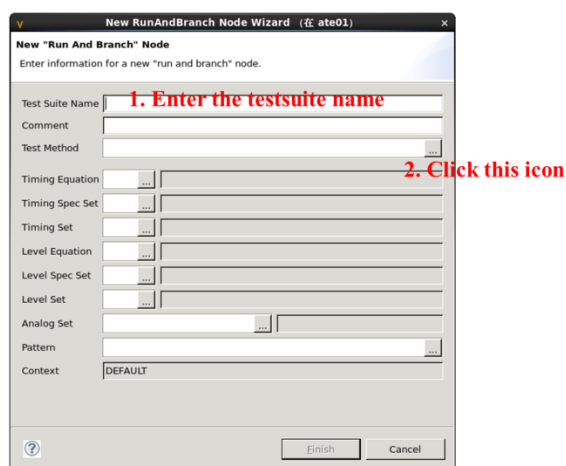


圖 9. RunAndBranch/RunTest Node Wizard

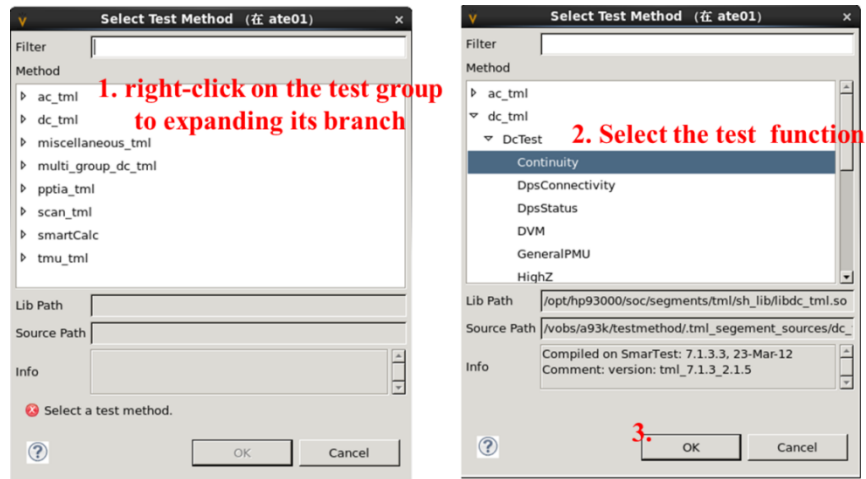


圖 10. Select Test Method

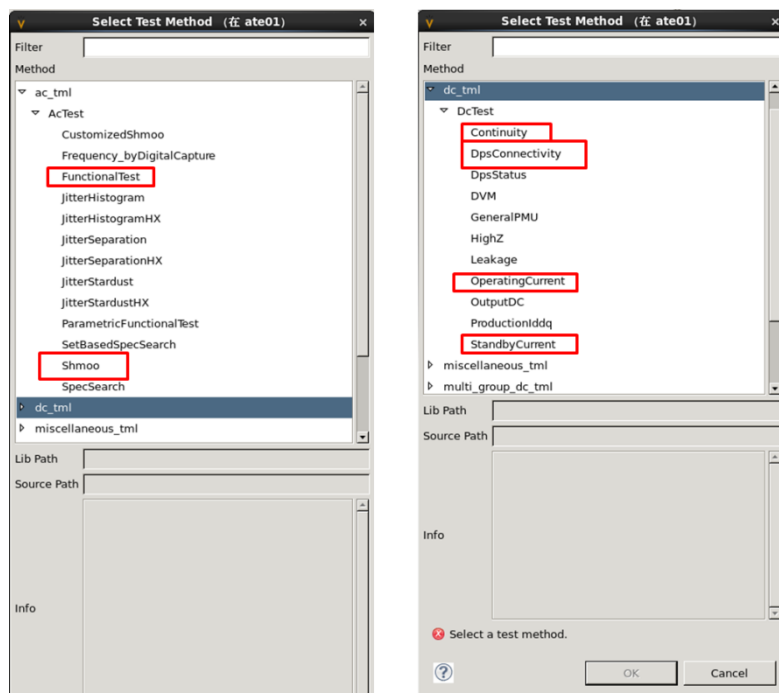


圖 11. Test Method Groups

而這些 test method group 下各有一些測試項如圖 11.所示，其中與功能測試無關的都在 dc_tml 下的 DcTest 中，如「continuity」、「OperationCurrent」及「StandbyCurrent」等測試項。而與功能測試相關的則屬於 ac_tml 下的 AcTest 中，如「FunctionalTest」及「Shmoo」等測試項。

