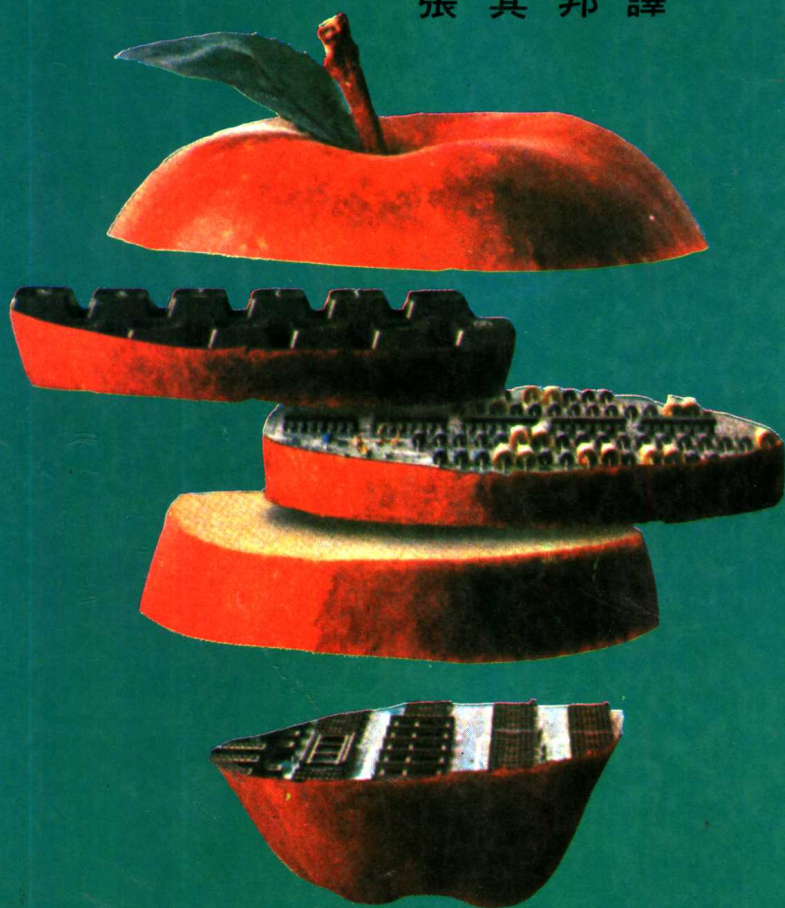


APPLE II

組合語言程式設計

張其邦譯



香港科技出版社

512439

組合語言程式設計

張其邦譯

香港科技出版社

APPLE II 組合語言程式設計

編譯者：張 其 邦

出版者：香 港 科 技 出 版 社

發行者：香 港 科 技 出 版 社

九龍彩虹道810號六樓

印刷者：永達印刷公司

香港黃竹坑建明工業大廈九樓D座

定 價：H. K. \$

序 言

歡迎加入我們的組合語言世界，在這裏你的字彙將限於 56 種三個字母的字——譬如像 LDA, STX, EOR 和 PLA 等等，用這些字來表示一些動作。你將會訝異於你可以這有限的幾個字做很多的事。你可以設計一個計算器，在你還不知道怎麼回事時，一個複雜方程式的解已被很快地被求出了；或是你也可以令微電腦去處理一些人類無法忍受的冗長、繁複的事務。就是這寥寥 56 個字的組合語言，再配合上微電腦，使得人們風靡於其強大的性能與無遠弗屆的用途。

作者在這裏並不是想要減損高階程式語言在使用上的方便特性，而是想藉著學習組合語言，使你更熟悉接近微處理器，清楚得如同自己親自製作的一樣。這會使你有更好的基礎來學習一般的計算機系統。也許某些年輕的讀者，會在研究微處理機中，突然觸發他的靈感，去更深入地研究符號邏輯、數位電子學、量子力學等科學。

希望你會喜歡學習 6502 組合語言的設計，以愉快的心情去讀，跳過較難的地方。也不需要以螢光筆去劃綫、做筆記、去記憶，或是加上任何箝制我們歡愉之心的枷鎖。本書之後將不附有測驗試題。

