

RPGII 的 邏 輯

張 誠 淑 編 譯

RRRRRRRRRR	PPPPPPPPPP	GGGGGGGGGG	IIIIII	IIIIII
RRRRRRRRRR	PPPPPPPPPP	GGGGGGGGGG	IIIIII	IIIIII
RRE RRE	PPP PPP	GGG GGG	III	III
RRE RRE	PPP PPP	GGG	III	III
RRRRRRRRRR	PPPPPPPPPP	GGG GGGGG	III	III
RRRRRRRRRR	PPPPPPPPPP	GGG GGGGG	III	III
RRRRR	PPP	GGG GGG	III	III
RRR RRR	PPP	GGGGGGGGGG	III	III
RRR RRR	PPP	GGGGGGGGGG	IIIIII	IIIIII
RRR RRR	PPP	GGG	IIIIII	IIIIII

復 文 書 局

TP331
10

*RPG*II的邏輯

張誠淑 編譯

復文書局

WES0067

RPG II 的邏輯

著作權執照台內著字第 號

版權所有



翻印必究

中華民國六十七年六月初版發行

中華民國六十九年十月修訂再版

中華民國七十一年五月修訂再版

平裝特價 132 元 精裝特價 172 元

著 作 者：

編 譯 者： 張 誠 淑

發 行 者： 吳 主 和

發 行 所： 瓊 文 書 局

地址：臺南市東門路421巷28號

門市部地址：台南市林森路二段 63 號

電話：(062)370003 · 386937

郵政劃撥帳戶 32104 號

No.28. LANE421 DONG-MEN
ROAD TAINAN TAIWAN REPUBLIC
OF CHINA

TEL: (062)370003 · 386937

本書局經行政院新聞局核准登記發給
出版事業登記證局版台業字第0370號

引 言

本書是程式設計的指引，與一般程式設計參考手冊不同。有關 RPG I 程式設計的論題都按照它們在資料處理工作中標準的用法，並儘可能舉例說明。和參考手冊不同的是本書中每章包含一個單元。但是，如果我們希望瞭解某一特殊的論題，可以直接由目錄找出討論該論題的章節。

雖然各章都是一個完整的單元，但自第一章至第六章在理論上是一章一章推進的，因此，希望讀者能逐章閱讀。若已學過 RPG I 程式設計的基本課程，就可以不必逐章閱讀第一至第六章。

爲了易於說明起見，本書中許多例子都以一張卡片代表一筆資料，這並不表示在這些情形裡讀，打卡機必需用來做爲輸入或輸出的裝置。我們可依照所使用的機型而決定任何一種適當的輸入／輸出裝置。

目 錄

第一章 RPG II 的邏輯

§ 1-1	資料處理的基本邏輯	2
§ 1-2	RPG II 的基本邏輯	4
§ 1-2-1	在 RPG II 基本邏輯裡的特殊步驟	6
§ 1-2-2	第一個程式週期	7
§ 1-2-3	RPG II 基本邏輯摘要	13
§ 1-3	RPG II 邏輯與指示器的關係	13
§ 1-3-1	1P (第一頁) 指示器	13
§ 1-3-2	最後資料卡指示器 (LR)	17
§ 1-3-3	資料鑑定指示器 (01-99)	20
§ 1-3-4	分欄指示器 (01-99)	26
§ 1-3-5	結果指示器 (01-99)	32
§ 1-3-6	停止指示器 (H1-H9)	36
§ 1-3-7	頁底指示器 (0A-0G ; 0V)	43
§ 1-3-8	配對資料卡指示器 (MR)	44
§ 1-3-9	各種指示器之設置	45

第二章 輸入表格的填法與用法

§ 2-1	控制分欄	46
§ 2-1-1	控制分欄的填法	48
§ 2-1-2	分離的控制分欄	49
§ 2-1-3	分離控制分欄的填法	52
§ 2-2	檢查資料的順序	52
§ 2-2-1	資料的順序	52
§ 2-2-2	檢查同組資料之順序	56
§ 2-2-3	同組中之錯誤資料	56