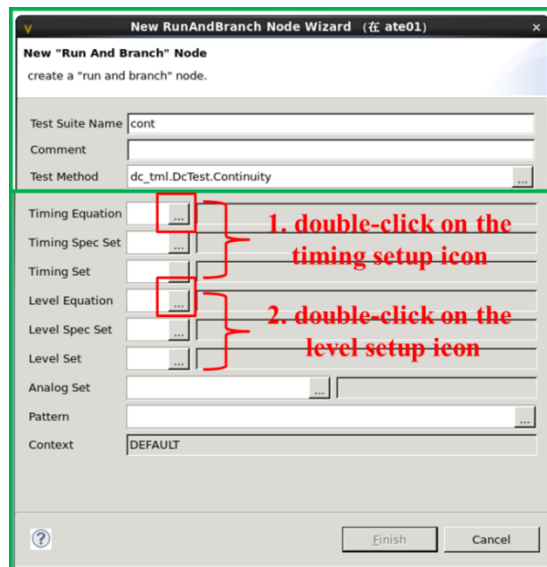


圖 12. Timing And Level Setup

接著再 Node Wizard 下應該可以看到 Test Method 欄位已經填入的「continuity」測試項的名稱。接下來就是選擇 Timing 及 Level Setup，首先在 Timing 相關欄位上按下右邊的選擇按鈕叫出[Select Timing]視窗(如圖 12. 所示)，然後選擇此測試項要使用的 Timing Specification 及 Timing Set，之後按下[OK]；Level setup 也適用同一方法選擇，如圖 13.所示。接著就可以在 Node Wizard 視窗中看到 Timing 及 Level Setup 相關欄位都已經選擇完成，如果此 test suite 有需要 pattern 輸入的話，則可以在 Pattern 欄位中選擇待測 pattern，如果不需要的話就直接按下[OK]，就可以看見 testflow 視窗中多出一個 test suite。如圖 14.為使用[Run and Branch]所完成的 test suite 範例。

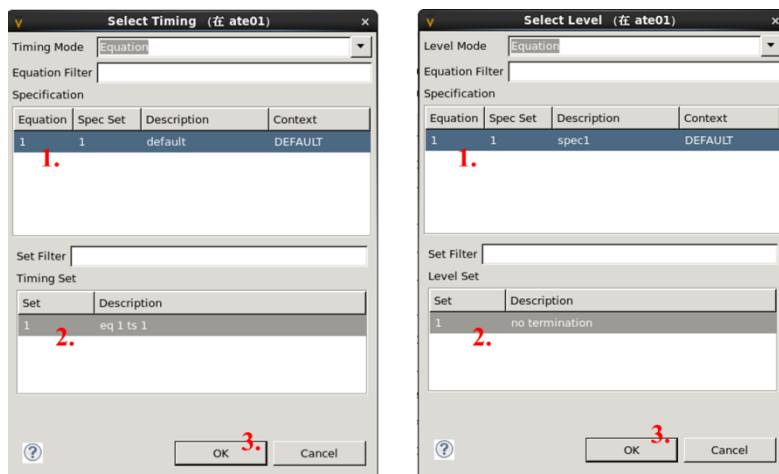


圖 13. Select Timing And Level Spec

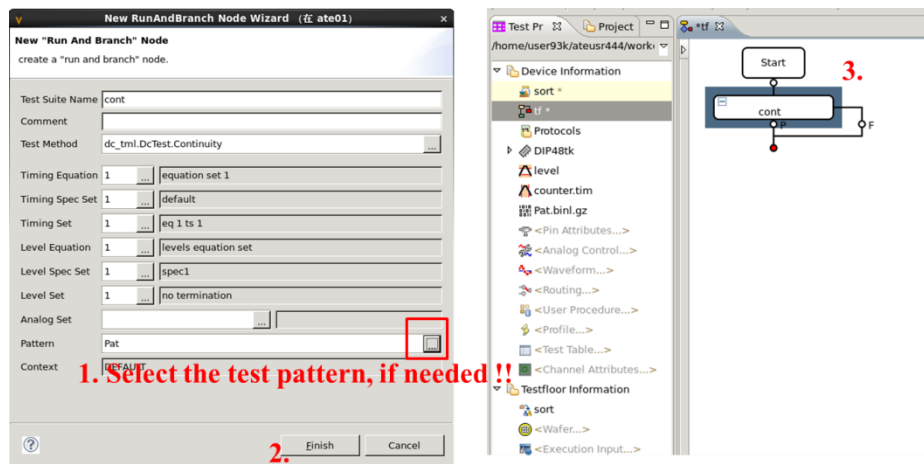


圖 14. A Simple Test Suite Example

某些 test suite 在量測時，為因應各 DUT 設計或製程上差異，我們還需要設定一些額外的參數。在「continuity」test suite 建立後，在「continuity」icon 上連點兩次就可以帶出此 test suite 的 Properties，為因應採用 CIC Cell-based design flow 所採用的 pad 設計方式，因此需要修改 test current 及 Limit Value。首先在 testCurrent 下輸入 -10[uA]，這是由於製程上的 pad 內建的 ESD 方向為連接 Signal-Ground 所致；接著展開 Limits→passVolt_mV→Test Number→Limit Value 輸入數值為 200 到 700 之間，如圖 15.所示。如此才算真正完成了「continuity」test suite 設定。