話說回來，本書也有使你成為一個勝任的組合語言程式師的嚴肅一面。如果你的目的是如此的話，你必須另加上一

種更慎重的學習態度：你必須花些時間利用你的個人電腦，以書上所提供的程式或你自己設計的程式，來練習書上的習題；而不是紙上談兵就可學會編寫程式的，你必須設計、編寫、執行、再除錯過許多程式之後，才能精通。你的個人電腦當然是一個重要的資源，但最寶貴的還是你的想像力與創造力。這樣的學習態度雖較嚴肅，但也不會減低你讀書的快樂心情。讓你的好奇心成為你創新的泉源，不要患得患失，如此學習一個新的程式語言的過程就會使你興趣盎然。

因為組合語言程式師比高階語言程式師更接近硬體部分，所以本書包含了一些基本的界面技術。如果你覺得硬體部分太難，可以跳過不讀，因為還有很多不需要積體電路、電路圖和配綫等基礎的挑戰性問題等著你。

目 錄

第 0 章	6502 組合語言指令集	1
	CPU 暫存器與狀態旗號	4
	6502 記憶體定址模式	7
	6502 指令集	18
第 1 章	微電腦系統	123
	宗 旨	123
	簡 介	123
	微電腦系統的元件	124
	記憶體空間和記憶體結構	126
	讀寫操作	132
	輸出入的基本概念	136
	APPLE 監督程式	140
第 2 章	編寫且執行簡單的組合語言程式	145
	宗 旨	145
	簡 介	145
	微電腦的指令	146

位址型態.....	154
微電腦程式.....	156
編寫一組合語言程式.....	157
JMP 和 BRK 指令.....	161
資料轉移指令的示範.....	163
第3章 分支和迴路	171
宗 旨.....	171
簡 介.....	171
處理機狀態暫存器.....	172
分支指令.....	176
副程式之使用.....	180
從 APPLE 的鍵盤讀入資料.....	181
增加 (Increment) 和減少 (Decrement)	
指令.....	183
定時迴路.....	185
符號問題.....	188
定時迴路和遊戲操作桿.....	190
定時迴路和揚聲器輸出.....	194
開關之讀入.....	195
第4章 邏輯運算和移動，旋轉運算	205
宗 旨.....	205
簡 介.....	205

邏輯運算.....	206
6502 中的邏輯指令	208
邏輯指令的用途.....	212
旋轉和移位指令.....	217
第5章 算術運算	231
宗 旨.....	231
簡 介.....	231
旗標修改指令.....	232
AND 指令	232
SBC 指令	237
十進位型態的算術.....	239
比較指令.....	241
4 位元乘法的程式.....	242
8 位元的乘法程式.....	246
除 法	247
ASCII 到 16 進位的轉換	249
第6章 位址型態：索引位址法	259
宗 旨.....	259
簡 介.....	259
絕對索引位址法.....	262
第 0 頁索引位址法.....	266
範例程式.....	268
4 位元組減法的程式.....	268

十進位型態的多精密度加法程式·····	269
找出列表中最大數的程式·····	270
三位元組數的增加程式·····	271
取三位元組數的補數的程式·····	272
將一 4 位元組數移位旋轉到另一個位置的程 式·····	273
3 位元組的 BCD 數的增加程式 ·····	274
檢查和的程式·····	274
ASCII 轉換成摩斯碼的程式·····	276
間接索引位址法·····	279
索引的間接位址法·····	291

第 7 章 副程式，堆疊和中斷····· 299

宗 旨·····	299
簡 介·····	299
副程式·····	301
堆 疊·····	303
堆疊的操作和用法·····	305
中 斷·····	310
APPLE II 處理中斷的結構·····	318
由中斷來控制的鍵盤輸入·····	322
24 小時的鐘 ·····	327
BASIC 和 6502 組合語言共用的方法 ·····	330

