

# 數位邏輯電路模擬器之設計與實作

莊作彬

國立屏東商業技術學院資訊科技系  
助理教授  
電腦輔助設計與高速運算實驗室  
tsobing@npic.edu.tw

林俊良

國立屏東商業技術學院資訊科技系  
電腦輔助設計與高速運算實驗室  
s93413045@student-mail.npic.edu.tw

## 摘要

本系統以簡單易懂的設計，擺脫一般硬體模擬器生硬難懂的操作，以清楚明瞭的體貼風格，詮釋硬體底層精密的結構，不論在輔助教學，學生撰寫作業、或者教師在批改成績上，都有很大的成效。加上嚴謹的設計規劃，對於日後新增元件，保留更多彈性發展的空間。

**關鍵詞：**數位電路設計、模擬器、合成

## 一、前言

對於研習數位邏輯的同學而言，最抽象的莫過於電子訊號的表達，傳統教授課程往往以黑板作圖為主，但對於訊號之間的轉換，卻無從表達，這是由於教科書本過於靜態，移除其中一個元件，所有數值也要重新擦掉重寫，若改由動態的電腦影像做為輔助教學，相信在表達電路微妙機制時，會更加的靈活生動，本論文特點是設計具有「互動性」的補助教學系統，透過按鈕、拖曳等行為，引導學習者「了解電路」、「使用電路」、「設計電路」，等進階的課程，促進師生良好的互動，期能達到溝通與分享的教學品質。

一般的硬體模擬開發環境，由於是設

計給專業人士使用，初學者不易上手，也很難應用在教學用途，網路上的硬體模擬器固然便利，但往往缺乏互動性以及自我設計的彈性，本系統目標在改良了這些的缺點，設計一套專為教學用的硬體模擬系統。本篇論文先以加法器產生系統為例，其他電路均可利用本系統所使用的原理實作。

本論文架構如下：第二節說明系統整體架構，第三節介紹所使用的相關技術以及系統實作考量，第四節是系統的執行結果，第五節則是結論。

## 二、整體架構說明

以往的模擬程式都是用單一語言設計，本系統的特色是使用二種語言，由於 C 語言功能強大，有利於日後程式的維護與更新，並可做更強大的運算，包括指標設計、鏈結串列，而且 C 語言執行效率高，有利於更複雜的計算；對於視窗介面，則使用了 VB 來製作，這樣可以大幅縮短開發時間。一般而言，使用者都是以圖形化介面進行操作，透過 VB 語言來進行核心的呼叫，最後再把核心運作的結果，以圖像方式顯現出來，VB 便成了使用者與核心溝通的橋樑，如圖 1 所示，讓使用者輕鬆的操作，免除額外的技術與知識。此外，由於執行與顯示二者作業並不重疊，各司

其職，對於開發者在日後的維護控管也會更加容易。

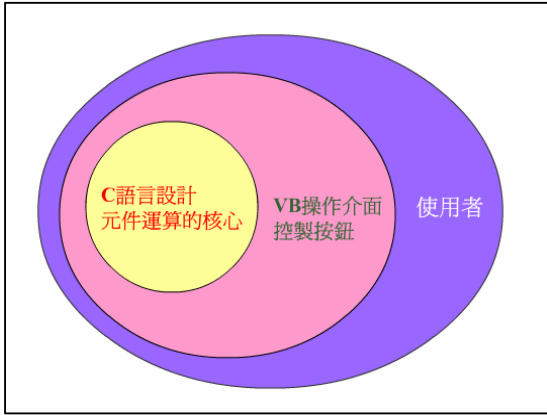


圖 1 整體架構圖

我們依據不同功能和目標，使用不同程式來分開撰寫，等到二邊都設計完成後，再把以 C 程式設計的部份匯出為 DLL 檔，以 VB 進行連結，這樣的優點就可以用 VB 設計的圖形介面來操作所有的功能，讓使用者只需要以選單、按鈕，就可以操作所有的功能，如圖 2 所示，優點是使用者完全不用懂程式內容和底層設計原理即可使用。<sup>[4]</sup>

