

第五章 驗測平台整合系統設計

第一節 驗測平台硬體結構與電路設計

本研究對於 FRC 功能的驗證與測試甚為重視，由於 FRC 可連接 PC 或是微控制器作為驗證與測試的連線工具，為求操作介面人性化，本驗證測試硬體平台將透過 PC 端軟體程式，把經影像切割處理後的指紋影像，透過與 FRC 傳輸交握演算法，經由 FRC 的控制，將指紋影像載入至外掛式記憶體中。

當 FRC 將指紋影像經指紋前處理與編碼的動作後，可將處理後的結果回傳至 PC 端驗證結果是否與預期符合，這樣的設計，目的在於提昇在設計過程的除錯的能力，縮短除錯的時程 運算結果的驗證與測試以及未來其他功能擴充的便利。

其中 PC 與 FRC 的傳輸交握方法將採標準並列埠交握協定(Standard Parallel Port, SPP)來實施，透過 PC 的印表機埠(printer port)將指紋影像傳輸到 FRC，其硬體驗證平台架構如圖 44 所示。

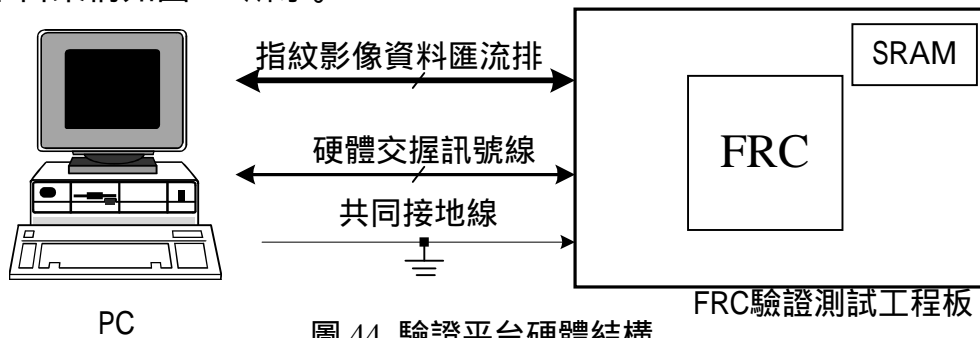


圖 44 驗證平台硬體結構

在 FRC 驗證測試工程板中，使用 Xilinx 公司出品的 FPGA，型號為 Virtex XCV300。其中 XCV300 中的 300 表示含有 30 萬閘數(gate count)的資源；此外 Virtex XCV300 為 PQFP240 的包裝，核心電壓固定為 2.5V，I/O 電壓的範圍為 3.3V 至 5V，可由使用者調控，。

因為 FRC 內部並沒有足夠的 SRAM，所以必需使用外掛式的記憶體做為存放指紋影像資料的緩衝區(fingerprint image data buffer)，目前選用記憶體型號為

BS62LV4001 的 CMOS SRAM，編號中的 LV 表示為 Low Voltage 的意思，也就是工作電壓可介於 2.4V~5.5V 之間。由於驗測工程板目前選用的電壓為 3.3V，所以正好可與其配合；編號中的 4001 的意思表示 512K x 8 bit，共 512Kbyte，本記憶體存取延遲時間為 70ns。

就外掛式記憶體而言，目前在本研究所設計的硬體驗測平台所選外部記憶體寬度使用的是 8bit，未來可選用的記憶體寬度為 16bit，那麼在整體設計的效能上更是可以大幅提昇，或是將目前選用的外掛式記憶體由原有的 SRAM 改選用 SDRAM，如此可以提昇對記憶體的存取速度。

電源電路的設計裡，因為 Xilinx Virtex XCV300 需要兩組電壓(核心電壓與 I/O 電壓)才可正常工作，其中一組電壓為 2.5V，另一組電壓則是 3.3V。正因如此，在設計上採用兩個線性電壓調整器(linear voltage regulator): LM317 作為電壓供應的調控元件，因為 LM317 可透過外部的電阻調整輸出電壓，所以很適合做為電源模組的主要元件，此外它也擁有寬廣的輸入電壓範圍，目前設計為 5~12V 的直流電壓輸入皆可正常動作。