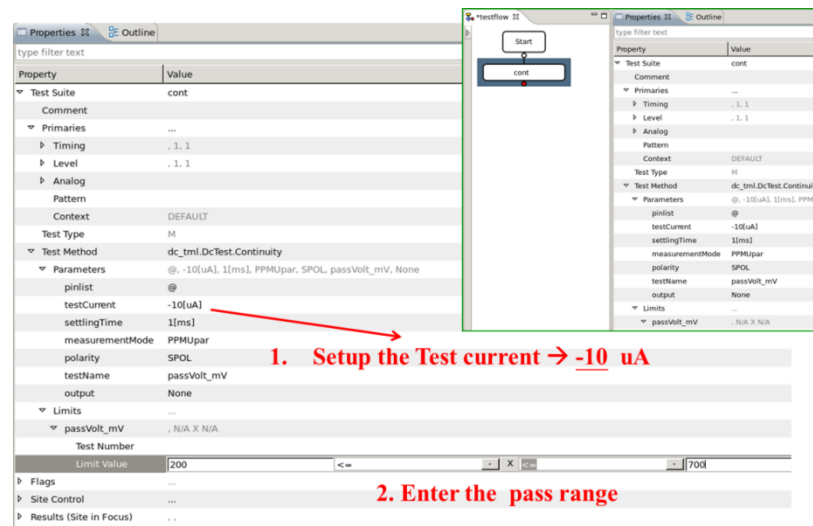


圖 15. Testsuite Properties – Continuity

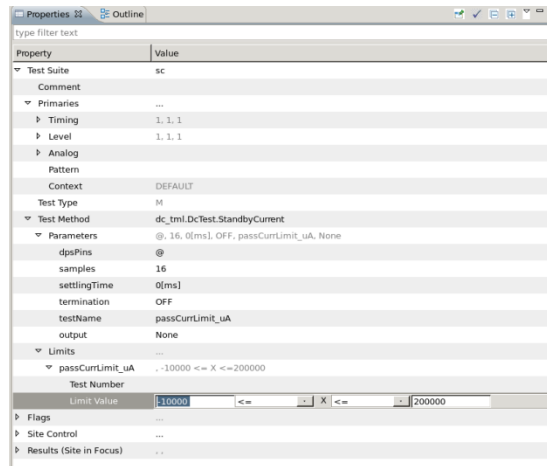
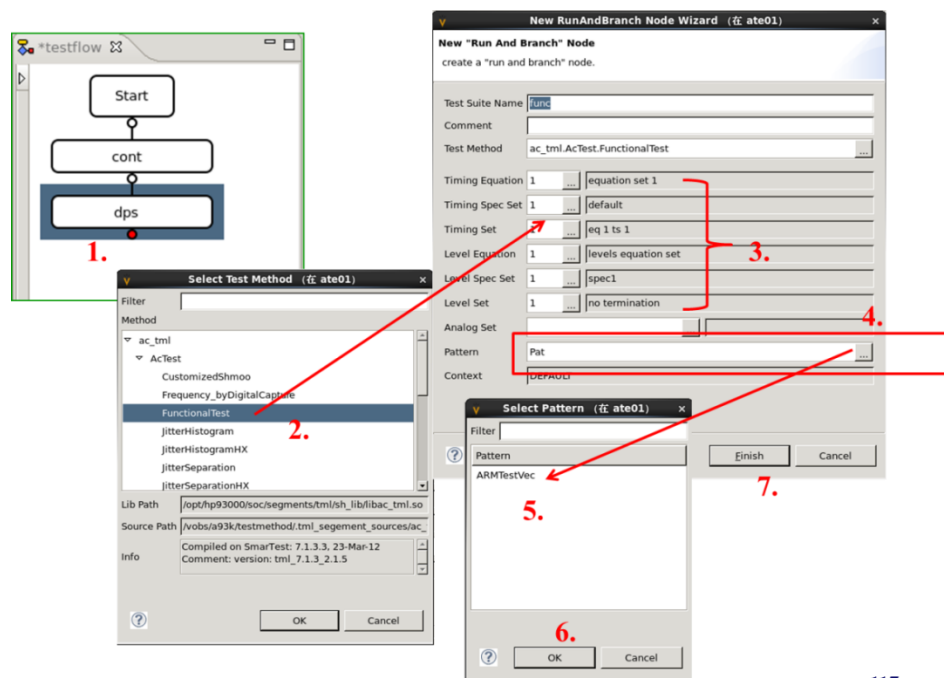