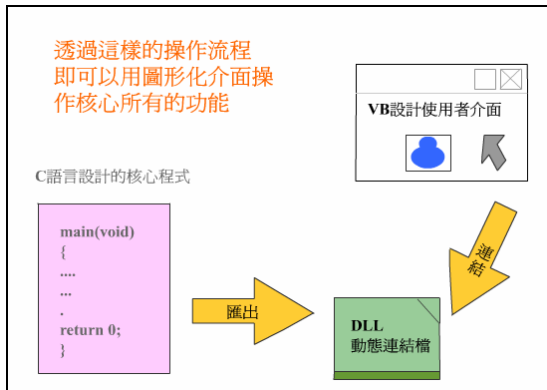


圖 2 動態連結 DLL

### 三、相關技術與系統實作

這個部份我們所要說明就是「使用的技術與實現的原理」，過去硬體模擬器有許

多缺點，要使用那些技術來達到改良的目標。我們的特色在於設計互動式操作模組、簡單親和力的控制流程、更有彈性的自由組合…這些理念到底是如何實現的？我們會在這部份做更詳盡的介紹。

首先，我們來探討製作流程，我們以鏈結串列為資料結構，新增節點（這裡稱做元件的容器），每個節點就當做一個容器的空間，如圖 3 所示，可以放入各式各樣的元件。

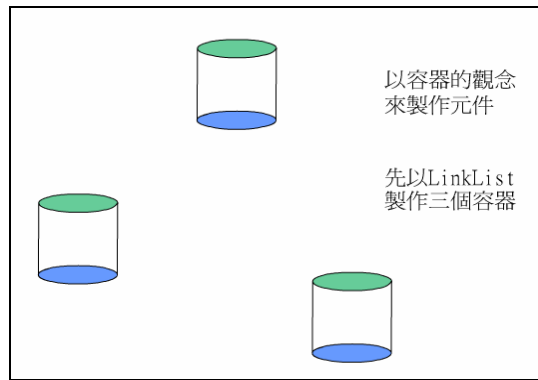


圖 3 鏈結串列

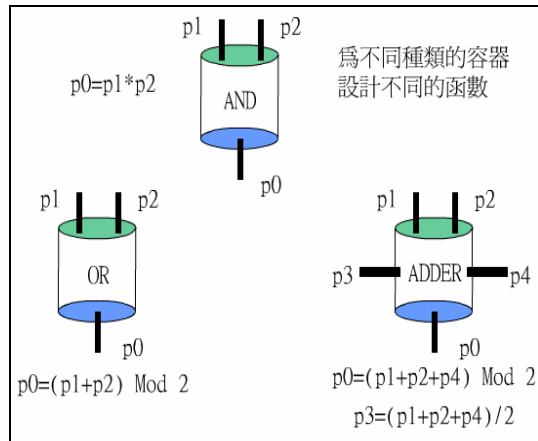


圖 4 不同容器對應不同函數

為了讓每個容器有不同的特性，我們會設計不同的函數，像 AND 的元件我們會用 C 語言寫成：

$$p0=p1 \& p2 ;$$

**p0 是輸出**

**p1 和 p2 是指輸入**

當容器執行所對應函數後，不同函數就會表現出不同的特性，透過這樣的方式，就能模擬出不同的數位元件，如圖 4 所示。

接下就會依照不同元件的類型，執行所對應的函數，不同函數，就會產生出不同效果，透過這樣，就能模擬出元件的特性，如圖 5 所示。

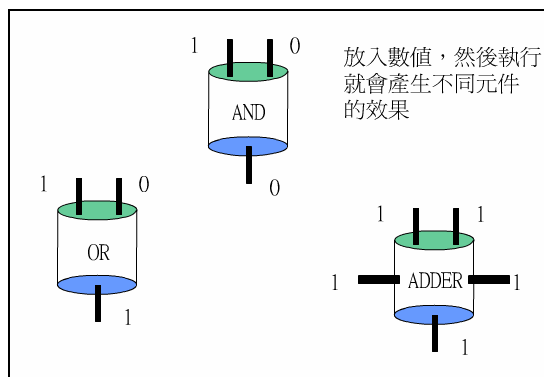


圖 5 執行所對應的函數

的效果，組合完成，而且程式不必寫死，只要設計一個，以後相同的部份，就不必重新製作，可大幅縮短開發時間，如圖 7 所示。\*核心開發([3][4])