整個 FRC 驗測平台硬體架構，如圖 45 所示，工程板大小約為 360mm x 240mm。其中因為所選用的記憶體為 512K x 8bit，所以在位址匯流排(address bus)的需求上，需要 19 條($2^{19}=512K$)硬體位址線，而資料匯流排(data bus)的寬度則為 8bit，所以共需要 8 條資料線，因為所輸入的為 8bit 灰階指紋影像，所以選用資料匯流排寬度為 8bit 的記憶體，正好可以配合。

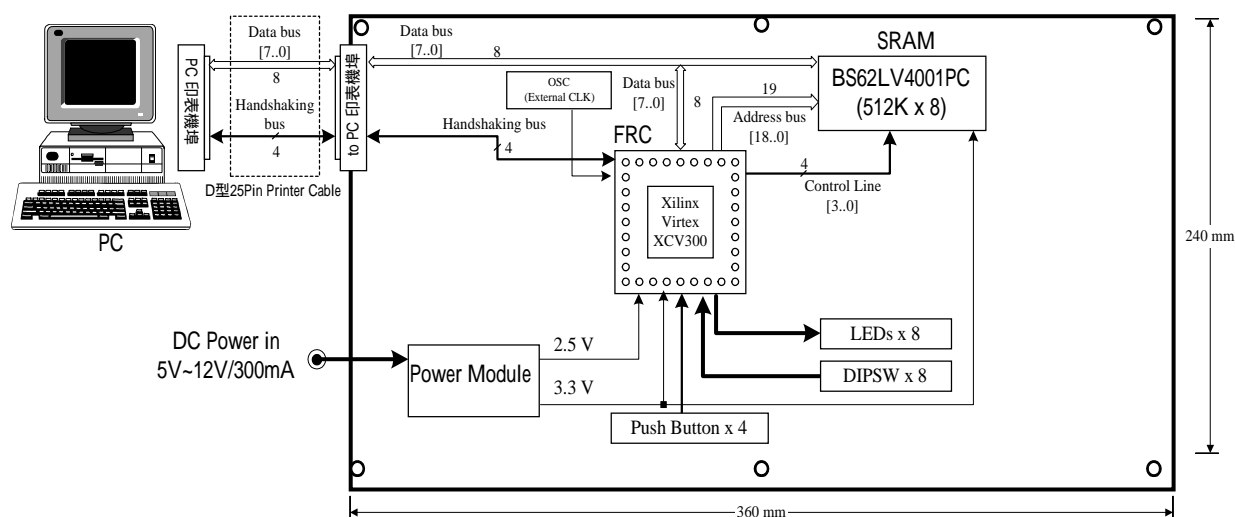


圖 45 驗測平台電路設計

在 PC 端與 FRC 的傳輸上，透過 PC 端的印表機埠與 FRC 進行傳輸交握，其命令與對應的動作如圖 46 所示。透過 PC 對 FRC 所下的命令，FRC 便會依其指定的命令執行對應的運算。

命令代碼	執行運算
1	Binarization
2	Thinning
3	Ending Point
4	Bifurcation Point

圖 46 FRC 命令與相對動作對照

第二節 軟體功能模組設計

在 PC 端所設計的軟體，名稱稱之為 FRCTOOL，用來與驗測平台達到對所設計的晶片驗測的目的。在程式語言的選用上，使用 Visual Basic 6.0 程式語言(以下簡稱 VB)所設計完成，其中為了讓整個程式更為模組化，所以將不同類型功能的函式，使用不同的模組分別管理，模組的功能分別為：傳輸交握通信模組、檔案結構處理模組、指紋影像前處理與編碼軟體演算法模擬模組，進制轉換處理模組以及影像顯示/存檔模組，整個 FRCTOOL 軟體功能模組的架構如圖 47 所示，各個模組功能逐一說明如下：