第8章 編寫程式的一些問題 341

簡 介 341

6502 位址型態的彙總 341

將 7 位元的字元碼加上同位元 345

二進位到 BCD 與 BCD 到二進位之轉換 347

N 個位元組的二進位的 BCD 的轉換常式 352

BCD 到 ASCII 的轉換常式 352

N 位元組 BCD 到二進位數的轉換 356

有號數目的算術 358

第9章 利用6522來設計程式 365

宗 旨 365

導 論 365

輸出入元件的回顧 368

6522 簡單的操作 371

更複雜的 I/O 動作：列表機界面 378

以 6522 來計時和計數 389

使用 T1 的自由運行狀態工作方式 398

利用 T2 來計算脈衝的次數 403

利用 T1 和 T2 來產生較長的延遲 405

精確的計時器 407

精確計時器的應用 414

第10章 應用	427
簡 介	427
傳送摩斯碼的程式	427
接收摩斯碼的常式	445
一個利用類比 / 數位轉換器的實例	448
數位 / 類比 (D/A) 轉換器界面的設計	458
附錄A 十進位、二進位與十六進位數字系統	469
宗 旨	469
簡 介	469
數	470
十進位數	470
二進位數	470
位元，位元組與尼波	473
十六進位數	474
附錄B 可附加的線路與程式	481
單一步驟的線路與程式	481
電腦輔助學習程式 (CAL)	489
附錄C 一些 IC 的接腳圖	493

第 0 章

6502組合語言指令集

在本章中，我們將詳細介紹 6502 組合語言中各個指令的用法。每個指令我們都儘可能地舉例說明並且附上詳細的執行過程圖，以幫助讀者能夠更清楚地了解各個指令。

在介紹 6502 各個指令的詳細用法之前，我們先將 6502 指令集大概地介紹一下。我們將這些指令分成三類：經常使用的（表 2-1）、偶而使用的（表 2-2）、和不常使用的（表 2-3）。這樣區分主要是為了便於初學者的學習。如果您是一位極具經驗的組合語言程式設計師，那麼這樣區分並沒有什麼意義；但是，如果您是一位新手，那麼建議您先練習以表 2-1 的常用指令來寫程式，等到已經熟悉了組合語言程式設計的觀念以後，再練習其他的指令並且適當地加以應用。

2 APPLE II 組合語言程式設計

表 2-1 經常使用的 6502 指令

指 令 碼	意 義
ADC	Add with Carry
AND	Logical AND
ASL	Arithmetic Shift Left
BCC	Branch if Carry Clear
BCS	Branch if Carry Set
BEQ	Branch if Equal to Zero ($Z = 1$)
BMI	Branch if Minus ($S = 1$)
BNE	Branch if Not Equal to Zero ($Z = 0$)
BPL	Branch if Plus ($S = 0$)
CMP	Compare Accumulator to Memory
DEC	Decrement (by 1)
DEX (DEY)	Decrement Index Register X (Y) by 1
INC	Increment (by 1)
INX (INY)	Increment Index Register X (Y) by 1
JMP	Jump to New Location
JSR	Jump to Subroutine
LDA	Load Accumulator
LDX (LDY)	Load Index Register X (Y)
LSR	Logical Shift Right
PHA	Push Accumulator onto Stack
PLA	Pull Accumulator from Stack
ROL	Rotate Left through Carry
ROR	Rotate Right through Carry
RTS	Return from Subroutine
SBC	Subtract with Borrow
STA	Store Accumulator
STX (STY)	Store Index Register X (Y)

表 2-2 偶而使用的 6502 指令

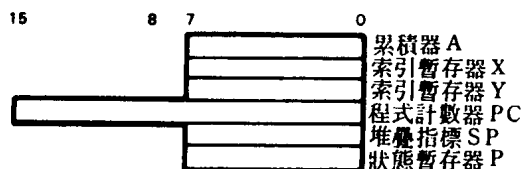
指 令 碼	意 義
.BIT	Bit Test
BRK	Break
CLC	Clear Carry
CLD	Clear Decimal Mode
CLI	Clear Interrupt Mask (Enable Interrupts)
CPX (CPY)	Compare with Index Register X (Y)
EOR	Logical Exclusive-OR
NOP	No Operation
ORA	Logical (Inclusive) OR
RTI	Return from Interrupt
SEC	Set Carry
SED	Set Decimal Mode
SEI	Set Interrupt Mask (Disable Interrupts)
TAX (TAY)	Transfer Accumulator to Index Register X (Y)
TXA (TYA)	Transfer Index Register X (Y) to Accumulator