圖 16. Pass Current Current Range at Standby Current

接下來需要建立「StandbyCurrent」test suite，如圖 8.及圖 9.相同的方式插入 test suite 後，我們可以使用 dc_tml→DcTest→StandbyCurrent 找到並選擇 standby current 測試項(如圖 11.所示)，接著仍參考圖 12.~圖 14.的步驟完成 test suite 的導入後。接著連點兩下此 test suite 展開 test suite properties，我們必須修改其 passCurrLimit_uA 的 Limit Value，可參考如圖 16.所示將 Limit Value 的範圍設定成 $\text{minValue} \leq X \leq \text{maxValue}$ 。這邊的 minValue 及 maxValue 與 Level Equation 設定有關，我們在 Level Equation 設定時曾設定了 ilimit 參數，這參數用於限定 DPS module 的 current limit，也就是限定 DPS module 可供給 DUT 的最大電流值，而當 DUT 內有任何短路路徑存在時，其 standby current 的值將會非常大，但最大不可能大於 ilimit 設定值，而由於當 standby current 的值若不在 passCurrLimit_uA 的 Limit Value 區間內才會產生 fail 的結果，因此通常我們會設定 maxValue 值為稍小於 ilimit 設定值；而 minValue 值為一比較小的負數值，其目的為避免過小的 standby current 量測結果由於準確度影響而產生負數值進而使量測出現 fail 的結果。設定方式可參考圖 16.所示。

接著要導入的測試項是「Functional Test」測試項，一樣如圖 8.~圖 14.的方式建立 test suite，但其中有的差別在於「Functional Test」需要使用 pattern 來進行測試，因此在 Node Wizard 設定中，必須在 Pattern 欄位上選擇一 pattern 進行測試，可參考如圖 17.所示流程。此 Pattern 即為我們在 Vector Setup 上所設定的 vector pattern。因此若要使用不同 pattern 進行量測前，必須先將 Vector Setup 內容修改後，方可在此 test suite 內呼叫新的 pattern 以進行量測。



117

圖 17. Insert Functional Test

一般當「Functional Test」測試完成後，已可確定此 DUT 的簡單功能性測試幾乎是通過了，接著需要對 DUT 於不同環境下進行測試，這時就要使用「shmoo」測試項。Shmoo plot 一樣需要 pattern 來進行，所以一樣參考如圖 8.~圖 14.及圖 18.方式建立 test suite 後，接著連點兩下 shmoo plot 的 test suite 叫出其 properties，這邊要設定此 shmoo plot 的 x 軸參數及 y 軸參數；一般建議 x 軸設定操作頻率，而 y 軸設定供應電壓，所以 shmoo plot 繪圖時將會先在某一供應電壓下將各操作頻率跑過一次後再累進供應電壓。Properties 設定上可參考圖 19 設定 x 軸參數，其中 spec 的選擇對應到 Level 或 Timing setup 時的 Spec 參數，這邊先選擇 TIMING 下的 freq 這個參數，而 start 表示由哪個頻率開始執行，stop 則是最後執行的頻率，step 表示由 start 跑到 stop 過程中的累進頻率。另外可參考圖 20.設定 y 軸參數，spec 可以選擇 LEVEL 下的 corevdd，start 跟 stop 則是填入其 DUT 製程的 core vdd 供應電壓，一般保守選擇為標準供應電壓的正負 10%設定為 start 及 stop，step則是由 start 到 stop 之間的累進電壓。設定完成的結果可參考圖 21.所示。由於機台軟體的限制，所以 shmoo plot 繪製時，只可允許在 start 到 stop 之間切出 100 個區間，因此使用者必須謹慎選擇其 step 累進值。