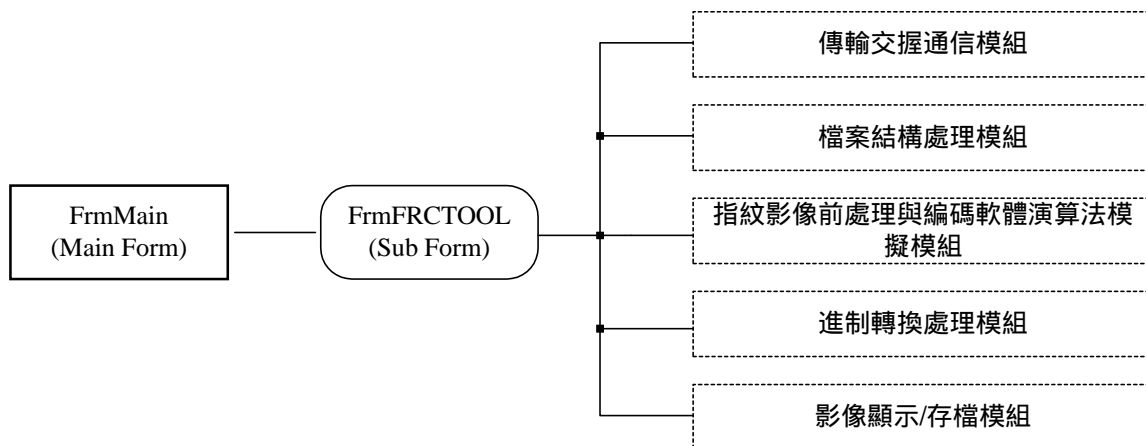


圖 47 軟體功能模組架構

(一) 傳輸交握通信模組

在 VB 中，沒有提供可以對 I/O 位址直接存取的函式庫，所以改採用由 Visual C++ 所建構出來的存取 I/O 的函式庫，達到對 PC 的印表機埠存取的動作。由於在傳輸交握上所採用的是 SPP 交握協定，所以必需要瞭解印表埠中資料暫存器如圖 48 所示，狀態暫存器如圖 49 所示，以及控制暫存器如圖 50 所示硬體所在位址，才可利用 I/O 函式庫，建立與驗測平台溝通的傳輸交握函式庫。

資料暫存器(基本位址)				
位元	D 型接腳	信號名稱	輸出/入方向	硬體反向
0	Pin 2	Data (bit0)	輸出/輸入 (透過控制暫存器的 bit 5 設定輸出入方向)	否
1	Pin 3	Data (bit1)		否
2	Pin 4	Data (bit2)		否
3	Pin 5	Data (bit3)		否
4	Pin 6	Data (bit4)		否
5	Pin 7	Data (bit5)		否
6	Pin 8	Data (bit6)		否
7	Pin 9	Data (bit7)		否

圖 48 印表機埠資料暫存器內容

狀態暫存器(基本位址+1)				
位元	D 型接腳	信號名稱	輸出/入方向	硬體反向
0		Time Out	可做 Time out 指示	
1		Unused		
2		Unused		
3	Pin 15	<u>nError</u>	輸入	否
4	Pin 13	Select	輸入	否
5	Pin 12	<u>PaperEnd</u>	輸入	否
6	Pin 10	<u>nACK</u>	輸入	否
7	Pin 11	Busy	輸入	是

圖 49 印表機埠狀態暫存器內容

控制暫存器(基本位址+2)				
位元	D 型接腳	信號名稱	輸出/入方向	硬體反向
0	Pin 1	<u>nStrobe</u>	輸出	是
1	Pin 14	<u>nAutoLF</u>	輸出	是
2	Pin 16	<u>nInit</u>	輸出	否
3	Pin 17	<u>nSelectIn</u>	輸出	是
4		Interrupt Enable (1:致能印表機中斷,0:除能印表機中斷)		
5		Data Directional Control (1:設定印表機埠為輸入,0:設定印表機埠為輸出)		
6		Unused		
7		Unused		

