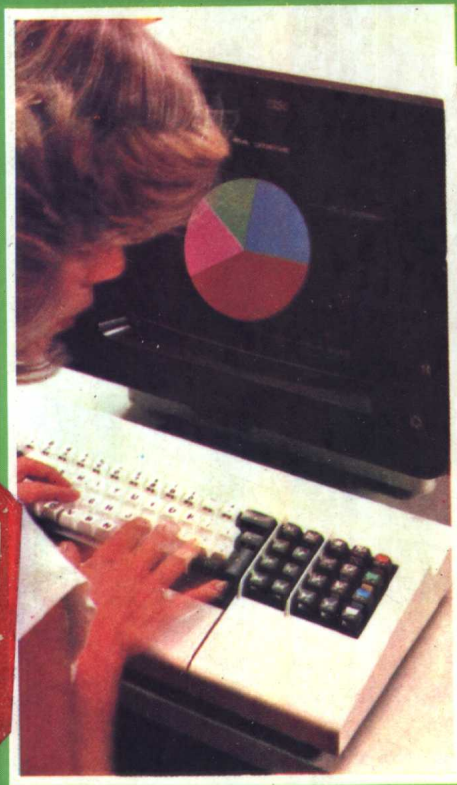
AIM 65 實驗手冊

與

研讀指引

洪欽銘 譯
陳燦輝



協群科技出版社

AIM 65 實驗手冊

與

研讀指引

洪欽銘・陳燦輝 譯

協群科技出版社

AIM 65 實驗手冊與研讀指引

編譯者：洪欽銘　陳燦輝

出　版：協群科技出版社

發　行：協群科技出版社

香港中環皇利街684號二樓

印刷者：廣源印務局

青山道875號工廠大廈

定價：H.K.$ 25.00

序

經由本書的各個實驗，對奇妙的微電腦世界，你將得到一紮實的認識。由於你將使用爲教學而設計的專業性微電腦——洛克菲勒公司的高級交談式微電腦AIM65 ，因此實驗時非常易於學習。

AIM65 微電腦的“頭腦”是一片6502微處理機，6502可能是目前市面上最高級的微處理機。6502也被使用於許多常用的微電腦中，這裡面包括 Apple Ⅱ及Ⅲ，Commodore Pet，Atari 400及800 ，以及 Synertek SYM-1，同樣的，6502也被使用工業與消費的各種產品中。

這些實驗所介紹的原理，其重要性遠超過AIM65 微電腦。在這些實驗中，你將學習6502 微處理機的內部結構與指令集、程式規劃的各種技巧（包括如何偵錯以及設計高效率程式）、外部設備如何與微電腦溝通、以及許多有趣且有價值的專題。

實驗內容

這本手册的實驗大多包含四個部份，它們依序爲：

目的——由這個實驗，你期望學到什麼。

實驗前的準備——在實驗進行前，你必需先研讀的有關書籍。

討論——實驗有關知識之研討，以及應用於解決實驗中各問題的原理說明。

步驟————步一步地進行，以解決實驗的問題。

目　　錄

1

瞭解 AIM 65

* 目　　的 *

熟悉AIM65，並學習如何輸入與執行一程式

* 實驗前的準備 *

研讀AIM65 User's Guide 1.1,1.2, 1.9, 2.1到2.8到及3.1到3.6節。研讀R6500 Programming Manual第1章及第2章2.0, 2.1節。

* 討　　論 *

在實驗期間你將使用AIM65，AIM65即Rockwell International公司製造的R6500 Advanced Interactive Microcomputer。在進行第一個實驗前，讓我們花點時間來檢查AIM 65。

AIM65是一部完整的微電腦，它具有一個打字機形式的鍵盤，一20字的顯示器，一20欄的印字機以及一些電子元件（積體電路晶片，電阻與電容）在這個實驗，以及隨後的各實體中，你將使用鍵盤將資料與指令輸入到電腦內。顯示器與印字機爲輸

出裝置——它們將程式與資料顯現出來。

AIM65 有三個開關，S1 開關，位於印字機的左下方，RESET 按鈕，它用來啟始（ initialize ）AIM65，使 AIM65 處於一已知狀態。

開關 S2，位於顯示器的左方，爲一 2 位置 SEPT／RUN 開關，於實驗 1，開關 S2 應撥至 SEPT 位置。

開關 S3，爲一 2 位置 TTY／KB 開關，用來選擇使用 AIM65 鍵盤（ KB ）輸入或外部電傳打字機輸入。在這本實驗手册之每一實驗，開關 S3 應置於 KB 位置。