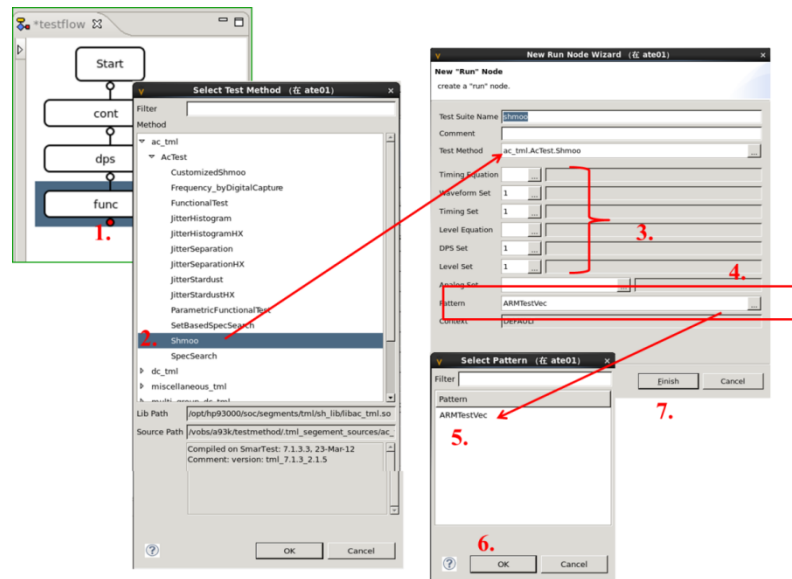


圖 18. Insert Shmoo Test

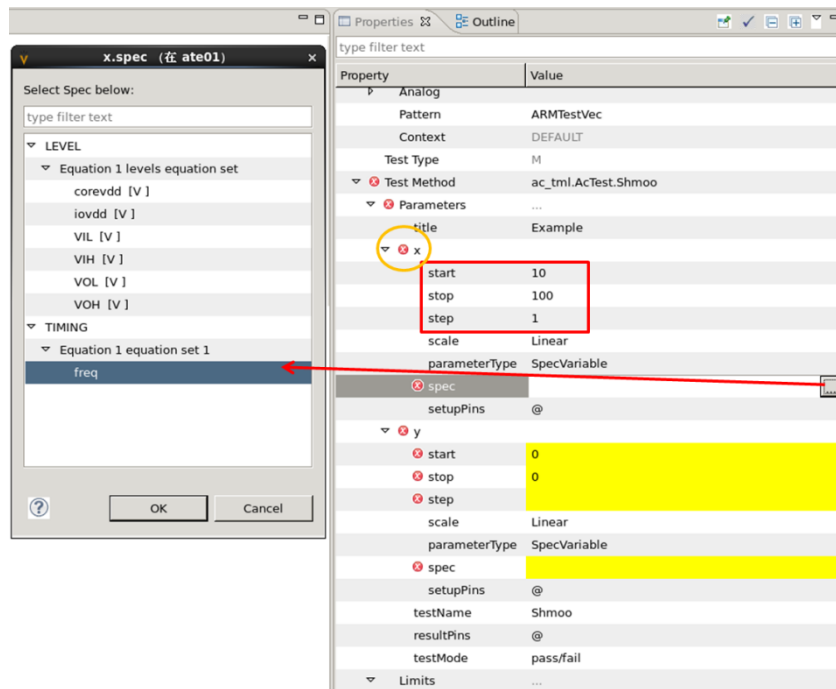


圖 19. Shmoo Testsuite Properties in x Spec

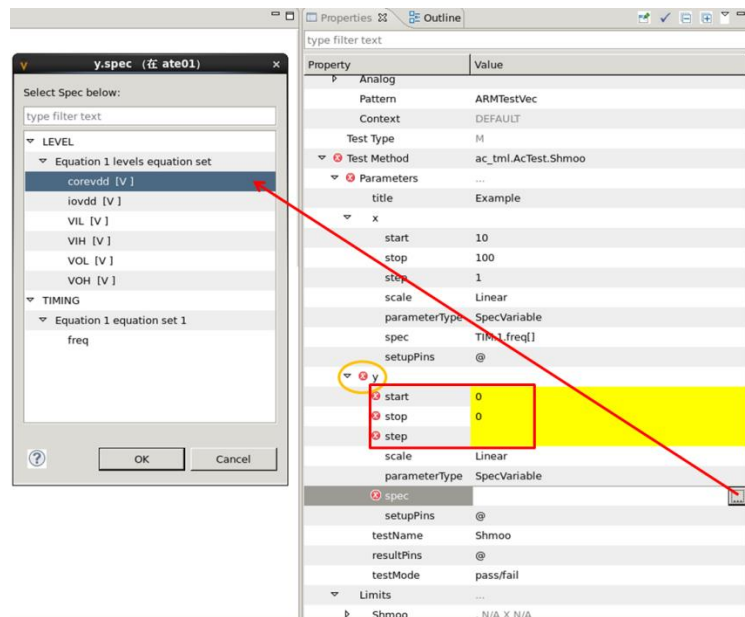


圖 20. Shmoo Testsuite Properties in y Spec

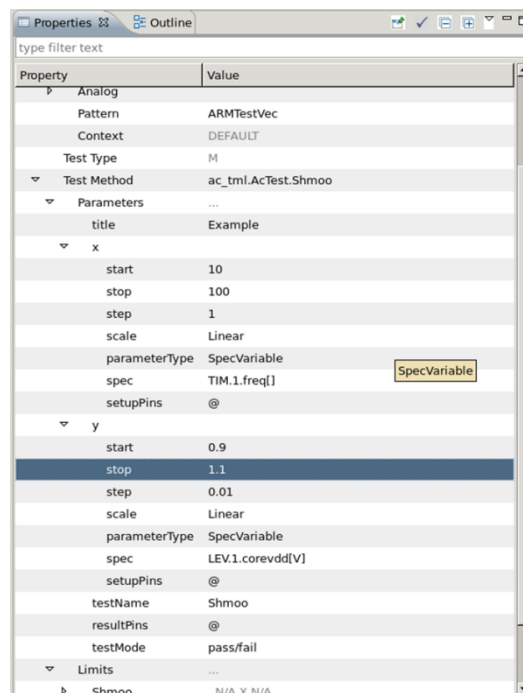


圖 21. Testsuite Properties - shmoo plot

最後則是量測 operation current，需要導入「operation current」測試項。機台所量測的 operation current 為 DUT 工作時的平均電流，因此需要重複執行 pattern 以獲得穩定的平均電流。此流程一樣參考如圖 8.~圖 14.及圖 18. 方式建立 test suite 後，接著連點兩下 Vectors icon 呼叫 Vector Setup，接著 GUI 介面的下方應該會出現 Vector 管理頁面，在待測 pattern 名字下連點兩下就可以將此 pattern 打開，接著我們必須在此 pattern 開頭的地方加入 loop

begin；結束的地方加入 loop end（因應不同 DUT 或 pattern 的設計，此 loop 可任意加在合適的地方），因此我們可參考圖 22.所示在開頭的 cycle 上按下滑鼠右鍵→[insert instruction]→[LOOP]（如圖 23.所示），再將 pattern 拉到最後的 cycle 上用同樣的方式插入[LOOPEND]（如圖 24.所示）即可。接著連點兩下 operation current test suite 的 icon 後開啟 properties，如圖 16.所示的方法輸入 passCurr_uA 的 limit value，這邊的 maxValu 建議可以設定是略小於 Level Equation 的 ilimit 參數設定值，而 minValue 則是略大於負的 ilimit 參數設定值。如此就完成了此 test suite 的設定。