§ 2-2-4	同組中有序和無序的資料型	59
§ 2-2-5	一組資料中無用或不希望有的資料	59
§ 2-3	分欄歸屬關係指示器	60
§ 2-3-1	OR 關係	61
§ 2-3-2	OR 關係與分欄歸屬關係	63
§ 2-3-3	分欄歸屬關係與控制分欄	63
§ 2-3-4	分欄歸屬關係與分離的控制分欄	64
§ 2-4	有條件的用到輸入資料束(外部指示器)	64
§ 2-4-1	利用一個程式做一個以上的工作	65
§ 2-4-2	完全處理所有資料束之前結束程式的執行	69

第三章 報表輸出的控制

§ 3-1	利用 OVERFLOW 和 FETCH OVERFLOW 控制報表的跳頁	72
§ 3-1-1	頁底指示器	72
§ 3-1-2	應用頁底指示器的描述	73
§ 3-1-3	防止資料跨印於兩頁報表上	76
§ 3-1-4	強迫跳頁(FETCH OVERFLOW)	80
§ 3-2	報表表格的調整	82
§ 3-3	編輯	83
§ 3-3-1	編輯方法	83
§ 3-3-2	編輯與終點位置	90
§ 3-4	利用 *PLACE 重複印出資料	91
§ 3-4-1	*PLACE 之填法	94
§ 3-4-2	印表的順序	94
§ 3-4-3	同一行上重複多次印一分欄	97

第四章 卡片的輸出

§ 4-1	打卡和印卡	99
§ 4-1-1	打卡	99
§ 4-1-2	印卡	99
§ 4-2	利用一資料同時做為輸入和輸出	106
§ 4-2-1	打資料於讀入之卡片上	107
§ 4-2-2	打卡於資料束中之空白卡	109
§ 4-2-3	如何決定混合資料束	112

§ 4-3	選擇卡片槽.....	113
§ 4-3-1	輸入和混合資料束卡片.....	113
§ 4-3-2	輸出資料束卡片選擇卡片槽.....	116
§ 4-3-3	合併輸入及輸出資料束卡片.....	119

第五章 在RPG II程式中之控制作業

§ 5-1	控制計算和輸出之指示器的附加用法.....	121
§ 5-1-1	防止作業執行於錯誤發生時.....	122
§ 5-1-2	控制特定程式作業之執行.....	126
§ 5-1-3	控制在跳頁發生時之計算.....	132
§ 5-1-4	依其他計算之結果而決定執行之計算.....	133
§ 5-2	依據資料束中下一筆資料而決定計算之執行.....	137
§ 5-2-1	處理卡片或磁碟資料束.....	137
§ 5-2-2	防止重複.....	141
§ 5-2-3	當一組資料中僅有一筆資料時執行某些特殊之作業.....	147
§ 5-2-4	處理一組中最後一筆資料時執行某些特殊之作業.....	149
§ 5-2-5	另外有關 look ahead 所應考慮的幾點.....	150
§ 5-3	移動資料.....	151
§ 5-3-1	移動資料之敘述.....	151
§ 5-3-2	利用 MOVE 由分欄中儲存 information	152
§ 5-3-3	將一分欄分為兩部份.....	156
§ 5-3-4	改變分欄的型式(文數或數字).....	157
§ 5-4	計算中之分歧.....	159
§ 5-4-1	越過某些計算.....	160
§ 5-4-2	倒逆分歧.....	166
§ 5-5	在計算中利用副常式.....	169
§ 5-5-1	在一週期中利用副常式做許多次同樣的計算.....	169
§ 5-5-2	利用副常式 overlay	181
§ 5-6	控制階指示器之特殊用法.....	182
§ 5-6-1	控制階指示器 L0	182
§ 5-6-2	控制階指示器做為計算之條件指示器.....	186
§ 5-6-3	合計印表.....	187
§ 5-7	二進位分欄運算.....	193
§ 5-7-1	BITON 運算碼.....	194

§ 5-7-2	BITOF 運算碼	194
§ 5-7-3	TESTB 運算碼	195
§ 5-7-4	範例	196
§ 5-8	加快作業速度(雙重1/0區域)	197
§ 5-8-1	雙重輸入區域	197
§ 5-8-2	雙重輸出區域	198

