

公告本

410311

申請日期	84 年 9 月 7 日
案 號	84109370
類別·C16	G06F 7/06

A4
C4

(以上各欄由本局填註)

發明專利說明書 410311

一、發明 名稱	中 文	編碼及解碼資料之方法及裝置
	英 文	Method and apparatus for encoding and decoding data
二、發明 人	姓 名	(1) 愛德華·史瓦茲 Schwartz, Edward L. (2) 麥可·高米希 Gormish, Michael J. (3) 詹姆士·亞倫 Allen, James D.
	國 籍	(1) 美國 (2) 美國 (3) 美國
	住、居所	(1) 美國加州門羅公園沙丘路二八八二號一一五室 c/o Richo Corporation, California Research Center, 2882 Sand Hill Road, Suite 115, Menlo Park, CA 94025 U.S.A. (2) 美國加州門羅公園沙丘路二八八二號一一五室 c/o Richo Corporation, California Research Center, 2882 Sand Hill Road, Suite 115, Menlo Park, CA94025 U.S.A. (3) 美國加州門羅公園沙丘路二八八二號一一五室 c/o Richo Corporation, California Research Center, 2882 Sand Hill Road, Suite 115, Menlo Park, CA94025 U.S.A.
	住、居所	
三、申請人	姓 名 (名稱)	(1) 理光股份有限公司 株式会社リコー
	國 籍	(1) 日本
	住、居所 (事務所)	(1) 日本國東京都大田區中馬込一丁目三番六號
	代 表 人 姓 名	(1) 屋間健治

裝

訂

線

410311

410311

申請日期	84 年 9 月 7 日
案 號	84109370
類 別	

A4
C4

(以上各欄由本局填註)

發 明 專 利 說 明 書

一、發明 名稱	中 文	
	英 文	
二、發明 創作人	姓 名	(4) 馬汀·保烈克 Boliek, Martin
	國 籍	(4) 美國
		(4) 美國加州門羅公園沙丘路二八八二號一一五室
	住、居所	c/o Richo Corporation, California Research Center, 2882 Sand Hill Road, Suite 115, Menlo Park, CA94025 U.S.A.
三、申請人	姓 名 (名稱)	
	國 籍	
	住、居所 (事務所)	
	代 表 人 姓 名	

裝

訂

線

410311

(由本局填寫)

承辦人代碼：

A6

大類：

B6

IPC分類：

本案已向：

國(地區) 申請專利，申請日期： 案號： ，☐有 ☐無主張優先權
美國 1994 年 9 月 30 日 S.N.08/316116 ☒無主張優先權

有關微生物已寄存於： ，寄存日期： ，寄存號碼：

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

經濟部中央標準局員工消費合作社印製

本紙張尺度適用中國國家標準 (CNS) A4規格 (210×297公釐)

410311

五、發明說明(7)

圖20為具有分開位元產生器之解碼器的一個實施例之方塊圖。

圖21為本發明之位元產生器的一個實施例之方塊圖。

圖22為本發明之長運轉單元的一個實施例之方塊圖。

圖23為本發明之短運轉單元的一個實施例之方塊圖。

圖24為本發明之啓始及控制邏輯的一個實施例之方塊圖。

圖25為使用司魯普(snooper)解碼器來重序資料的一個實施例之方塊圖。

圖26為重序單元之另一個實施例的方塊圖。

圖27為使用合併佇列之重序單元的另一個實施例之方塊圖。

圖28為使用本發明之高頻帶寬度系統的方塊圖。

圖29為使用本發明之高頻帶寬度匹配系統的方塊圖。

圖30為使用本發明之即時視頻系統的方塊圖。

圖31指出本發明之編碼資料記憶器的一個實施例。

圖32為本發明之解碼的時間圖(timing diagram)。

圖33指出編碼效率(相對於一致性使碼長度正常化)對MPS概率之圖形。

本發明之詳細敘述：

敘述一種方法及裝置用於資料的平行編碼及解碼。在以下的敘述中，提出數目的特定細節，例如特定數目的位元、編碼器的數目、特定概率、資料的型式等等，以提供本發明之較佳實施例的完整瞭解。可瞭解對熟於此技藝之人仕而言，可以在沒有這些特定細節下來實施本發明。而

