教學用微電腦發展系統之規劃

徐弘洋

摘　要

鑑于市售微電腦發展系統價格高昂，綫路複雜，在教學應用上不僅無法普及且解說綫路原理時，困難多多。本文以價格低廉，綫路明朗化爲原則，儘可能保存基本的、常用的發展與測試功能而規劃出這套教學用微電腦發展系統。

本套系統固然以教學目標爲出發點，但對中小型微電腦產品的發展與測試工作，仍具有相當程度的克服能力。

1 引　言

微電腦產品（microprocessor-controlled product）是晚近幾年來最受矚目的科技產物，其與傳統電子產品不同處在於具備可程式化（programability）之特性，產品的動作完全由內儲存程式來控制（stored program control）。因此，發展微電腦產品所需之技術與工具迥異於傳統電子產品之開發。

自微處理機（microprocessor）推出後，發展微電腦產品所使用的發展工具（development tool）不外乎下列兩種組合：

1.由微處理機製造廠商所推出的MDS（microprocessor development system）＋ICE（in circuit emulator）。

2.傳統電子界所使用的mini 電腦＋邏輯狀態分析儀（logic state analyzer）。使用MDS或mini電腦來發展微電腦產品所需的控制程式，而產品的測試與除錯則需借助ICE或邏輯狀態分析儀的動態行爲（dynamic behaviour）監督能力了。

不論使用上述那一種組合來發展與測試微電腦產品，其共有的特色是價格高昂與本身之線路複雜。然而具有此等特點的儀表却是非常不適合教學上之使用。教學用微電腦發展工具應具備下列二點基本要求：

1.價格低廉，一人一機。如此方能達到教學目的，發揮教學效果。

2.工具本身的線路應儘量簡化，以讓學生容易學習。如此不僅可學會發展工具的使用方法，也能了解發展與測試微電腦產品的原理。

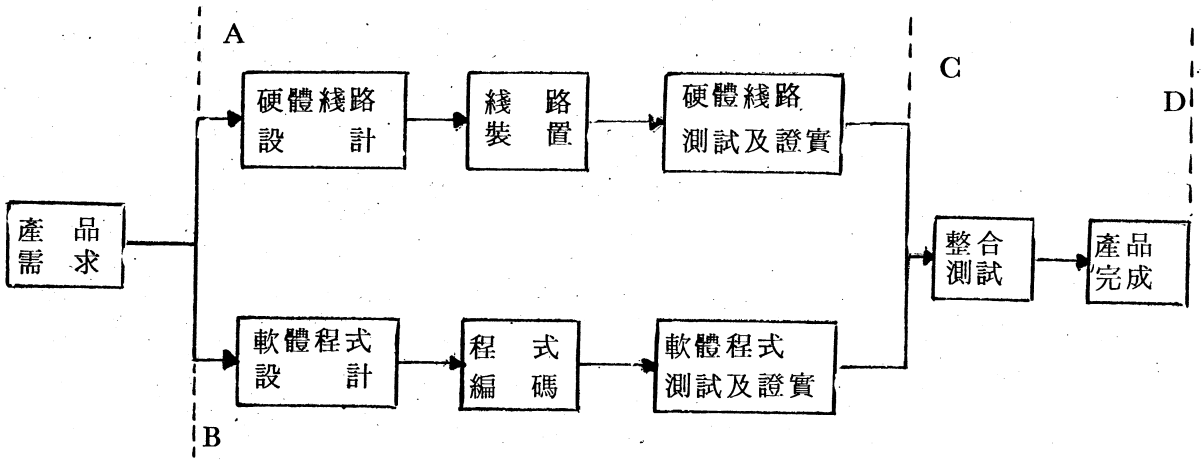
由于微電腦產品的應用領域日漸廣大，可能遭到的技術難題，故障型態均難以預知，因此了

解發展與測試的原理委實比簡單地學會操作方法來的重要。

本文首先檢視上述發展微電腦產品所使用之各種工具，如ＩＣＥ，邏輯狀態分析儀，Ｍ-ＤＳ之性能，以提出一微電腦發展工具所需具備的基本要求。然後以低價位，線路單純爲出發點規劃一套專供教學使用之微電腦發展工具。