第六章 配對分欄與多資料束之處理

§ 6-1	檢查在一資料束中資料之次序	202
§ 6-1-1	資料束中只含一種資料型	202
§ 6-1-2	資料束中含有一種以上之資料型	205
§ 6-2	一資料束中含有一種以上資料型並利用配對分欄及分欄歸屬關係	207
§ 6-2-1	所有資料型具有相同的配對分欄	207
§ 6-2-2	不同資料型具有不同的配對分欄	209
§ 6-3	配對資料：每一資料束中有一種資料型	212
§ 6-3-1	處理順序：一個以上之配對資料在次資料束中	212
§ 6-3-2	處理順序：一個以上之配對資料在主資料束中	220
§ 6-3-3	配對資料：在另一資料束中沒有配對之資料	222
§ 6-4	配對資料：在一資料束中有一種以上之資料型	226
§ 6-4-1	配對分欄在同樣資料束中不同的位置	226
§ 6-4-2	處理沒有配對分欄之資料	230
§ 6-5	配對資料：在一資料束中所有之資料均處理過時	233
§ 6-6	在同一資料束中利用配對分欄與控制分欄	235
§ 6-7	決定資料束應為主要或次要	241

第七章 程式控制輸入和輸出

§ 7-1	改變資料束處理的次序(FORCE 運算)	242
§ 7-1-1	處理下一筆資料束之描述	243
§ 7-1-2	交互處理兩資料束	244
§ 7-1-3	由一資料束中強迫若干筆資料	247
§ 7-1-4	利用 Look Ahead 以決定是否應強迫某一資料束	257
§ 7-2	處理強求資料束(DEMAND資料束)	267
§ 7-2-1	利用 READ 及強求資料束時應考慮的事情	271

§ 7 - 3	重複輸出 (EXCPT 運算)	271
§ 7 - 3 - 1	利用 EXCPT 及 *PLACE	272
§ 7 - 3 - 2	控制 EXCPT 運算之用法	274

第八章 對照表

§ 8 - 1	簡介	276
§ 8 - 1 - 1	尋找單一的對照表	278
§ 8 - 1 - 2	對照表輸入資料之設計	278
§ 8 - 1 - 3	附加表格與對照表輸入資料之描述	280
§ 8 - 1 - 4	尋對照表之運算 (LOKUP)	283
§ 8 - 2	兩對照表之尋表	285
§ 8 - 2 - 1	兩對照表輸入資料之設計	286
§ 8 - 2 - 2	在附加表格上描述兩對照表	289
§ 8 - 2 - 3	尋表運算 (LOKUP) 之填法	290
§ 8 - 3	利用對照表資料於計算和輸出內	291
§ 8 - 3 - 1	依尋表結果控制之計算	291
§ 8 - 3 - 2	尋表成功後參考資料	292
§ 8 - 3 - 3	尋找小於、大於、或等於之情況	294
§ 8 - 3 - 4	將資料移入一對照表 entry 內	298
§ 8 - 3 - 5	對照表內容之修正	300
§ 8 - 4	對照表的裝填	303
§ 8 - 4 - 1	Compile 之對照表	303
§ 8 - 4 - 2	Pre-execution 時間之對照表	304
§ 8 - 4 - 3	Pre-execution 時間對照表之裝填	304
§ 8 - 5	整個對照表之輸出	307

第九章 排列 (ARRAYS)

§ 9 - 1	何時應用排列而不用對照表	309
§ 9 - 2	排列之定義	310
§ 9 - 3	參考在一排列中之所有元素	311
§ 9 - 3 - 1	排列至排列之計算	314
§ 9 - 3 - 2	利用排列及單一分欄 (或常數) 之計算	316
§ 9 - 3 - 3	排列內之所有元素相加	317
§ 9 - 3 - 4	整個排列之輸出	318

§ 9 - 3 - 5	各組和之累積.....	322
§ 9 - 4	參考排列中單獨之各元素.....	329
§ 9 - 4 - 1	排列之指標.....	329
§ 9 - 4 - 2	排列中單獨元素之輸出.....	332
§ 9 - 4 - 3	僅參考一分欄之部份.....	333
§ 9 - 5	排列之 lookup	337
§ 9 - 5 - 1	尋排列中之一特殊元素.....	337
§ 9 - 5 - 2	尋找排列中一個以上的元素.....	343
§ 9 - 5 - 3	輸出於排列尋找之間.....	345
§ 9 - 6	排列之裝填.....	346
§ 9 - 6 - 1	Compile 時間之排列	346
§ 9 - 6 - 2	Pre - execution 時間之排列.....	347
§ 9 - 6 - 3	將輸入資料存入 execution 時間之排列內.....	350
§ 9 - 6 - 4	排列 data 連續的在一筆以上之資料上.....	356
中英名詞對照.....		364
索 引.....		366