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(1)

本案為1993年12月23日於美國申請的第08/172,646號美國專利申請案，標題為"Method And Apparatus For Parallel Encoding and Decoding of Data"之後續案，其為1993年2月10所申請的第08/016,035號美國專利申請案，標題為"Method And Apparatus For Parallel Decoding and Encoding of Data"之後續案。

發明領域：

本發明係關於資料壓縮及解壓縮系統的領域；特別是，本發明係關於一種方法及裝置用於壓縮/解壓縮系統中的資料之平行編碼及解碼。

發明背景：

現今已廣泛地使用資料壓縮，特別是用於儲存及傳送大量的資料。許多不同的資料壓縮技術存在於習知技術中。壓縮技術可以區分成兩種廣的分類：有損耗編碼及無損耗編碼。有損耗編碼含有會導致資訊損耗之編碼，使得無法保證原始資料之完美重建。在無損耗壓縮中，全部的資訊被保持，且資料是以可允許完美重建的方式被壓縮。

在無損耗壓縮中，輸入符號被轉換成輸出碼字。如果壓縮成功的話，碼字是以比輸入符號更少的位元來表示。無損耗編碼方法包括編碼之字典方法(例如Lempel-Ziv)、掃描寬度、列舉(enumerative)編碼及一致性(entropy)編碼。

五、發明說明(2)

一致性編碼是無損耗編碼方法所構成，其利用已知或估計的符號概率，而可以壓縮資料接近一致性極限。一致性碼包括赫夫曼碼(Huffman codes)、數學碼及二進位一致碼。二進位一致性編碼器為無損耗的編碼器，其只作用於二進位(yes/no)決定，通常表示為最大概率符號(MPS)及最小概率符號(LPS)。二進位一致性編碼器之例子包括IBM's Q-編碼器及稱為B-編碼器之編碼器。B-編碼器之其它的資訊，請參閱1993年12月21日公告之J.D. Allen的第5,272,478號美國專利，標題為"Method and Apparatus for Entropy Coding"，其亦讓渡給本發明之受讓人。請亦參閱M.J. Gormish與J.D. Allen摘錄於Proc. Data Compression Conference, 1993年3月30日Snowbird UT第449頁之"Finite State Machine Binary Entropy Coding"。B-編碼器是一種二進位一致性編碼器，其使用一有限狀態機用於壓縮。

圖1指出使用二進位一致性編碼器之習知壓縮及解壓縮系統之方塊圖。為了編碼，資料被輸入至上下文模型(CM)101。CM101將輸入資料傳送進入一組或順序的二進位決定，並提供上下文筐(context bin)給各決定。順序的二進位決定及其有關的上下文筐，從CM101被輸出至概率估計模組(PEM)102。PEM102接收各上下文筐，並對於各二進位判定(binary decision)產生概率估計。實際的概率估計典型地以一等級(class)來表示，稱為PClass。各PClass係使用於一範圍的概率。PEM102亦決定二進位判

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (3)

定(結果)是否在其最可能的狀態(亦即此判定是否對應MPS)。
位元流產生器(BG)模組103接收概率估計(亦即PClass)
，及二進元判定是否可能成為輸入之決定。響應此一動作
，BG模組103產生一壓縮資料流，輸出0或更多的位元，以
表示原始輸入資料。

為了解碼，CM104提供上下文筐至PEM105，且PEM105
根據上下文筐而提供概率等級(PClass)至BG模組106。BG
模組106被耦合以接收概率等級。響應概率等級及壓縮資
料，BG 模組106送回一個位元，表示二進位判定(亦即事
件)是否在其最可能的狀態。PEM105接收此位元，根據所
接收的位元來更新(update)概率估計，並將結果送回至
CM104。CM104接收所送回的位元，並使用送回的位元以產
生原始資料，且更新下一位二進位判定之上下文筐。