2 現有發展工具之性能

發展微電腦產品的過程可由圖示如下：



上圖所示可劃分爲三個明顯的階段：

1. 硬體的設計與證實階段（Ａ點到Ｃ點）。
2. 軟體的設計與證實階段（Ｂ點到Ｃ點）。
3. 軟體與硬體之整合測試階段（Ｃ點到Ｄ點）。

分析以上階段的設計活動（design activity），很明顯地，第一階段的活動內容與傳統電子產品之設計過程幾乎相同，設計過程所賴的工具以示波器（scope），複用表（multimeter）等傳統設計工具也足可應付。第二階段軟體程式的發展對電子業設計人員雖是十分嶄新的工作，然而此部份的活動內容與中大型電腦軟體程式之發展測試過程十分類似。就設計所使用之工具如編校程式（editor），組繹程式（assmbler），偵錯程式（debugger）而言，兩者幾乎完全相同。

比較困難的是第三階段軟體硬體的整合測試（integration test），這項工作對電子工程師或程式設計師均是一項挑戰性之工作。在微處理機應用的萌芽初期，用以控制微電腦行爲的控制程式（control program）尚十分簡單，因此這項工作未引起多大的困擾。

然而工程師發現，經由軟體控制方式，微電腦的功能可充分發揮，可塑性（flexibility）可大爲提高，應用領域也日漸廣大，控制程式因而愈演愈變複雜，此時軟硬體整合測試的工作遂成一項困難而不可忽視的工作了。邏輯狀態分析儀及ＩＣＥ的出現純粹爲克服此項困難。微電腦產品發展的成敗關鍵固然在此，而開發微電腦產品的技術與工具之瓶頸亦在此階段

底下簡略敍述MDS，ＩＣＥ，mini電腦，邏輯狀態分析儀等所提供發展工具之性能：

編校程式（editor）：發展控制程式時所遇到的第一個發展工具。協助使用者將原始程式（Source program）存入貯存設備。一般具有增入（insert），刪除（delete），修改（modify）一行（line）或一字（character）之功能，對貯存設備亦有存取之能力。

組譯程式（assembler）：將編校程式所建立的原始程式翻譯成目標程式（object program）及相關之連結資料（linkage information）。最簡單的形式爲行組譯程式（line assembler），只允許操作碼（operation code）符號化，位址則不允許，因此可逐行譯碼。一般組譯程式則允許位址也符號化，使用起來較方便靈活。複雜的組譯程式甚至允許巨集指令（macro instruction）之使用。在開發程式時以允許巨集指令的組譯程式功能最强，但在軟硬體作整合測試時以行組譯程式的使用最爲靈活方便。

編譯程式（compiler）：發展微電腦產品不一定用組合語言，某些發展系統亦提供高階語言如ＰＬ／Ｍ，ＰＬ／Ｚ等等。編譯程式的功能爲將用高階語言所寫的程式譯解爲目標程式。

連結程式（linker）：將由組譯程式或編譯程式所譯解的目標程式單元連結成可供微處理機執行的機器碼（executable code）。利用連結程式，可將一件問題分成數個單位，分開設計，翻譯成目標程式後再將之連結成單一控制程式。

載入程式（loader）：將可執行之機器碼載入之主記憶體，以準備執行。

偵錯程式（debugger）：協助使用者監督程式之執行以利用於錯誤的發現（error detection），檢出（error location）及排除（error correction）。一般偵錯程式應具備下述功能：