1

RPG II的邏輯

假使我們準備送一份帳單給客戶，我們必需做些什麼呢？在有所行動之前，我們需要一些資料。我們必需要知道三件事情：(1)月初對客戶的結存(2)購貨量(3)付款的情形。只要我們一有了以上的三種資料，就能算出客戶應付的總數並將總數填在帳單上。對每一個客戶都是同樣的做法。

我們可以很容易的交給電腦做這一類的工作。然而，電腦也必需知道我們所必需知道的資料。我們還必需告訴它：我們希望知道的是什麼，我們怎麼做出這個結果。這些說明都該寫在資料束表格(File Description)；附加表格(Extention)；計行表格(Line Counter)；輸入表格(Input)；計算表格(Calculation)；及輸出表格(Output)內。

在我們算一份帳單時，第一步先讀入一筆資料，其次計算，最後將計算的結果記下來；接著我們按同樣的邏輯對下一個顧客做同樣的工作。

電腦也是照著同樣的一個邏輯來做這個工作。這邏輯由 RPG II 的 compiler 供給而並非由我們所填的 RPG II 表格供給的。當我們的原始程式經過 compile 以後，所有原始指令都照著 RPG II 程式邏輯的骨架而形成一個完整的程式。

在做算帳的工作時，如果發現一個客戶的付出比他所除欠的多時，怎麼辦呢？我們馬上知道差額並且將這差額記在帳上。RPG II 的程式怎麼瞭解這種情形呢？同時它又怎麼知道在這種情形發生時怎麼辦呢？

RPG II 程式用訊號來表明在某一特殊情況發生時該怎麼辦。這訊號就叫做指示器(indicators)。有許多種不同種類的指示器表明各種不同的情況。一個程式設計師必需要知道怎麼樣利用指示器而讓電腦知道我們想要它們做些什麼。

RPG II 的邏輯就建立在這些指示器上，指示器的開或關(on or off)影響程式作業的順序。RPG II 邏輯的建立是在特殊的時間裡探試各種不同的指示器。由探試指示器，程式就瞭解下一部該做些什麼。

RPG II 程式邏輯考慮到了各類的工作。我們必需瞭解這邏輯而後才能正確的使用它。

由於 RPG II 的邏輯是一個相當複雜的題目，在本章中我們一段段的分析，讀完了本章後，對於 RPG II 程式邏輯將會有著很清晰的概念。

1-1 資料處理的基本邏輯

通常在一個輸入資料束裏的所有資料並非一次讀入，我們所用的電腦可能不夠大到在同一時間裏儲存並處理所有的資料，因此，資料是一筆一筆的讀入，也就是一次讀入一筆。如圖 1-1 所示，每筆資料讀入後都經過三個步驟。

一個週期（cycle）表示從讀入一筆資料，執行所有指令，到讀入下一筆資料之前，這中間的過程就是一個週期。因此週期就說明了程式邏輯。對所讀入之每一筆資料都要經過一個週期。

如果需要每個客戶購貨的詳細清單，讓我們看看如圖 1-1 的三個步驟。輸入資料束按著客戶號碼的漸增次序排列。每筆資料包含著客戶姓名（NAME），客戶號碼（NUM），和欠帳（CRG）。輸入的每筆資料都要印出，每一行印一筆。

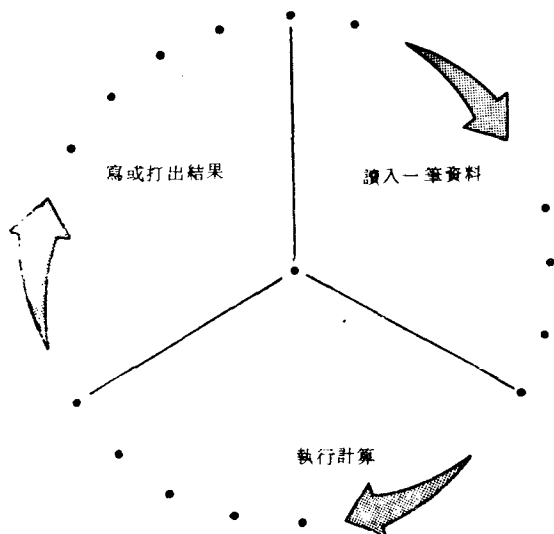


圖 1-1 資料處理的基本邏輯

整個作業過程開始：首先讀入第一筆資料，沒有任何運算，接著這筆資料印出，到此第一個週期結束。第二週期開始於讀入下一筆資料，圖 1-2 表示細目印表工作的輸入和輸出。