圖 50 印表機埠控制暫存器內容

根據圖 48~50 可以得知，若要控制印表機埠，必需要先決定印表機埠的基本位址；通常印表機埠固定基本位址為 0x378、0x278 或 0x3BC，這可以由 PC 端的

BIOS 決定，或是使用組合語言來決定硬體位址，本設計所選用的是一般較常用的 0x378(LPT1)。本設計所使用的傳輸通信交握控制流程如圖 51 所示，而圖 52 為本方法之軟體 Pseudo Code。整個傳輸交握通信模組的函式架構如圖 53 所示。

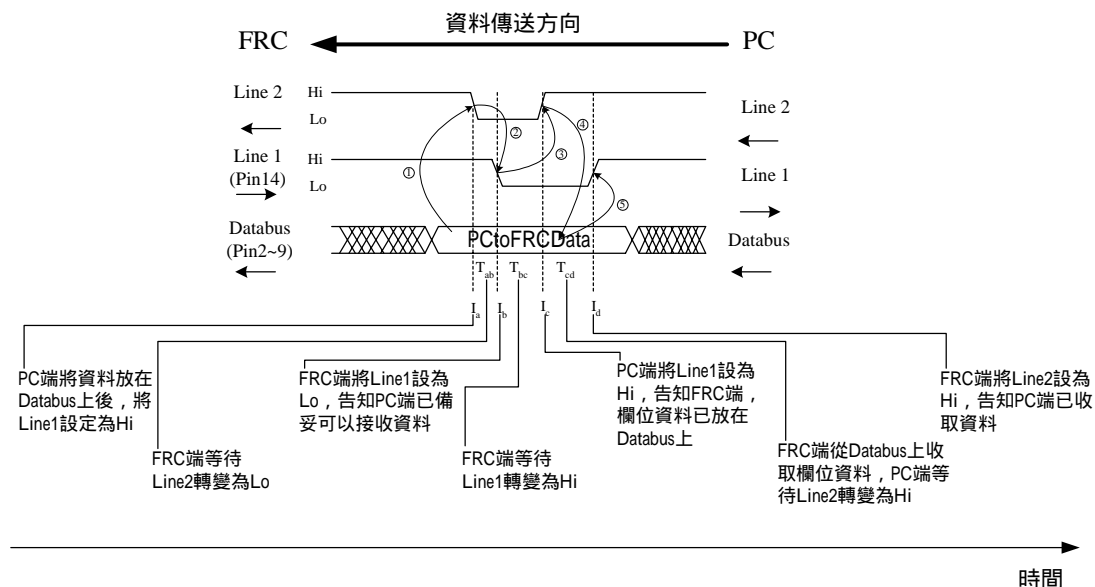


圖 51 傳輸交握控制流程

```

Public Function SPPXferToFRC(Data as Byte)
    Output(Data)           ;將傳送的資料放在資料匯流排上
    SetLine1Lo             ;設定 Line1 為低態
    WaitLine2Lo           ;等待 Line2 為低態
    SetLine1Hi            ;設定 Line1 為高態
    WaitLine2Hi           ;等待 Line2 為高態，表示 FRC 已取得資料
End Function
    
```