AIM 65 監督程式（ Monitor ）

AIM65 的作業程式（ Operating program ）包含在兩塊 ROM 內，這兩塊 ROM 被放在鍵盤的 PRINT 按鍵上方。此程式被稱爲監督程式（ Monitor ）。Monitor 依你輸入鍵盤之命令，控制整個 AIM65 之操作。這些命令包含檢查特定記憶位址與暫存器之內涵（如果需要，亦可更改內涵），啟始程式之執行，控制印字機爲 ON 或 OFF，以及其他各種系統控制功能。Monitor 之命令在 AIM65 User's Guide 第 3 章中有詳盡之說明，同時 AIM65 Summary Card 亦將其列出。

幾乎所有的 Monitor 命令在此實驗手册中均被使用到，但於實驗 1，我們只使用了下列 8 個命令：

命令	說明
*	改變程式計數器
I	輸入助憶指令
RESET	進入和啟始 Monitor
G	開始執行使用者程式

K	記憶體內涵解組合
M	顯示特定記憶體內涵
/	改變目前記憶體內涵
R	顯示暫存器內涵

R6502 指令集

微電腦（microcomputer）是由微處理機（microprocessor，通常亦稱之爲CPU，或中央處理單元）、記憶體以及一個或更多的輸入／輸出（I/O）裝置組成。於AIM65微電腦內，微處理機爲 Rockwell R6502，R6502 爲一8位元之微處理機，也就是說它一次祇能執行8位元之指令與資料操作。R6502由記憶體內一個接一個提取程式內之指令以執行程式。

有些指令可以放在一個記憶位置內（含8位元，或1位元組），其他指令因包含運算元（資料或記憶位置），因此必須佔兩個或三個記憶位置。所有指令的第一位元組爲運算碼，或稱OP-Code，它告訴R6502 那一個操作要被執行。

R6502 執行的所有指令，稱爲R6502 指令集。R6502指令集共有56個指令，它們被整理於表1-1及1-2，除此之外，56個指令內有許多指令操作於一種以上之定址模式。R6502 共有13種定址模式，此使得R6502 成爲目前最通用且功能最強之微處理機。定址模式在R6500 Programming Manual一書中討論得很詳細。

對AIM 65輸入程式時，你將輸入指令的組合語言（assembly language）形式，亦即使用表1-2最左邊那一欄，3個英文字母組成助憶碼。AIM65的 Monitor 自助地將每一個助憶碼翻譯成OP-Code，以輸入到R6502。對於可以操作於一種以上

上之定址模式的指令，Monitor 將依指令之運算元，而產生適當的OP-Code 。表1-3指出R6502 之13種定址模式，其運算元之一般格式。表中每一個"a"代表一16進位數（0到0及A到F），由此表知，假如你使用立即定址法指定運算元爲16運制之F3，你就要輸入＃F3於運算元位置，例如，要存F3進入累積器，你應輸入指令LDA＃F3。注意，除了隱含定址模式外所有定址模式均需運算元。

表1-1 R6502 指令助憶碼

ADC Add Memory to Accumulator with Carry
AND "AND" Memory with Accumulator
ASL Shift left One Bit (Memory or Accumulator)

BCC Branch on Carry Clear
BCS Branch on Carry Set
BEQ Branch on Result Zero
BIT Test Bits in Memory with Accumulator
BMI Branch on Result Minus
BNE Branch on Result not Zero
BPL Branch on Result Plus
BRK Force Break
BVC Branch on Overflow Clear
BVS Branch on Overflow Set

CLC Clear Carry Flag
CLD Clear Decimal Mode
CLI Clear Interrupt Disable Bit
CLV Clear Overflow Flag
CMP Compare Memory and Accumulator
CPX Compare Memory and Index X
CPY Compare Memory and Index Y

DEC Decrement Memory by One
DEX Decrement Index X by One
DEY Decrement Index Y by One

EOR "Exclusive-or" Memory with Accumulator

INC Increment Memory by One
INX Increment Index X by One
INY Increment Index Y by One

JMP Jump to New Location
JSR Jump to New Location Saving Return Address

LDA Load Accumulator with Memory
LDX Load Index X with Memory
LDY Load Index Y with Memory
LSR Shift One Bit Right (Memory or Accumulator)