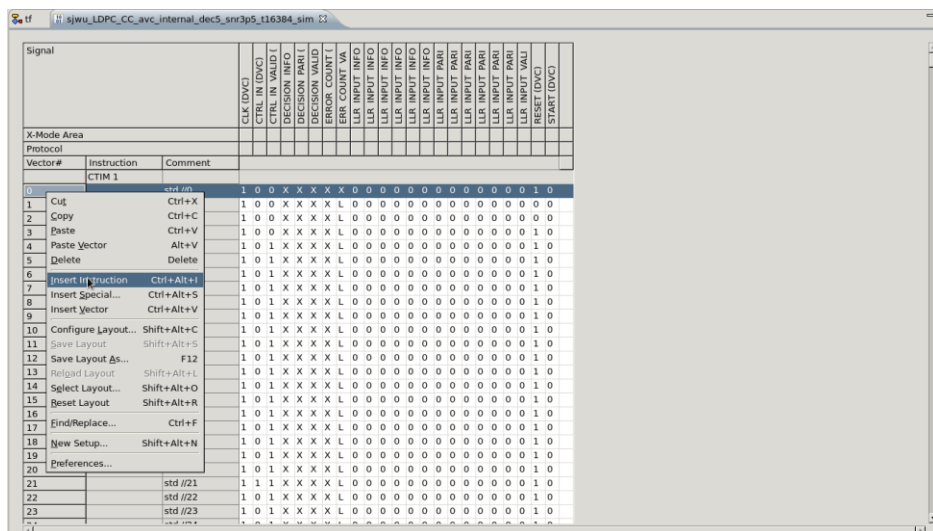


圖 22. insert instruction

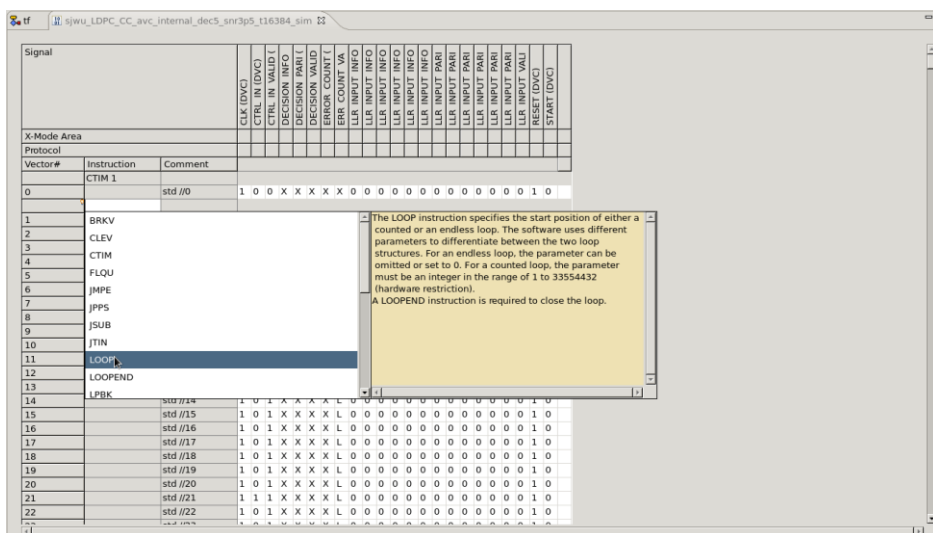


圖 23. loop begin

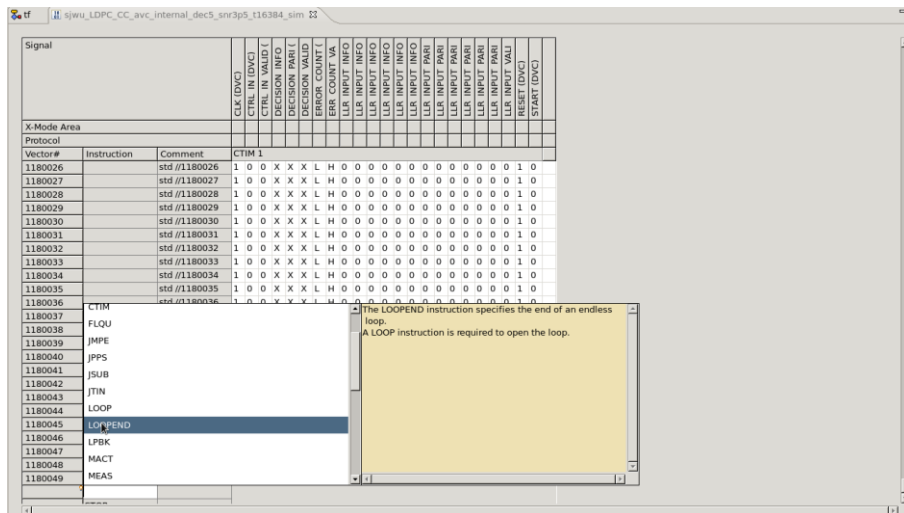


圖 24. loop end

四、晶片測試及結果分析

上述步驟完成後，應該可看見一完整的 testflow 如圖 25.所示。而 SmarTest 執行時將由上而下依序執行，當 test suite 出現 fail 時會以紅色表示，pass 則以綠色表示。執行方式可分為[Run Testflow]或[Run Selected]，[Run Selected]表示只執行選擇的 test suite，而[Run Testflow]則表示所有 test suite 都要執行。執行方式可在任一 testsuite 上按滑鼠右鍵，選擇其中一種執行方式即可，如圖 26 所示。

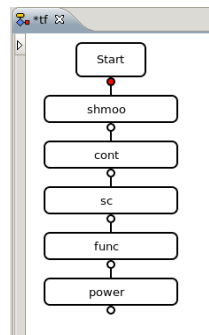


圖 25. 簡單且完整的 Testflow

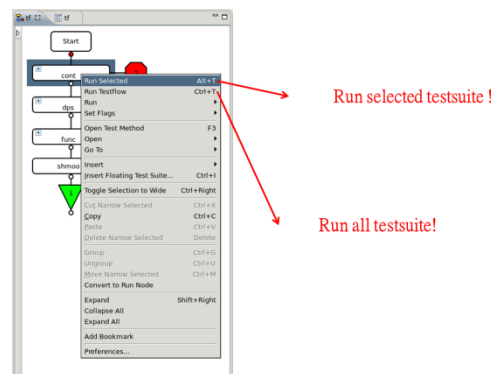


圖 26. Run Testsuite

執行完的結果將可在 ui_report_window 視窗中看到。

首先在「continuity」部分，以本中心目前提供的 cell-based design flow designkit library 的 pad library 所設計的晶片應該量測出的電壓值都幾乎落在約 400mV 左右，所以此 test suite 的 pass limit 設定在 200mV 到 700mV 之間是會出現 pass 的結果。若當量測結果接近 0V 時表示有可能此 pad 到 ATE 的傳輸路徑上有 ground 存在；而量測結果如果過大，表示此路徑有可能開路或短路到 power supply 端。