以函數來模擬元件的方式，一開始是在設計多位元加法器時研究出來的，後來發現不只是加法器，其它的組合元件都可以同樣的方式一併完成，所以利用此特性，相繼的製做出不同的元件，組合成一套模擬開發系統。

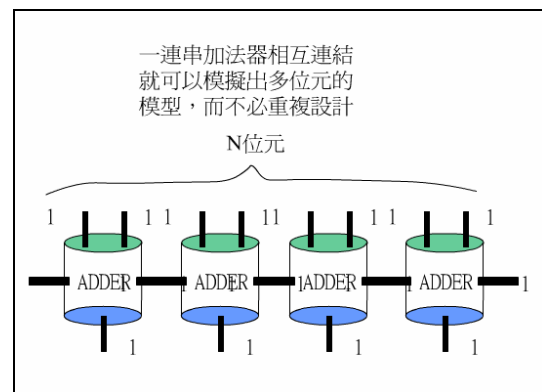


圖 7 加法器的鏈結效果

每個容器所產生出來的值，是可以傳給不同容器的，透過這種方式，每個容器就不再是單一的特性了，會彼此相互傳值，來影響其它容器的輸出入，就像是一整片的電子元件，一個接著一個影響，即可詮釋出硬體電路的機制，如圖 6 所示。

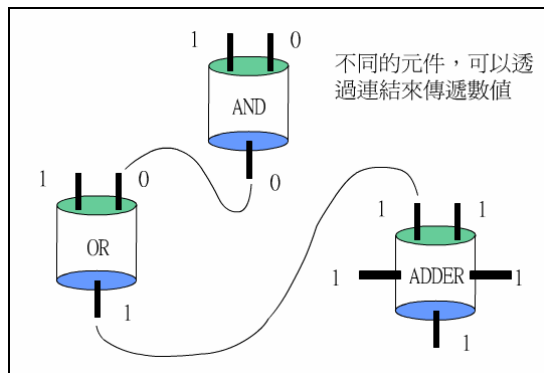


圖 6 元件間的鏈結會相互影響

一套模擬系統最難設計的就是執行系統，要怎麼運算才能做到每個完件都有算到？如何排列出先後次序？這需要定義優先次序，上游元件算完後，才能算下游的元件。對於此，本系統在這裡設計了很靈活的演算法，可有效的做執行的排程，還有更進階的死結的偵測。(圖 8)

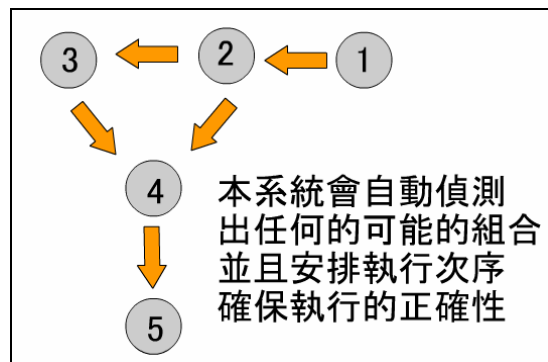


圖 8 設定排程的優先次序

設計多位元加法器，也可以透過鏈結

如何為任意元件安排執行順序？（如圖 9 所示）圓圈內的數字為元件的編號，我們會從 1 號開始詢問，如果 1 號上游沒有元件、或者上游元件已執行，我們才會開始執行 1 號的計算，接下來詢問 2 號上游是否有元件？有無執行？…以此類推，等到完成詢問所有的序號，只要這回合仍有元件未執行，就會在下一回合從未執行的元件開始詢問，同樣也是從小到大的編號。