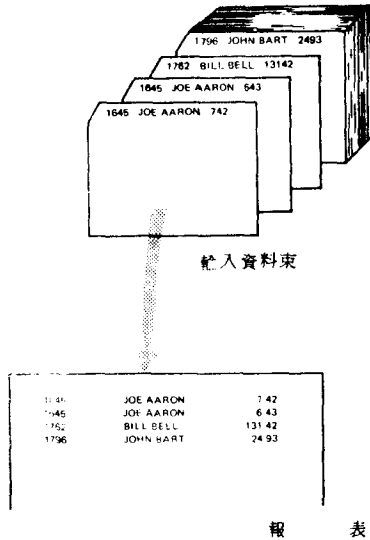


圖 1-2 細目印表工作

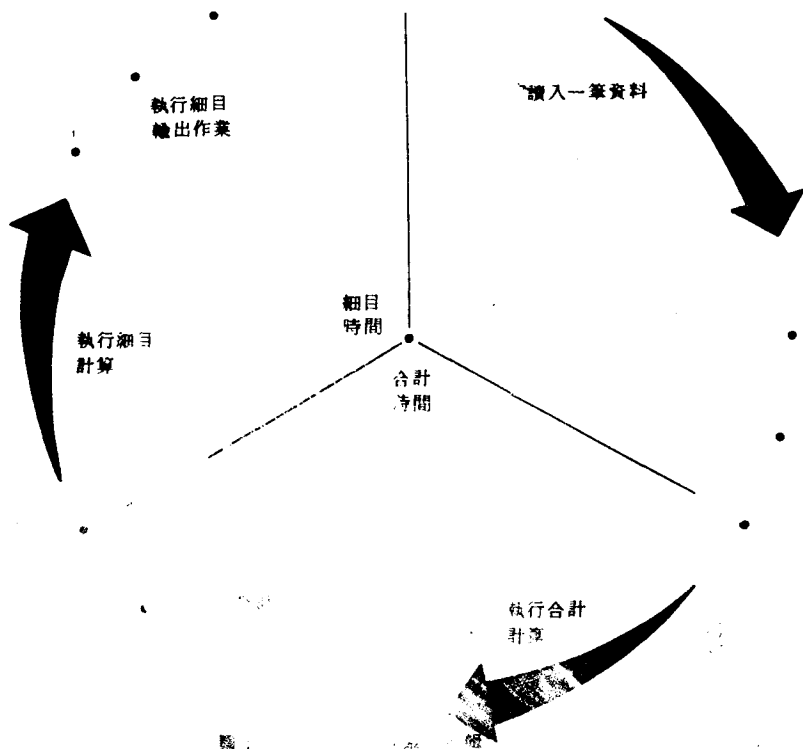
1645	JOE AARON	7.42
1645	JOE AARON	6.43
		13.85*
1762	BILL BELL	131.42
		131.42*
1796	JOHN BART	24.93
1796	JOHN BART	2.98

圖 1-3 計算並印出總和

假如不僅單單做出每個客戶的欠帳單而還想知道每個客戶的欠帳總和，如圖 1-3 所示，則必需在每一筆資料印出之外加上計算總數的手續。但什麼時候即出這總數呢？當然，對每個客戶，他的欠帳總數要印在該客戶所有資料之下方（如圖 1-3）。但是到目前為止我們所討論的三個步驟，都沒有準備印總數，同時也無法在每一筆資料之間區別出什麼時候某一客戶的資料已全部讀過了。如果 RPG I 的邏輯只有我們所說的三個步驟，勢必無法解決這些問題。