使用二進位一致性碼之解碼器例如IBM之Q-編碼器及
B-解碼器會發生一問題：即使當在硬體中實施時，它們的
速度很慢。它們的操作需要單一的大且慢之反饋迴路。欲
使解碼方法重新開始，上下文模型使用過去的解碼資料來
產生上下文。概率估計模組使用上下文以產生一概率等級
。位元流產生器使用概率等級及壓縮資料來決定下一個位
元為可能或不可能的結果。概率估計模組使用此可能/不
可能結果，以產生一結果位元(並更新上下文之概率估計)
。上下文模型使用此結果位元以更新過去資料之歷史。解
碼單一位元時需要全部這些步驟。因為上下文模型在其可
以提供下一個上下文之前，必須等待結果位元以更新其歷

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(4)

史，下一個位元之解碼必須等待。最好在解碼下一個位元之前，可以避免必須等待反饋迴路。換句話說，最好在一次可以解碼一個以上的位元或碼字(codeword)，以增加壓縮資料被解碼的速度。

使用二進位一致性碼之解碼器的另一個問題是：必須處理可變長度資料。在大部份的系統中，欲被解碼之碼字具有可變長度。或者，其它的系統編碼可變長度符號(未解碼的資料)。當處理可變長度資料時，需要使在位元位準(bit level)之資料移位，以提供正確的下一個資料用於解碼或編碼操作。在資料流上之這些位元位準操作將需要昂貴且/或緩慢的硬體且/或軟體。此外，習知系統需要在時間臨界反饋迴路中完成此移位，反饋迴路會限制解碼器之性能。最好可以將資料流之位元位準操作從時間臨界反饋迴路移離，使得可以使用平行的方式來增加速度。

本發明提供一種無損耗壓縮及解壓縮系統。本發明亦提供一種即時(real-time)編碼器及一種即時解碼器，其可以分別平行地編碼及解碼資料。本發明之編碼器及解碼器創造一種平衡的平行一致性系統，其可在高速/低價位的硬體中執行即時編碼及即時解碼。

發明節要：

敘述一種方法及裝置用於解壓縮及壓縮資料。本發明提供一種編碼器，可使用在具有可解碼由編碼器所產生的資訊之解碼器的一種壓縮系統中。本發明之編碼器包括一

五、發明說明(5)

編碼器，用於響應資料而產生碼字資訊。編碼器亦包括一重序單元(reorder unit)，其響應來自編碼器之碼字資訊而產生一編碼後的資料流。此重序單元包含一運轉計數重序單元，用於將碼字排列成解碼順序，及一位元封包單元可將可變長度碼字組合成固定長度空白字，並以解碼器所需之順序將固定長度空白字輸出。

圖形之簡要敘述：

從本發明之許多不同的實施例之附圖及以下的詳細敘述，將更完整地瞭解本發明，然而本發明並不限於特定的實施例，較佳實施例之目的是便於說明及瞭解。

圖1為習知的二進位一致性編碼器及解碼器之方塊圖。

。

圖2A為本發明之解碼系統的方塊圖。

圖2B為本發明之編碼系統的一個實施例之方塊圖。

圖2C為本發明之解碼系統的一個實施例之方塊圖，其平行地處理上下文筐。

圖2D為本發明之解碼系統的一個實施例之方塊圖，其平行地處理概率等級。

圖3指出本發明之非空白碼流。

圖4指出當從一範例組資料導出時，空白碼流的一個實施例。

圖5為本發明之R-編碼器的位元流產生器及概率估計表的一個例子。

五、發明說明(6)