從以上規則我們可以得到執行的結果：

第一回合完成執行：6

第二回合完成執行：2、3、5

第三回合完成執行：4

第四回合完成執行：1

\*用此運算法不斷的循環處理，就可以依序把所有的元件執算出來，而且不會重覆又能有次序的相互傳值達到模擬元件的效果。

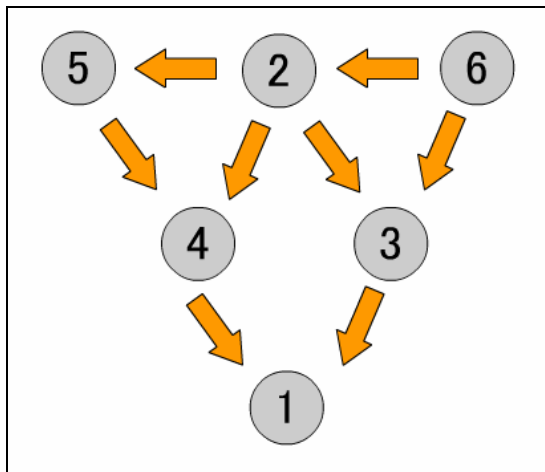


圖 9 為任意組合的元件做優先排序

在有些特別的情況下，我們沒辦法把所有元件值給算出來，這就是所謂的「死結」，死結為什麼會發生？基本上，我們一

開始在設定外部連結時，就已經巧妙的設定優先序列了，舉個例子來看：我們現在設計二個元件，加法器 A 與加法器 B，我們定義 B 元件是需要向 A 來取得進位輸出，其實這當中已意味著，需要等到 A 執行後，才能做 B 的運算，A 有高於 B 的執行優先權，換一種角度來看，B 會隨著 A 值不同而不同，但 A 卻不受 B 影響，A 決定 B 往後的運算，所以 A 執行權限較高。本執行系統即是以此為出發點，但在有些時候，使用者不周密的設計，就會在此發生問題，例如我們定義：B 向 A 取值，且 A 又向 B 取值，這樣的情況下，二個元件會不斷相互等待，即使過了數個回合後，仍舊沒辦法完成執行。

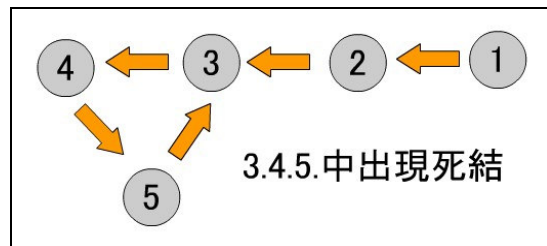


圖 10 因為相互等待而發生的死結

我們以圖 10 為例，元件 1、2 在執行都不會發生問題，到元件 3 時，因為 3 號被設定需要 5 號的值，5 號需要 4 號，4 號又需要 3 號，造成循環等待而發生死結，對此，本系會自動偵測出來，使用原理如下：

第 1 回合完成執行元件：1、2

第 2 回合完成執行元件：無

第 3 回合完成執行元件：無

第 4 回合完成執行元件：無...

由於是死結，所以往後回合都沒辦法有正確的執行...

此這裡，我們可以發現，只要有一回合沒有執行到元件，且仍有元件未執行，這就表示死結已發生了，所以我們只要在這裡設計一個計數器，記錄每回合的執行數目，如果為 0，本系統會以跳出視窗告知死結。

在此我們可以得到結論：**一個有效的設計，會在每回合至少有一元件被執行，在  $N$  個元件的情況下，執行複雜度為  $1 < O(n) \leq N$ ，最多需要  $N$  回合，最少 1 回合即完成所有執行。**

#### 四、系統執行結果

本系統的硬體需求很低，我們在 Windows XP 作業系統下，使用 Pentium 4 (2.4GHz) 的 CPU，512 RAM 就能順利執行所有功能。使用者初次使用本系統，就會發現相當的精簡，而且直覺易上手，從功能表大致上就能看出其功能，像是建立元件，開啟檔案，存檔，執行運算，這樣主要的功能會很明確的標示，如圖 11 所示，而後續處理的動作，也會幫使用者著想，比如：開啟舊檔之後，就會自動執行，並且更新結果，大的架構完成後也會接連細部的處理，使用者不需擔心下一個步驟該做什麼，另外，對於較需要花時間的處理，系統也會使用時間刻度的顯示條，告知現在所進行的進度，有別於一般的模擬軟體，這是相當體貼的設計，如圖 12 所示。