1-2 RPG II 的基本邏輯

因此，RPG I 的邏輯就是這三個步驟的延伸，在一個週期中兩個不同的時間裏做計算和輸出（參考圖 1-4），這兩個時間分別是細目時間及合計



時間。所謂合計時間就是將一組相關資料卡中的資料累積起來的所有操作的時間，如圖 1-3 中印出 JOE ARRON 欠帳總數就是一個合計運算。所謂細目時間就是處理每一筆單一資料的時間，例如印 JOE ARRON 每筆欠帳的時間。記住！細目運算是對讀入之每筆資料，而合計運算只在某一同類資料都讀入之後才做（如圖 1-5）。

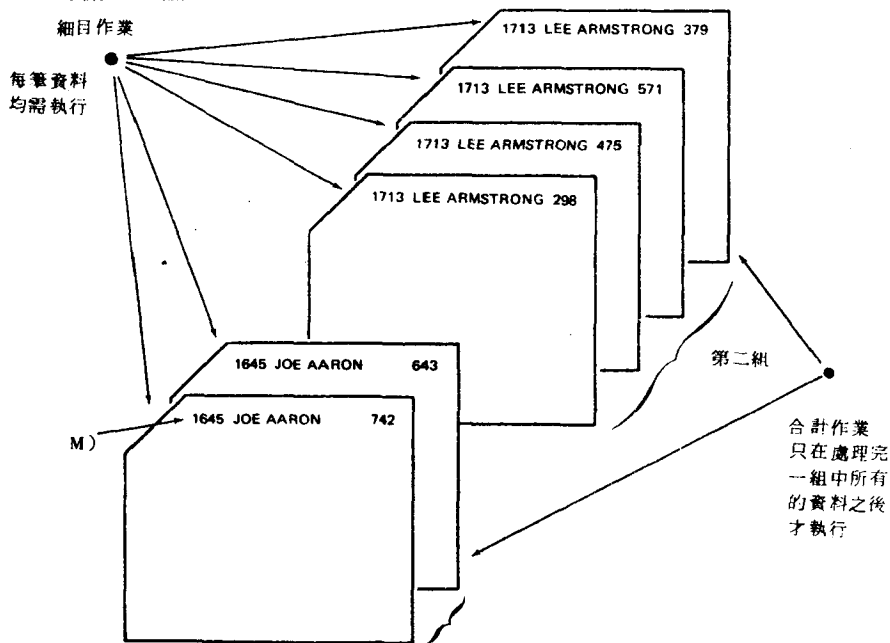


圖 1-5 細目及合計作業

因為 RPG II 的基本邏輯只是一個骨架，我們必需供應一些資料始能正確的完成我們的工作。例如，RPG II compiler 供給我們能做細目及合計運算的骨架，而由我們告訴它什麼時候該做合計的工作，哪些計算和輸出該在細目時間做，哪些又該在合計時間做。

別忘了我們能告訴程式在某一特定情況下做些什麼的唯一辦法就是指示器。控制階指示器（control level indicators）用來告訴程式：

- 1 什麼時候執行合計作業（total operations）。
- 2 哪些作業屬於合計作業。

假如我們想知道整個月裏每個客戶的欠帳總額，我們怎麼知道什麼時候

記錄下每一客戶的累積額？當我們碰到一筆資料的客戶號碼（NUM）不同時，我們就知道應該對上一客戶做合計的工作，然後才能對這筆有著不同客戶號碼的資料有所行動。

檢查任何分欄（field）之內容有無改變時，這分欄就叫做控制分欄。利用輸入表格第 59—60 格給予該分欄一控制階指示器（L1—L9）來表示哪些是控制分欄，我們也可以在計算表格上第 7—8 格填入控制階指示器來告訴程式哪些運算是合計運算，而其餘不受控制階指示器限制的運算都是細目運算。在輸出表格內，細目或合計輸出，並不用控制階指示器來表示，而是在第 15 行上填 T 表示合計輸出；填 H 或 D 表示在細目時間裏的作業。

1—2—1 在RPG II基本邏輯裡的特殊步驟

圖 1—4 表示出在一個 RPG II 程式週期裏的一般次序，事實上 RPG II 的邏輯在一個週期裏包含了一定的步驟。在我們做一件工作時，常會自問：「我現在要不要做這個？」「所有我要的資料都齊全了嗎？」「下一步該做什麼？」同樣的，RPG II 的邏輯也問這些問題，它利用我們的程式來找答案同時決定下一步該做什麼。這些問題，及找出答案所採取的各步驟在圖 1—6 中顯示出來。