NOP No Operation

ORA "OR" Memory with Accumulator

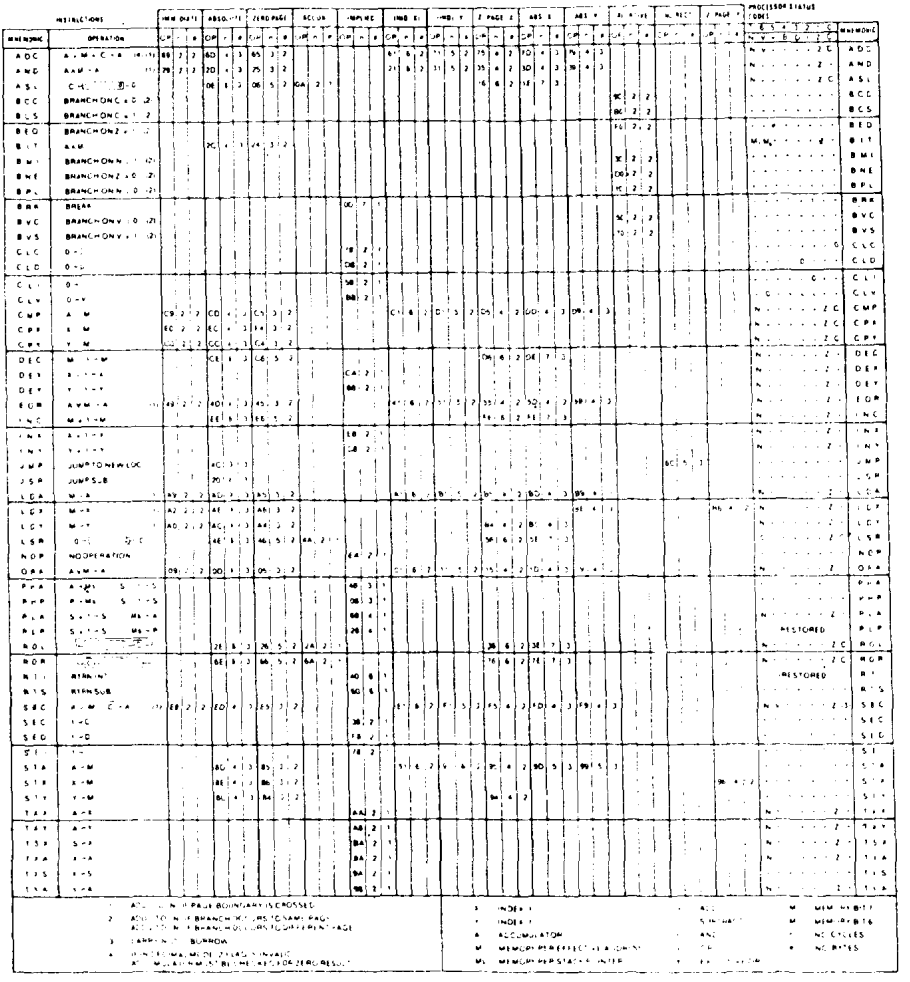
PHA Push Accumulator on Stack
PHP Push Processor Status on Stack
PLA Pull Accumulator from Stack
PLP Pull Processor Status from Stack

ROL Rotate One Bit Left (Memory or Accumulator)
ROR Rotate One Bit Right (Memory or Accumulator)
RTI Return from Interrupt
RTS Return from Subroutine

SBC Subtract Memory from Accumulator with Borrow
SEC Set Carry Flag
SED Set Decimal Mode
SEI Set Interrupt Disable Status
STA Store Accumulator in Memory
STX Store Index X in Memory
STY Store Index Y in Memory

TAX Transfer Accumulator to Index X
TAY Transfer Accumulator to Index Y
TSX Transfer Stack Pointer to Index X
TXA Transfer Index X to Accumulator
TXS Transfer Index X to Stack Register
TYA Transfer Index Y to Accumulator

表 1-2 R6502 指令集



16 進制數之表示法

在這個實驗，以及此手册中之大部份的實驗，我們將使用AIM65助憶指令輸入模式，將程式輸入到微電腦之記憶體。助憶指令輸入模式將所有資料及位址視爲16進制之數目。爲了避免寫成“16進制數F3”或“hex F3”，此後我們將使用一"$"符號於數目前，以表示此數目爲16進制數；亦即，此手册中，16進制數F3，將寫成$F3。（'$'符號，於助憶指令輸入模式時不可使用。）

研究題目（Assignment）

寫一程式將$AC 值載入累積器內，然後將累積器之內涵存入位址$40。此程式開始於位置$0200。要這樣作，你應該使用下列二指令：

指　令	說　明
LDA	Load Accumulator with Memory
STA	Store Accumulator in Memory

表1-3　6502 定址模式

模　式	運算元格式 t
立即	#aa
絕對	aaaa
零頁	aa
隱含	
間接絕對	(aaaa)
絕對指標，X	aaaa,X or aaaaX
絕對指標，Y	aaaa,Y or aaaaY
零頁指標，X	aa,X or aaX
零頁指標，Y	aa,Y or aaY
指標間接	(aa,X) or (aaX)
間接指標	(aa),Y or (aa)Y
相對	aa or aaaa
累積器	A