圖 11 清楚明白的功能表設計

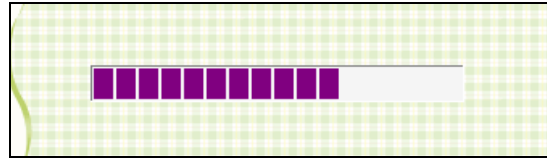


圖 12 使用進度列告知執行進度

使用本程式設計的元件，在儲存後的容量非常的小，使用變動長度的 Record，平均儲存 100 個元件，不到 3KB，如圖 13 所示，即使磁碟片也能便於攜帶，這非常有利於課堂的輔助教學，而且操作容易，學生使用本軟體可以在家完成數位邏輯的作業，最後把完成的組合電路儲存起來，再轉交由老師批改，而教師也可以利用本系統立即執行，驗證是否正確，大幅減少批改作業的時間，有利於師生的互動。

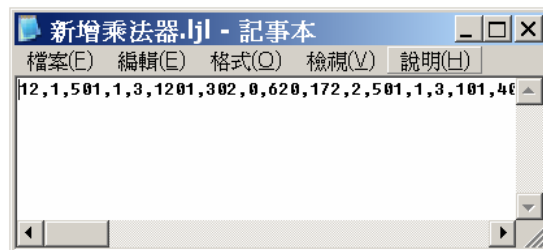


圖 13 變動長度儲存，耗費容量小

針對於不同的使用者，滿足不同的需求：透過適當的詢問對話框，如圖 14 所示，來了解使用者的取向，有時，預先設想的流程看起來似乎是好的，是體貼顧客，但對於有特別用途的客戶。卻未必能達到滿足，所以在這裡，我會設計二種介面，**圖型介面**與**文字介面**，如圖 15 所示，讓不同用途的使用者做適當的切換，一般模式下，可以先把常用的功能在一開始時都匯入，另外再加一個喜好設定，這樣就能適時做個人品味的設定。

\*介面設計([1][2])

\*線上技術支援 ([5][6])

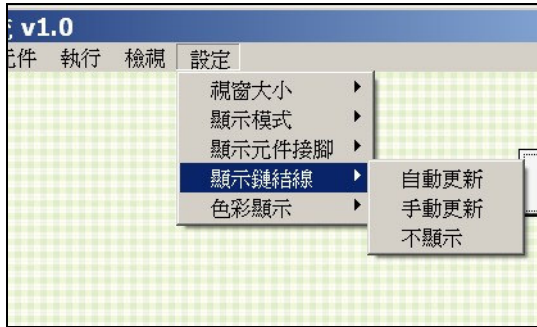


圖 14 針對細節，設定個人習慣

圖 17 改良後，融入舒適的美學，簡潔易懂



圖 18 非常清楚直覺的對話框

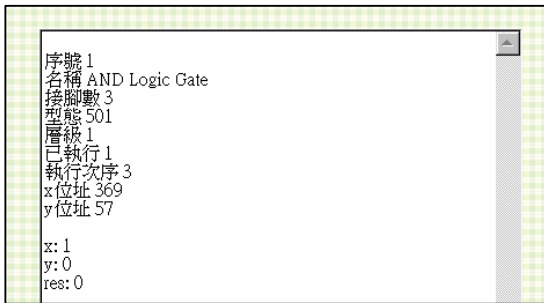


圖 15 本系統亦提供文字的介面

用不同色彩的方塊來表達不同意義，是本系統的特色之一，如圖 19 所示，右邊方塊白色表示未執行的，左邊粉紅色則是已完成執行，綠色方塊則是有外部的鏈結，至於大圖型中的 5，則表示優先執行順序，大方塊 5 會先執行，然後才是大方塊 6，系統會自己運算先後次序，並在註記在色塊當中，清楚的表達每個元件的狀態。