1. 檢視記憶體內容。
2. 更改記憶體內容。
3. 檢視暫存器內容。
4. 更改暫存器內容。
5. 從某一位址開始執行。

6.設定程式中斷點（ break point ）。

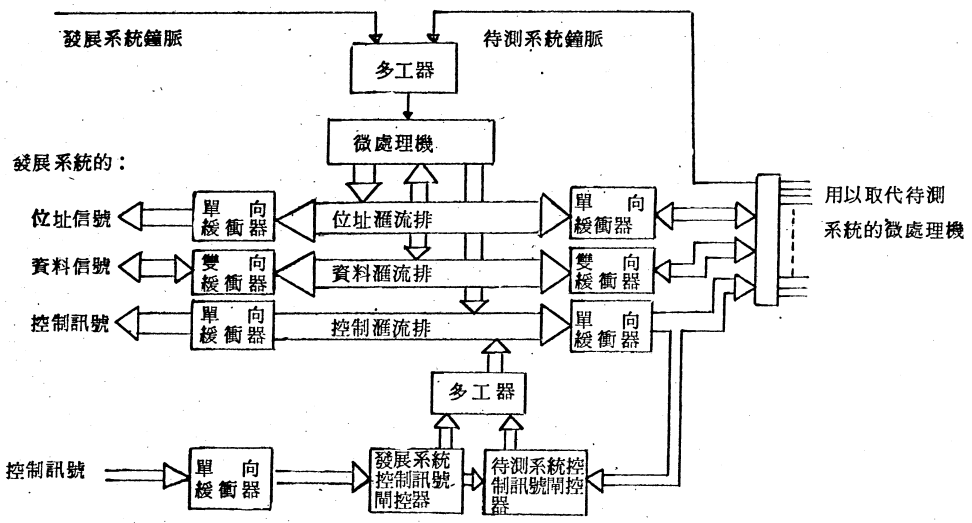
7.從輸入埠（ input ）讀取資料。

8.將資料送到輸出埠（ output port ）。

模擬程式（ simulator ）：在mini 電腦上發展微電腦控制軟體時，因中央處理單元（ＣＰＵ）無法執行微電腦之機器碼，爲了先確定程式邏輯的正確，mini電腦乃模仿微處理機的行爲，以供使用者檢視程式之執行結果及執行路徑。

邏輯狀態分析儀：微電腦產品與傳統電子產品最大不同在於微電腦產品的動態行爲完全由程式所控制，不似傳統電子產品一般在某些測試點上有著固定的波形信號在重複循環，只要監視這些點的信號即可判斷產品的正確與否。因此示波器等工具在傳統電子產品的測試上是一大利器，但在微電腦產品的測試就顯得力有未逮了。邏輯分析儀的最大功能在於能追踪微處理機的位址滙流排（ address bus ），資料滙流排（ data bus ），控制滙流排（ control bus ）各接點的信號狀態，並記錄到記憶體內達 256 或 4,096 個狀態。這些狀態可在稍後由顯示幕顯示出來以供使用者追踪程式執行之路徑以及於不同時序中各點之信號關係。

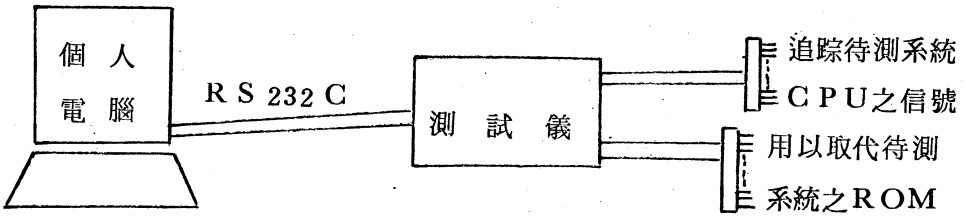
ＩＣＥ（ in circuit emulator ）：ＩＣＥ的動作原理是將待測系統（ target system ）的微處理機用一組測試綫路來取代，測試綫路能提供原有微處理機所有的信號外，並能監督及控制這些信號。ＩＣＥ 的典型結構如下圖所示：



ＩＣＥ的功能一般可分爲ＣＰＵ模擬及ＲＯＭ模擬。ＣＰＵ模擬涵括了上述偵錯程式（ debugger ）及邏輯分析儀的能力。ＲＯＭ模擬則以ＩＣＥ本身的ＲＡＭ來取代待測系統的ＲＯＭ，如此可輕易地在ＩＣＥ上更正軟體錯誤以驗證系統的正確功能，同時也易於撰寫待測系統硬體綫路的測試程式（ test program ） 以利於硬體的檢修。

3 教學用發展系統之性能

依據前節所述微電腦發展工具所提供的性能，再加上價格低廉，綫路簡單化爲主要考慮因素，所規劃設計而成的教學用微電腦發展系統之結構如下圖所示；



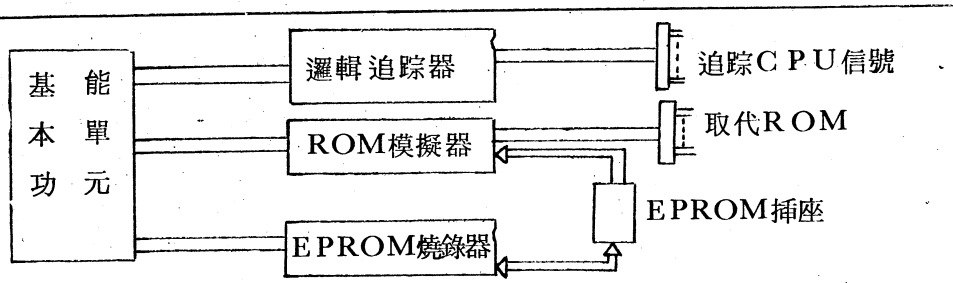
上圖可略分爲兩大部份：個人電腦及其界面，測試儀及測試夾。底下分別敍述這兩部份的功能。