圖 52 傳輸交握 PSEUDO CODE

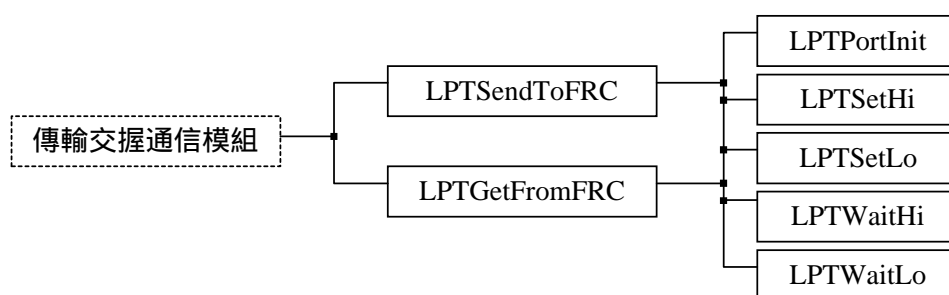


圖 53 傳輸交握通信模組函式架構

(二) 檔案結構處理模組

檔案結構模組主要是處理將指紋影像檔案，以 16 進制的方式將內容表示出來的轉換，由於本研究所選用選擇的指紋影像檔案格式為 8bit 灰階影像，所以每一像素剛好為 1 個 Byte，為了在設計的過程中可以方便的觀察指紋影像與其所對應

(四) 進制轉換處理模組

由於在處理原始指紋 8bit 灰階影像時，其每筆像素的內容都是以 1Byte(範圍為 0x00~0xFF)來表示，為方便對於影像資料的輸出入，所以設計進制轉換處理模組，方便進制的相互轉換，圖 57 為進制轉換處理模組的函式架構，其中提供 16 進制、十進制與二進制的數值與字串的相互轉換。

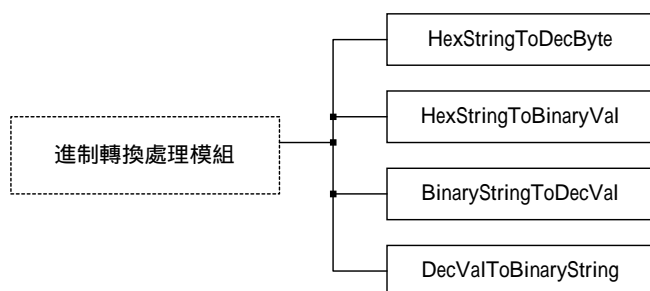


圖 57 進制轉換處理模組函式架構

(五) 影像顯示/存檔模組

在 VB 程式開發中，若要顯示灰階指紋影像，必需透過呼叫 Windows API 的介面，才可順利達成，圖 58 為影像顯示/存檔的函式架構，其中除 SaveFPIMG 不是呼叫 API 介面外，其餘皆是利用 API 達成指紋功能。

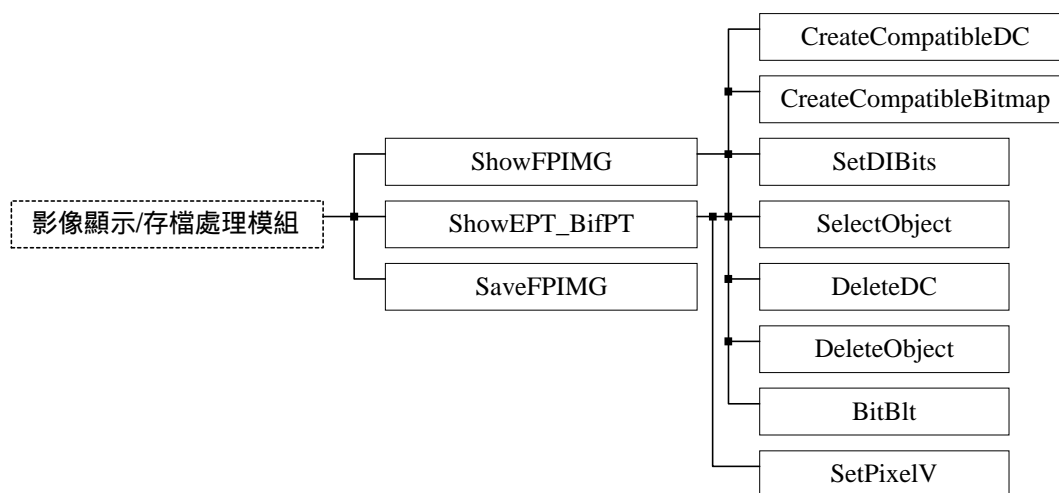


圖 58 影像顯示/存檔模組函式架構