本系統有別於一般較舊式的網路模擬器，純然只是用圖片拼湊而成的，如圖 16 所示，我們提供了相當有彈性的互動，精簡的圖示，如圖 17，還可以接受滑鼠點擊、拖曳，把所有細部的功能都設計在滑鼠鍵中，富豐的色彩表達以及人性化的介面，如圖 18，擺脫以文字陳述的設計，取而代之的是讓使用者盡情發揮的寬廣空間。\*國外網站參考 ([7])

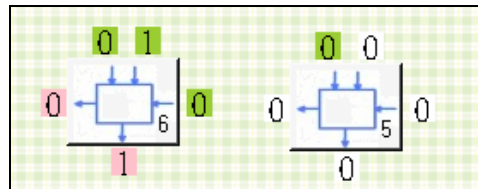


圖 19 用不同色塊來表達不同的用途

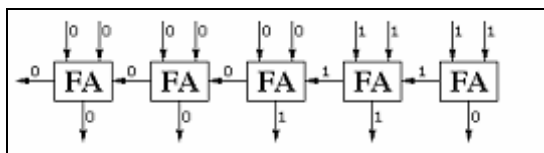
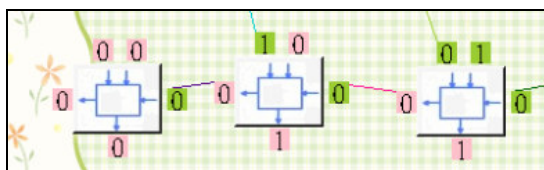


圖 16 舊式模擬器單調缺乏互動

**簡易輸入框**，是本系統創新突破的設計，透過簡易輸入框，可以讓使用者輸入數值，當數值改變後，就會立即把結果一並更新，而且輸入框只要用滑鼠右鍵點一下，就可以改變成 1 或 0，在舊式的模擬器往往需要多出一個輸入視窗，此設計是相當精簡而體貼的，也更加凸顯本系統的便利性。



在下圖 20 中，上方 6 組白色的就是簡易輸入框，只要輸入數值，下方會元件也

會一併更新，只要在輸入框上面點滑鼠右鍵，就會變化成 0 或 1，對於教師上課授課，這是相當實用的設計，只要上課前做好元件模型，透過修改輸入框值，就可以讓同學看到每個元件數值的變化，相當生動有趣。

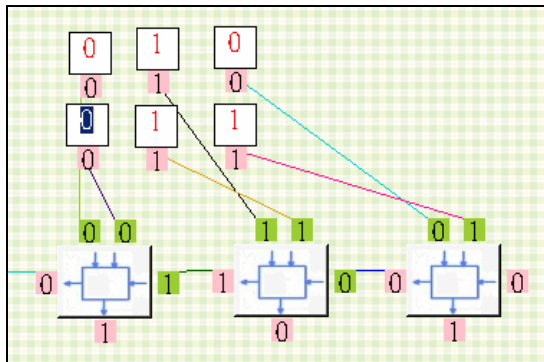


圖 20 上方輸入框可以幫助修改輸入參數

對於電路的表達，我們設計了不同以往的顯示方式，舊式的模擬器的線路都只有單色來表達線條，這樣容易造成混亂，當元件數越來越多後，在辨視上會有極大的困難，如圖 21 所示，若改良為多顏色的線路，就顯得清晰有層次感，追蹤更為容易，讓使用者能更清楚閱讀資料，如圖 22 所示。