* 實驗步驟 *

1. 打開你的AIM65 電源，在此“啟始”狀態，程式計數器與累積器的內涵是什麼？
   PC=　　A=

2. 在步驟1中，你使用監視器（Monitor）的那一個命令，來決定暫存器的值？

3. 記憶位置 $40 目前之內涵爲什麼？
   位置 $40 之內涵爲：

4. 在步驟3中，你使用的那一個命令來決定位置 $40 之內涵。

5. 寫一2指令之程式，將 $AC 值載入累積器，然後將累積器之內涵存於位置 $40。

6. 你的載入指令，是使用那一種定址模式？

7. 你的存入指令，是使用那一種定址模式？

8. 改變程式計數器內涵，使它指到你程式的起始位址 $0200。你使用Monitor 的那一個命令將 $0200 載入程式計數器？

9. 置AIM65 於助憶指令輸入模式。你使用Monitor 的那一個

命令來作。

10. 現在顯示器顯示什麼？

顯示器顯示的內容指出什麼？

11. 將步驟 5 中之程式輸入電腦，輸入後按 ESC 鍵。

12. 程式計數器現在的內涵是什麼？

解釋爲什麼程式設計器爲這個值。

13. 改變程式計數器，使它指到你程式的起啟位址，然後執行程式。你使用Monitor 的那一個命令來執行程式。

14. 此刻，你所期望程式計數器、累積器與記憶位置 $40之內涵是什麼？

PC=　　A=　　$40=

使用Monitor 內適當的命令，來證明你的答案。

15. 假如你以最有效率之方法規劃程式，你的 2 指令的程式應佔住記憶體的 4 位元組，從位址 $0200 到 $0203 ，列出這 4 個位址之內涵，並對每一位址之內涵提供一簡短說明：

位 置	16 進制內涵	內涵之說明
$0200		
$0201		
$0202		
$0203		

2

加 運 算

* 目 的 *

學習如何執行有號數以及無號數的加運算。

* 實驗前的準備 *

研讀AIM65 User's Guide 6.2 節與6.8 節，以及R6500 Programming Manual 的 2.2.1 節與第3章。

* 討 論 *

R6500 指令集只有一個加的指令ADC，ADC 表示 將記 憶體之內涵連同進位旗標C 累加到累積器A中（A中爲A+ M+C）。這個指令將某一特定記憶位置之內涵與狀態暫存器進位旗標C，累加到累積器中中。ADC 指令可用來對下列三種整數執行加的運算：有符號二進數、無符號二進數，以及二進碼十進數(Binary -Coded - Decimal number)

加運算時之狀態旗標：

ADC 指令影響處理機狀態暫存器中四個旗標

- 假如二進數相加後，和超過 255_{10}（FF_{16}）；或二進碼十進數（BCD）相加後，和超過99，那麼進位旗標C將被設定爲1，否則連位旗標爲0。
- 假如和爲零，則零旗標Z將設定爲1，否則爲0。
- 假如和的位元7爲1，則符號旗標N被設定爲1，否則被設定爲0。祇在有號數相加時，N旗標才有意義，在這種情形下，N旗標指出運算結果爲負值（N=1）或正值（N=0）
- 假如兩同號數（同爲正或同爲負）相加，其和超過 $+127_{10}$（$7F_{16}$）或 -128_{10}（80_{16}），則溢位旗標V設定爲1，此將使累積器 位元7的值改變（0變爲1或1變爲0）；若和未超過，則V爲0。

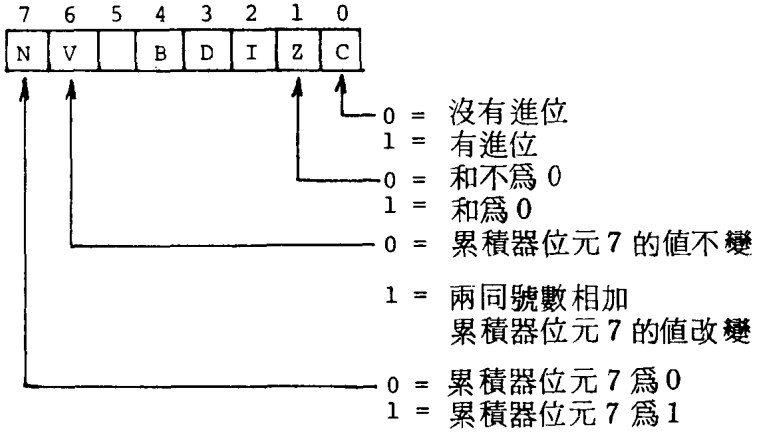


圖 2-1 加運算影響的狀態旗標

執行加運算時所使用的R6502 指令

在這個實驗中，你將使用下列的R6502 指令