個人電腦所擔負的工作主要作爲提供測試儀與使用者相互溝通所需的端末機（ terminal ），同時也提供編校程式，組譯程式，偵錯程式等發展工具，以利軟體程式之開發與測試。由於晚近幾年來，個人電腦發展蓬勃，不僅價位低廉且有豐沛之套裝程式（ package ）可資利用，採用個人電腦的理由不僅因其價位遠低於端末機，更由於上述發展工具如組譯程式等等可輕易獲得，如此可縮短教學用發展系統本身的開發時間。

測試儀與個人電腦之間採用ＲＳ 232 Ｃ界面規格。只要具有ＲＳ 232 Ｃ規格的個人電腦甚或學習機（ evaluation system） 均可使用在本系統上。現以蘋果二號（ Apple Ⅱ ）個人電腦爲例，列舉可由套裝軟體獲得及自行開發之發展工具：

1. 編校程式，套裝軟體。
2. 組譯程式，套裝軟體。
3. 偵錯程式，套裝軟體。
4. Z—80交互組譯程式（ cross assembler ），自行發展。
5. Z—80反組譯程式（ disassembler ），自行發展。
6. 模擬程式，自行發展。

測試儀的功能方塊圖可如下圖所示，分爲四個功能單元：



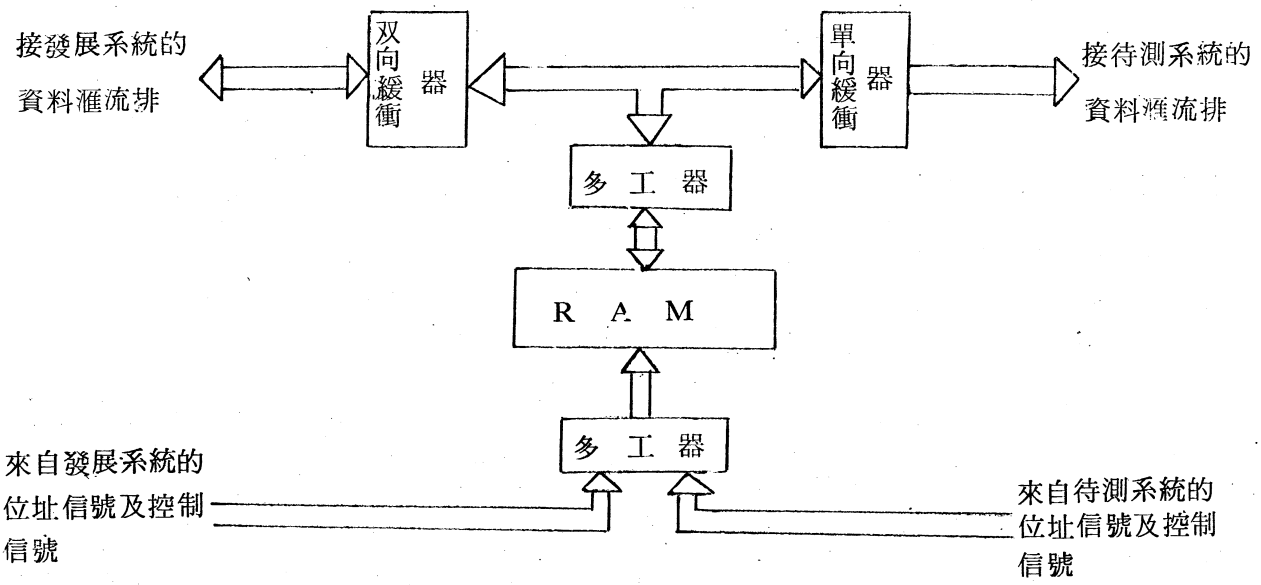
各單元所執行的功能說明如下：

1.基本功能單元（basic function unit）：這單元提供類似偵錯程式的功能，可以顯示，更改記憶體與暫存器的內容，可以設定四個中斷點及可從任一點開始執行程式。此外尚具有向上載入（upload），向下載入（download）的功能，可使測試儀與個人電腦間記憶體的內容互相交換，以利程式的測試。爲了使檢查機器碼（machine code）方便起見，6502 與Z—80微處理機的反組譯功能也包括在內。