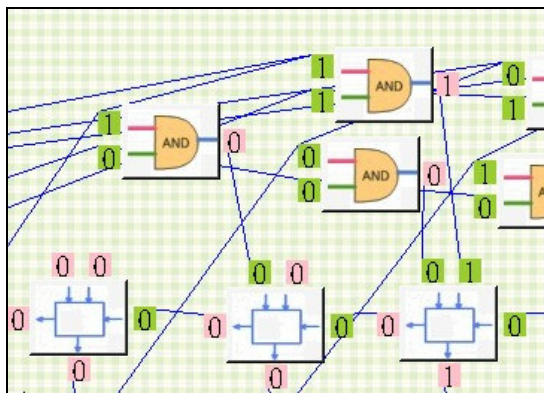


圖 21 單色的電路表示，容易混亂

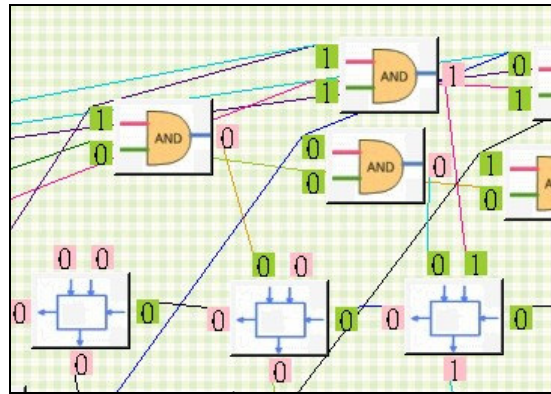


圖 22 改成多色顯示，清晰有層次感

**死結偵測：**對於不良設計而造成的死結問題，本系統亦提供偵測的機制，在執行過程中若發現死結會跳出警示，並標示異常迴路，正常執行的元件會以粉紅色來顯示，錯誤迴路則以白色，提醒使用者修正，如圖 23 所示。

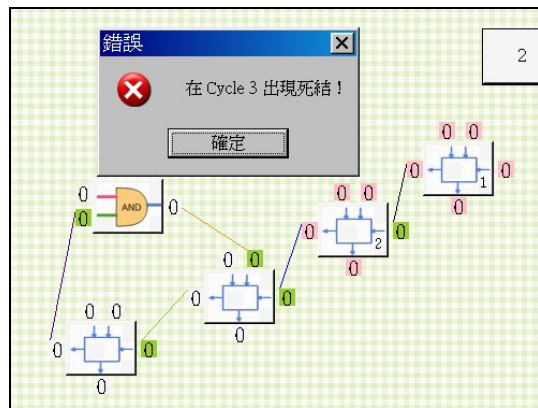


圖 23 系統偵測出死結時會發出警訊

**更多人性化的細部設計：**儲存檔案在使用時雖然方便，但因為沒有顯示會存到那個目錄，往往使用者不注意，便把先前的檔案給覆蓋掉了，本系統會顯示即將覆蓋的檔案及目錄，再三提醒使用者，這是一般的商業軟體很少考慮到的部份，如圖 24 所示。

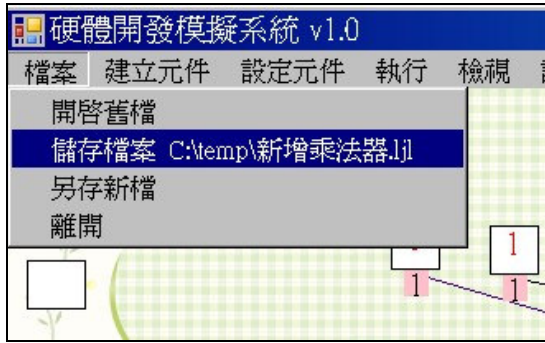


圖 24 對於每筆資料的儲存，都有明確的顯示

## 五、結論

在本論文當中，我們提出模擬器設計的新思維，改良以往生硬難懂的操作流程，取而代之的是簡潔貼心的設計，並融合新技術：死結錯誤偵測、簡易輸入框、多色電路表示法...等強大功能，在教學實務的應用上，有相當顯著的成效，增進師生的互動，提升更優良的教學品質。

## 參考文獻

- [1] 黃世揚、何嘉益，Visual Basic.Net 學習範本，文魁資訊，2003。
- [2] 羅慧真，Visual Basic.net 完全探索，學貫圖書，2002。
- [3] 吳勁樺，資料結構教學範本 C++，金禾資訊，2006。
- [4] 位元文化編著，Visual C++.Net 入門進階，文魁資訊，2004。
- [5] Microsoft 官方線上 MSDN 教學：  
<http://www.microsoft.com/taiwan/msdn/library/>
- [6] 程式設計師俱樂部：  
<http://www.programmer-club.com/index.asp>
- [7] 國外硬體模擬相關資訊：  
<http://www.ecs.umass.edu/ece/koren/arith/simulator/>