圖 6 爲本發明之編碼器的一個實施例之方塊圖。

圖 7 爲本發明之位元產生器的一個實施例之方塊圖。

圖 8 爲本發明之重序單元的一個實施例之方塊圖。

圖 9 爲本發明之運轉計數重序單元的一個實施例之方塊圖。

圖 10 爲本發明之運轉計數重序單元的另一個實施例之方塊圖。

圖 11 爲本發明之位元封包單元的一個實施例之方塊圖。

圖 12 爲本發明之封包邏輯的一個實施例之方塊圖。

圖 13 爲本發明之編碼器位元產生器的一個實施例之方塊圖。

圖 14A 爲本發明之解碼系統的一個實施例之方塊圖。

圖 14B 爲本發明之解碼器的方塊圖。

圖 14C 爲本發明之 FIFO 結構的一個實施例之方塊圖。

圖 15A 指出本發明之解碼管道(pipeline)的一個實施例。

圖 15B 指出本發明之解碼器的一個實施例。

圖 16A 爲本發明之移相器(shifter)的一個實施例之方塊圖。

圖 16B 爲本發明之移相器的另一個實施例之方塊圖。

圖 17 爲具有外部上下文模型之系統的方塊圖。

圖 18 爲具有外部上下文模型之另一個系統的方塊圖。

圖 19 爲本發明之解碼器的一個實施例之方塊圖。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

410311

五、發明說明(7)

圖20為具有分開位元產生器之解碼器的一個實施例之方塊圖。

圖21為本發明之位元產生器的一個實施例之方塊圖。

圖22為本發明之長運轉單元的一個實施例之方塊圖。

圖23為本發明之短運轉單元的一個實施例之方塊圖。

圖24為本發明之啓始及控制邏輯的一個實施例之方塊圖。

圖25為使用司魯普(snooper)解碼器來重序資料的一個實施例之方塊圖。

圖26為重序單元之另一個實施例的方塊圖。

圖27為使用合併佇列之重序單元的另一個實施例之方塊圖。

圖28為使用本發明之高頻帶寬度系統的方塊圖。

圖29為使用本發明之高頻帶寬度匹配系統的方塊圖。

圖30為使用本發明之即時視頻系統的方塊圖。

圖31指出本發明之編碼資料記憶器的一個實施例。

圖32為本發明之解碼的時間圖(timing diagram)。

圖33指出編碼效率(相對於一致性使碼長度正常化)對MPS概率之圖形。

本發明之詳細敘述：

敘述一種方法及裝置用於資料的平行編碼及解碼。在以下的敘述中，提出數目的特定細節，例如特定數目的位元、編碼器的數目、特定概率、資料的型式等等，以提供本發明之較佳實施例的完整瞭解。可瞭解對熟於此技藝之人仕而言，可以在沒有這些特定細節下來實施本發明。而

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(8)

且，爲了避免不必要的麻煩，習知電路是以方塊圖之形式而非詳細地指出。

以下之詳細敘述的某些部份是以一電腦計憶器中的資料位元上之操作的符號表示及演算法來表示。這些演算的敘述及表示方式爲熟於資料處理領域所常用的手法，可將他們的工作內容傳送至熟於此技藝之其他人仕。通常，在這裡演算式可以視爲一種能導致理想的結果之前後一致的順序之步驟。這些步驟需要物理量的實體操作。通常，雖然不是必要的，這些量是做成能夠被儲存、傳送、組合、比較、或操作之電或磁信號的形式。主要是爲了普遍使用之理由，已多次地證實這些信號如做成位元、數值、元件、符號、字元、項數、數字之形式是方便的。

然而，須注意全部這些或類似的術語是與適當的物理量有關，且僅爲適於應用至這些量的方便標示。除非特別地指出，否則從以下的討論可明顯看出，在本發明中使用例如"處理"、"計算"、"計數"、"決定"、"顯示"等等術語之討論，是指電腦系統或類似的電子計算裝置之處理及動作，可操作及轉換由物理(電子)量所表示的資料於電腦系統之暫存器及記憶器中，形成在電腦系統記憶器或暫存器(或其它的資訊儲存、傳送或顯示裝置)中由物理量所表示的類似資料。