表 2-3 不常使用的 6502 指令

指 令 碼	意 義
BVC	Branch if Overflow Clear
BVS	Branch if Overflow Set
CLV	Clear Overflow
PHP	Push Status Register onto Stack
PLP	Pull Status Register from Stack
TSX	Transfer Stack Pointer to Index Register X
TXS	Transfer Index Register X to Stack Pointer

CPU暫存器與狀態旗號

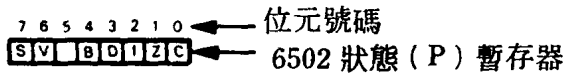
6502 微處理機包含有：一個累積器 (Accumulator)、一個狀態暫存器 (Status register)、二個索引暫存器 (index register)、一個堆疊指標 (Stack Pointer)、及一個程式計數器 (Program Counter)。這些暫存器可以將其圖示如下：



6502 的狀態暫存器包含了六個狀態旗號和一個中斷 (interrupt) 控制位元。這六個狀態旗號分別是：

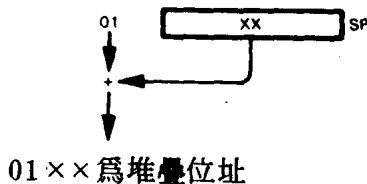
- 進位 (C)
- 零值 (Z)
- 溢位 (V)
- 符號 (S)
- 十進位模式 (D)
- 中止 (B)

各旗號在狀態暫存器中所對應的位元位置為：



索引暫存器 (X 與 Y) 的長度僅為 8 個位元，與一般典型微電腦的索引暫存器不同。

在 6502 的記憶體內有一個堆疊區 (stack area)，此堆疊區由一個長度為 8 個位元的堆疊指標指向，所以 6502 堆疊區的最大長度為 256 個位元組。此 CPU 固定在堆疊位址 (stack address) 的高次位元組 (high-order byte) 補上 01_{16} ，所以記憶體位址從 0100_{16} 到 $01FF_{16}$ 永久指定為堆疊區：



如果您視 6502 為一個獨立的產品，那麼 256 個位元組的堆疊區雖然短了些但並不會給您帶來任何的不方便。事實上，對於任何典型的微電腦應用而言，256 個位元組的堆疊區已經很夠用了。由於其位址是在記憶體的前頭，所以記憶體位址低的地方必須設計為讀 / 寫記憶體 (read/write memory)。為了避免與符號狀態旗號 "S" 混淆起見，我們以 "SP" 代表堆疊指標。

6502 程式計數器與一般微處理機的程式計數器完全相同。進位狀態旗號 "C" 是用來存放任何算術運算後最高位元的進位值

6 APPLE II組合語言程式設計

。在移位 (shift) 與旋轉 (rotate) 指令執行時，進位旗號也包括在運作之內。6502 進位旗號唯一一項不尋常的特性是：在減法運算時，其意義與一般情況正好相反。執行 SBC 指令之後，如果需要借位 (borrow) 則進位旗號會被清除為零；如果不需要借位則進位旗號被設定為 1。注意 SBC (聯合進位旗號的減法，Subtract with Carry) 指令的運算結果為：

$$(A) = (A) - (M) - (1 - C)$$

其中 M 為其他運算元，此種用法與目前大多數的微處理機不同，您必須格外注意。

6502 的零值狀態旗號與一般微處理機的相同。當任何算術或邏輯運算之後得到的結果為零時，此旗號即被設定為 1。若任何算術或邏輯運算之後得到的結果不為零，則此旗號即被設定為 0。

6502 的符號狀態旗號與一般微處理機的相同。此旗號為任何算術或邏輯運算之後所得到之結果的最高效位元。因此，若符號狀態旗號為 "1" 則表示結果為負值；若為 "0" 則表示結果為正值。如果您使用帶符號的二進位算術 (signed binary arithmetic) 運算，則此旗號會按運算後的結果而設定為 "1" 或清除為 "0"。假如您不是使用帶符號的二進位算術運算，那麼您可以不管此旗號；不過，您也可以用它來驗證運算後的結果的最高效位元值。

當設定十進位模式狀態旗號為 "1" 時，會使得 ADC 與 SBC 兩指令執行 BCD 運算。因此，當十進位模式狀態旗號被設定為 1 且執行 ADC 或 SBC 指令時，CPU 假設原來的兩個 8 位元值均為