在「standby current」量測時若出現數值過大或過小則量測結果會 fail，這有可能發生在電源供應時有其他路徑放電，例如短路於 ground 端導致放電，所以會出現很大的靜態電流。若量測數值出現在 current limit 區間內的話表示 pass。一般此測試項的 pass 及 fail 並不具太大的參考價值，量測是否真正通過還是需要以使用者自行評估量測所得的真正電流消耗量為主。然可將此評估之電流消耗量設定於 test suite 中，以方便檢查通過與否。

而「functional test」必須要全部的 cycle 中所有的 strobe point 所抓取到的資料都與 pattern 中各 cycle 的預期結果相同才算通過。只要上句敘述有任何一條條件不滿足則出現 fail，針對 fail 的結果可以用 SmarTest 內建的除錯工具來進行分析及除錯。

「shmoo」的量測只是用來觀察不同環境下的「functional test」的結果，其結果如圖 27.所示，其中綠色表示此環境下的功能測試結果為 pass；而紅色表示 fail。所以一般 shmoo plot 結果應該可以看見電壓增加時，可應用的操作頻率也會隨之增加。

另一個是「operation current」量測，跟「standby current」量測相同的是當量測電流數值過大或過小時就會出現 fail，通常平均動態電流消耗會比設計時的規格要大一點，主因是量測環境的影響。一般此測試項的 pass 及 fail 並不具太大的參考價值，量測是否真正通過還是需要看量測所得真正電流消耗量為主。其結果如圖 28.所示。

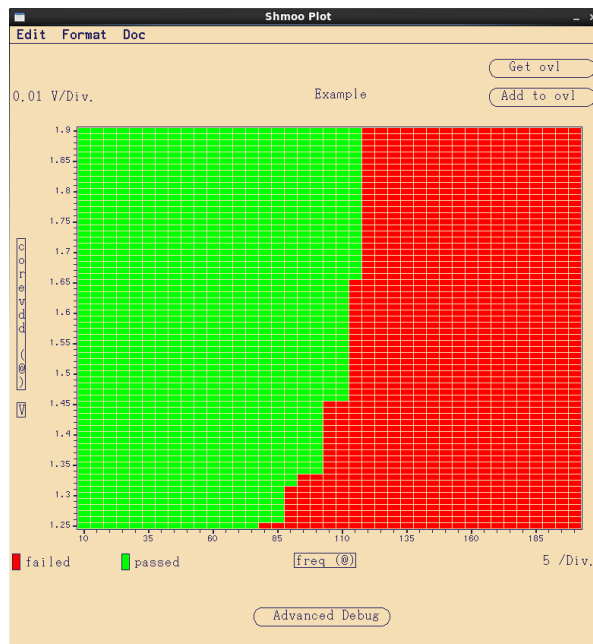


圖 27. Shmoo plot 繪製結果

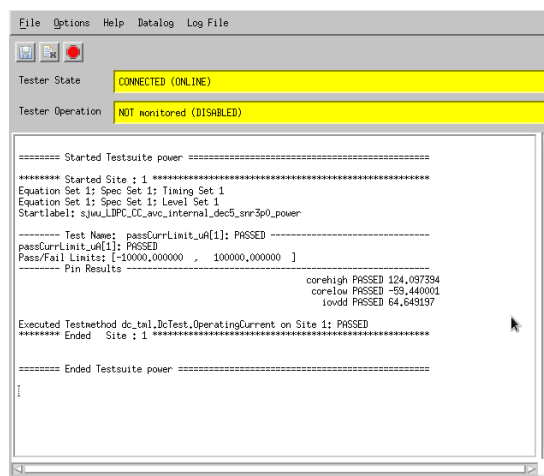


圖 28. Operation Current 量測結果

五、結論

本文簡述了本中心 Advantest V93000 PS1600 SmarTest testflow 建立方法，並介紹數位晶片量測時所需的基本測試項目。使用者可在前往本中心使用機台之前先詳閱『Advantest V93000 PS1600 基礎操作簡介』及本文件，相信能對機台操作有一定程度的了解。

六、參考文獻

- [1] verigy technologies, “93K Basic User Training – Pin Scale” July. 2010.
- [2] Advantest V93000 PS1600 基礎操作簡介