2.EPROM燒錄器（EPROM writer）：可將測試好的程式燒錄到EPROM以供待測系統之用。除燒錄外，尚可將EPROM內容讀入系統的RAM以供檢視或修改，可檢查EPROM是否爲空白？可複製（copy）EPROM等等。能處理的EPROM包括 2716 ，2732 ，2764 等。

3.ROM模擬器（ROM emulator）：將發展系統的RAM用來取代待測系統的EPROM，包括 2716 ，2732 ，2764 ，主要是利用RAM爲可讀可寫的記憶元件。這樣的取代，至少有兩點莫大的好處。第一，不需花費額外的代價即可設計一些簡單的測試程式，以測試待測系統的硬體，這種程式主要目的在硬體上產生一些重複性的信號以供檢修或確認硬體功能之用。第二，當待測系統的軟體有錯誤時，錯誤的檢出，更正，然後再燒製一個新的EPROM，以證實錯誤是否已排除，這段過程相當費時且繁瑣，如利用ROM模擬器則可在發展系統上的RAM直接修正，直接驗證，就顯得方便省時多了。用試誤法（try and error）來檢出錯誤原因時，唯有利用ROM模擬器才有可能。（試誤法雖然是不被建議的偵錯方法，但實用上却是相當有效的方法）。

ROM模擬器的動作原理是利用双埠出入的RAM來達成的，如下圖所示。至于RAM裏面的資料則可利用向下載入（download）的功能由個人電腦的記憶體載入，或由EPROM燒錄器的EPROM直接讀入。



4.邏輯追踪器（ logic tracer ）：基本上是用來作軟硬體整合測試時檢查錯誤之來源。可連續記錄CPU的狀態達 2048 次，每一狀態所記錄的信號包括16條位址滙流排，8 條資料滙流排及 8 條控制滙流排上的信號狀態。所記錄的狀態可在記錄終了予以顯示在個人電腦的顯示幕（ monitor ） 上，顯示的格式可以是二進制的，以指示各點信號之邏輯狀態，也可以是反組譯碼的形式，以協助使用者追踪程式執行所經的路徑。

追踪器可以由使用者自行設定開始記錄的始點或終點，此卽觸發條件（ trigger ）。觸發條件可以設定在某一任選之位址上及某一執行週期（ execution cycle)如記憶體讀取週期，記憶體寫入週期，輸入埠讀取週期或輸出埠寫入週期等。

當觸發條件符合時，追踪器有下列三種記錄方式：

1.開始記錄 2048 個狀態，如此可追踪觸發條件發生後程式執行的路徑。

2.停止記錄；如此可追踪到達觸發條件前程式所執行的路徑。

3.繼續記錄 256 × n 個狀態，n 值可預先設定，可爲 1 至 7 之間任一整數。如此可追踪觸發條件前（ 8 − n ）× 256 個狀態及觸發條件後 n × 256 個狀態。

邏輯追踪器主要以偵測微電腦產品的動態行爲爲目的，在故障確認（軟體錯？硬體錯？或軟硬體交界發生錯誤之認定），故障排除方面爲一强而有力的工具。

4 結　論

這系統雖然以教學目的爲出發點而規劃設計的發展系統，但仍具有相當程度的實用價值。微電腦產品的時序若非十分嚴格（ critical ），控制軟體的邏輯若非十分複雜，則利用

這一套教學用微電腦發展系統就足可應付各種發展、測試所遭遇的問題了。

5 感　謝

本系統之研製成功，蒙本校電子科七十級黃天進、林培元、吳進安、簡金堆、詹惠傑、尋民智等六位同學助力甚多，在此申致謝之忱。

參考書目

1. 〃 Microprocessor Development and Development Systems 〃
McGRAW — HILL Book Company Limited
Edited by Vincent Tseng

2. 〃 Practical Troubleshooting Techniques For Microprocessor Systems 〃
Prentice — Hall, Inc.
James W. Coffron

3. 〃 Practical Hardware Detail 8080 , 8085 , Z 80 , and 6800 Microprocessor Systems 〃
Prentio — Hall , Inc.
James W. Coffron

4. 〃 Troubleshooting Microprocessors & Digital Logic 〃
Robert L . Goodman