本發明亦關於執行操作之裝置。此裝置可以針對需要的用途而特殊地製成，或者可以包含一般用途的電腦，可由電腦中所儲存的電腦程式選擇性地啓動或重建。在這裡

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(9)

所表示的顯示及演算式並不是關於任何特殊的電腦或其它的裝置。依據這裡的揭示，能以程式來使用許多不同用途的機器，或者可以方便地建造更特殊的裝置以執行所要的方法步驟。從以下之敘述可以明顯看出不同的這些機器所需之構造。此外，本發明並不是針對任何特殊的程式語言加以敘述。可以使用許多不同的程式語言來實施本發明中所敘述的揭示。

平行一致性編碼：

本發明提供一種平行一致性編碼系統。此系統包括一編碼器及一解碼器。在一實施例中，編碼器執行即時資料之編碼。同樣地，在一實施例中，解碼器執行即時資料之解碼。此即時編碼器及即時解碼器可一起形成一種平衡的編碼系統。

本發明提供一種系統，其可平行地以無損耗的方式來解碼編碼後的資料。資料是藉由使用多解碼源而被平行地解碼。各多解碼源為從資料流被指定欲解碼之資料(例如碼字)。資料流之指定是發生在飛行上，其中解碼源同時地解碼資料，藉以平行地解碼資料流。欲能夠指定資料，其方式使得可有效地使用解碼源，所以資料流被重序。這是稱為使資料流平行化。資料之重序可允許各解碼源來解碼任何或全部的編碼資料，而不需要等待來自上下文模型之反饋。

圖2A指出沒有習知的緩慢反饋迴路，本發明之解碼系

五、發明說明(10)

統。輸入緩衝器204接收編碼資料(亦即碼字)及來自解碼器205之反饋信號，並以預定順序(例如上下文筐順序)而供給編碼資料至本發明之解碼器205，其解碼編碼後的資料。解碼器205包括多個解碼器(例如205A、205B、205C等等)。

在一實施例中，各解碼器205A、205B、205C等等被供給一群上下文的資料。各解碼器中之解碼器205，從輸入緩衝器204被供給在一群上下文中的各上下文筐之編碼資料。使用此資料，各解碼器205A、205B、205C產生一群上下文筐之解碼資料。上下文模型並不需要結合編碼資料與特殊群的上下文筐。

解碼後的資料被解碼器205送至解碼資料儲存207(例如207A、207B、207C等等)。須注意解碼資料儲存207可以儲存中間資料，此中間資料不是被編碼或未編碼，例如運轉計數。在此情形中，解碼資料儲存207是以緊密的形式而非一致性編碼的形式來儲存資料。

獨立地操作，使上下文模型206被耦合，以響應一反饋信號而接收來自解碼資料儲存207(亦即207A、207B、207C等)之預先解碼的資料，此反饋信號送至解碼資料儲存207。所以，存在兩種獨立的反饋迴路，其中一個位在解碼器205及輸入緩衝器204之間，而第二個則位在上下文模型206與解碼資料儲存207之間。由於除去大的反饋迴路，所以解碼器205中之解碼器(例如205A、205B、205C等)能夠解碼與它們有關的碼字，只要一旦它們從輸入緩衝器

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (11)

204被接收。

上下文模型提供編碼系統之記憶器部份，並根據記憶器而將一組資料(例如影像)分成不同的分類(例如上下文筐)。在本發明中，上下文筐是視為獨立順序組的資料。在一個實施例中，各群的上下文筐具有其自己的概率估計模組，且各上下文筐具有其自己的狀態(在這裡分享概率估計模組)。所以，各上下文筐可以使用不同的概率估計模組且/或位元流產生器。