依照 RPG II 的邏輯，一筆資料讀入之後，程式就檢查它控制分欄的內容是否改變（程式總是將控制分欄存起來以便於比較），如果改變了，就令適當的控制階指示器（我們所指定的）為開，這就表示所有同類的資料已經讀過了，接著就該做它們的合計作業。控制階指示器在下一筆資料讀入之前關掉。

注意在合計時間及細目時間之間的步驟，週期一開始時讀入的資料在此時移入處理位置（processing area）並且可用來計算和輸出，但這筆資料並不適用於合計時間，合計作業只對上一筆做累積的工作，控制階指示器為開時，細目作業之執行在上一筆合計作業執行之後。

為什麼合計作業在細目作業之前？想想看如果處理某一筆使控制階指示器為開的資料時，會發生什麼事呢？這筆資料的 information 會加入上一組資料的 information 裏。合計輸出所得的結果包含了屬於下一組的一筆資料（亦即剛讀入的那一筆資料）上的 information，這結果是錯誤的，為了防止下一組中的第一筆資料累積到前一組的資料裏，合計作業必需在細目作業之前。

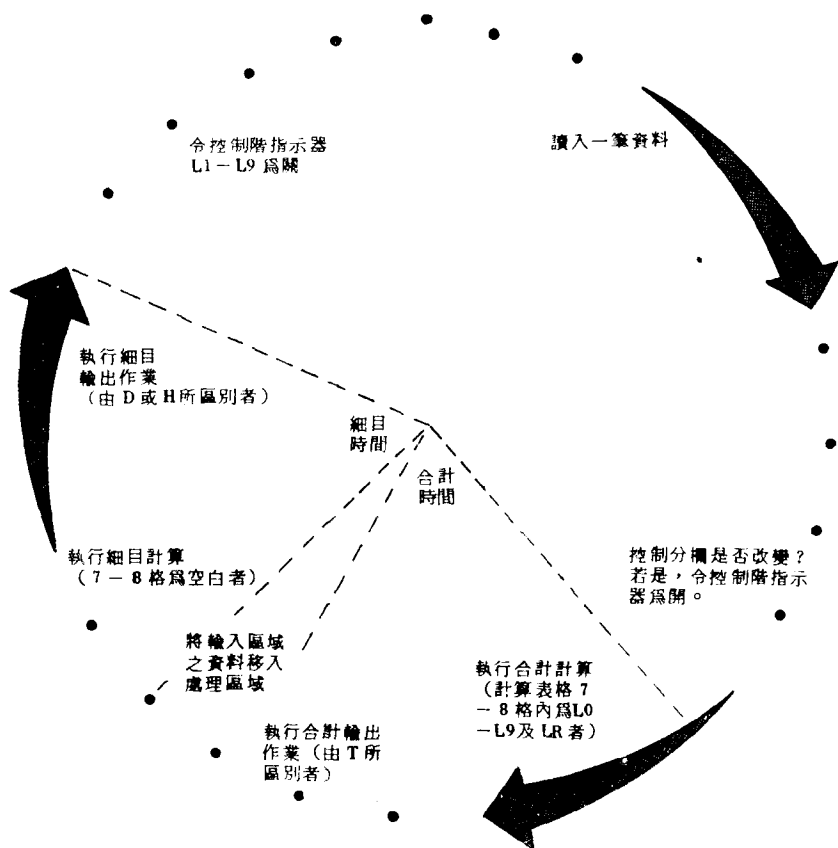


圖 1-8 在 RPG II 合計及細目時間邏輯裡的各個步驟

1-2-2 第一個程式週期

資料有控制分欄時，第一個程式週期與其餘的程式週期稍有不同，含有控制分欄的第一筆資料使控制階指示器為開，因為在第一筆資料讀入之前，主儲存體裏有著空白的控制分欄區，這第一筆資料的控制分欄與主儲存體裏空白的控制分欄不等，造成了控制階的改變，為了防止在第一週期裏輸出空白的合計資料，RPG II 的邏輯令第一週期不執行合計作業。

註：如果起先的資料都不包含控制分欄，則所有這些資料的程式週期以