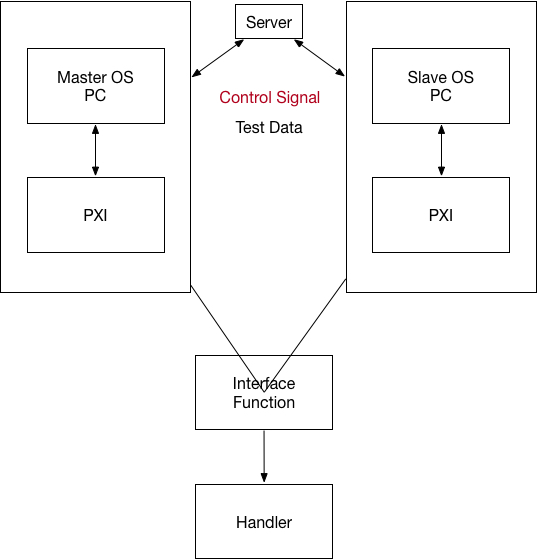
# MultiStation SGOS

Project Close Report

目的:

由於IC測試項目的日趨複雜，有時無法在同一台測試機上完成Device的所有測試，亦或單一台測試機上的資源不夠用來單獨完成Device的所有測試。於是產生將兩台測試機透過TCP/IP連線視為一台擁有兩倍資源測試機的做法，而從Handler角度來看則是分為兩站做”Serial Test”。此作法需要一獨立程式，作為同步各站資料的Server，另外操作介面程式(SGOS)則需要修改可對應Master-Slave模式的操作。以兩站測試為範例，其關係如下圖；



SGOS經過修改對應Master-Client模式後，操作都只要在Master PC上操作，其餘各站Slave PC則會在建立連線後同步動作。因此從操作角度上看可以視為一台測試機。

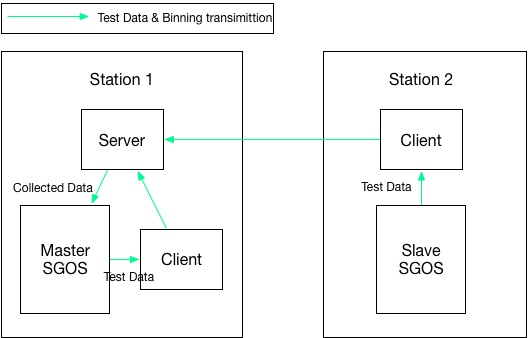
產學合作:

在上列關係圖中Server此一Block負責資料的同步與操作的同步連線，因需要TCP/IP的相關知識。因此與明新科技大學資工系戴建誠教授產學合作，開發多站式pipe-line結構測試系統。

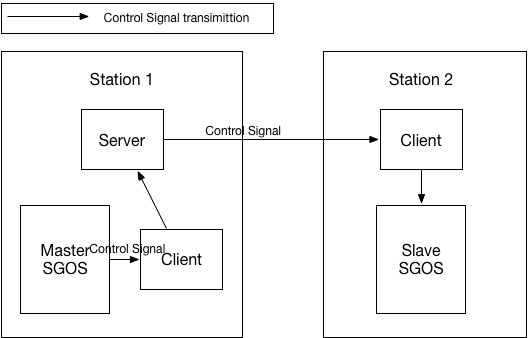
**明新大學**:

Server程式以Python程式語言開發，負責傳遞各站資訊到Server，並且整合資訊後回傳給Master SGOS。以及同步Control Signal至各站。其工作流程如圖:

I: 傳遞資訊給Server及整合後傳給Master SGOS



II: 傳遞及同步Control Signal至各Slave站



Project File List:

lib\\_\_init\_\_.py

lib\fileDirectory.py

lib\log.py

getDicConfig.py

MultiThreadSocketService.py

SendSocketClient.py

安裝設定步驟.pdf

Server程式與SGOS之間的溝通訊號指令如下所列，可分為兩類:

1. Control Signal:

start : 告訴Server新的Lot測試準備開始

isStart: 詢問Server是否Lot測試已經開始

getFilename: 取得Lot測試存檔檔名

isStartDone: 詢問各Client是否已同步start

endLot: 告訴Server要結束Lot測試

isEnd: 詢問Server是否結束Lot測試

isEndDone: 詢問各Client是否已同步endLot

1. Testing values transmission:

setDUTComplete: 將此DUT測試標記為完成測試(\*1)

EOT : 告訴Server此站測試結束

isDutEOTComplete: 詢問Server此DUT是否完成全部測試

getDutEOT : 從Server取得全部測試資料

sendBinSum : 傳送各站binning summary至server

sendBinSumDone : 訊問各站binning summary傳送是否完成同步

getBinSum : 取得各站binning summary資料

\*1: 從Handler端看multi-station測試，又稱Serial Test，若其中一站測試結果為FAIL，則此DUT皆不會做後續未測試站的測試並直接分類。而做完全部測試站的測試的DUT且最後一站結果為PASS的DUT才會被分類為PASS

**Sigurd**:

修改SGOS內容以對應Master-Slave操作模式，包含透過Client程式傳遞資料與Control Signal。讓SGOS可設定成Master或Slave兩種模式。Slave Tester除了產生當站的測試報表以外，Master Tester則將各站測試資料彙整合併後，產生完整的測試資料報表。

要在SGOS開啟MultiStation 測試，有兩處設定需要加上去

1. SG9000II.ini

* [Tester]
* Name=TSG01
* OSVer=V1.17
* OSVerControl=V1.17.0.0030
* Interface=Engineer
* SetupPW=OFF
* HWScan=OFF
* SocketControl=OFF
* SecretEng=OFF
* MultiStationNo=1

在[Tester]欄位中增加”MultiStationNo”以設定此台Tester為第幾Station，在SGOS中，1為Master，比1大的數字為Slave Station，0則為無MultiStation

1. 測試程式的Table file

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TestID | UserFunc | TestItem | StopOnFail | PAT | MultiStaionNo |
| 10 | Default | Interface\_test\_Bin1 | 0 | 0 | 2 |
| 20 | Default | Interface\_test\_Bin2 | 0 | 0 | 2 |
| 30 | Default | Interface\_test\_Bin3 | 0 | 0 | 2 |
| 40 | Default | Interface\_test\_Bin4 | 0 | 0 | 2 |
| 50 | Default | Interface\_test\_Bin5 | 0 | 0 | 1 |
| 60 | Default | Interface\_test\_Bin6 | 0 | 0 | 1 |
| 70 | Default | Interface\_test\_Bin7 | 0 | 0 | 1 |

在Table file的最後面新增”MultiStationNo”欄位標示每一test item屬於哪一站

Source Code Modification:

1. TTimer為BCB VCL Library提供的時間物件, 此物件為MultiThread運作，可以設定每隔多久時間便執行物件中的Public函式。

Control Signal的同步便使用了此一Timer物件，可參考TimerMultiStationTimer()函式。對於Master站而言，Start送出後 TimerMultiStationTimer便會不斷詢問isStartDone。對Slave站而言，一按下start後，TimerMultiStationTimer()便會不斷詢問isStart。而當測試開始後Slave站的TimerMultiStationTimer()則會改為不斷詢問isEndLot，直到接收到Master傳送的endLot訊號。

而在測試中，Timer\_INTTimer()除了不斷詢問SOT訊號以外，Master站的Timer\_INTTimer()會不斷詢問isDUTEOTComplete，以便接收到前幾站有Fail的DUT資料。

1. Bin Summary: 為了列印出各站統合的Bin summary資料，所以除了test data以外各站還會傳送Bin summary資料。在Master站以StationBinCount存放資料，而StationBinCount的型態則與BinCount相同。只差在BinCount是以各site存放資料，而StationBinCount是以各站來分類存放資料。