於是，資料被重序或平行化，且來自資料流之資料被指定給各別的編碼器以解碼。

將平行化加至等級一致性編碼模型：

欲使資料流平行化，資料可以依據上下文、概率、覆蓋(tiling)、碼字順序(根據碼字)等來分割資料。編碼後的資料流之重序是與平行化、用以平行化資料之方法、或在任何其它點的概率無關。本發明之編碼系統的平行編碼器部份是表示於圖2B中，可饋入由上下文模型(CM)所微分的資料。

參照圖2B，上下文相依平行編碼器部份包含上下文模型(CM)214、概率估計模組(PEMs)215-217、及位元流產生器(BGs)218-220。耦合CM214以接收編碼後的輸入資料。CM214亦被耦合至PEMs215-217。PEMs215-217亦分別耦合至BGs218-220，其分別輸出碼流1、2、3。各PEM及BG對包含一編碼器。所以，指出的平行編碼器具有三個編碼器

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(12)

。雖然只有指出三個平行的編碼器，亦可以使用任意數目的編碼器。

CM214是以相同於習知CM的方式而將資料流分成不同的上下文，並將許多資料流送至平行的硬體編碼源。各別的上文文，或成群的上下文，被送至各別的概率估計器(PEMs)215-217及位元產生器(BGs)218-219。各BGs218-220輸出一編碼後的資料流。

圖2C為本發明之解碼系統的解碼器部份之一個實施例的方塊圖。參照圖2C，指出上下文相依平行解碼器具有BGs221-223、PEMs224-226及CM227。碼流1-3分別被耦合至BGs221-223。BGs221-223亦分別耦合至PEMs224-226。PEMs224-226被耦合至CM227，其輸出重建後的輸入資料。此輸入是來自許多碼流，如圖所示為碼流1-3。一碼流被指定給各PEM與BG。各BG221-223送回一位元，其表示二進位判定是否在其最可能的狀態，PEMs224-226使用它而送回解碼後的位元(例如二進位判定)。各PEMs224-226是與BGs221-223的其中之一連接，表示碼欲被使用而從其輸入碼流產生一資料流。CM227藉由從位元流產生器以適當的順序來選定解碼後的位元，可產生一解碼後的資料流，藉以重新產生原始資料。於是，CM227從適當的PEM與BG得到解壓縮後的資料位元，使資料重序而成原始順序。須注意此種設計的控制是與資料流的方向相反。BG與PEM可以在CM27227需要之前，例如超前一個以上的位元，先解碼資料。或者，CM227可以要求(但不是接收)一位元來自一

五、發明說明 (13)

BG與PEM，然後在使用此原始要求位元之前，要求一個以上的位元來自其它的BGs與PEMs。

圖2C所示之造形是設計成可緊密地耦合PEM與BG。IBM之Q-編碼器為具有緊密耦合的PEM與BG之編碼器的一個良好例子。在PEM與BG之間的局部反饋迴路並不會限制系統之性能。

在不同的設計中，PEM可以將資料微分，並將其送至平行的BG單元。於是，會只有一個CM與PEM，且BG被折返(replicated)。以此方式可以使用適應的Huffman編碼與有限狀態機器編碼。

一類似的解碼系統表示於圖2D中，使用PEM來使資料微分，並將其送至平行的BGs。在此情形中，概率等級被平行地處理，且各位元流產生器被指定至一特定的概率等級並接收結果之知識。參照圖2D，編碼後的資料流1-3被耦合至多位元流產生器(例如BG232、BG233、BG234等)的其中之一，多位元流產生器被耦合可接收編碼後的資料流。各位元流產生器被耦合至PEM235。PEM235亦被耦合至CM236。在此造形中，各位元流產生器將編碼後的資料解碼，且PEM235(而非CM236)選擇解碼之結果。各位元流產生器接收來自與一概率等級相關的來源之編碼資料(亦即編碼資料可以是來自任意的上下文筐)。PEM235選擇使用概率等級之位元流產生器。此概率等級是由CM236所提供的上下文筐來表示。以此方式，解碼資料可藉由平行地處理概率等級而產生。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(14)

本發明之平行解碼系統存在有多種實施方式。在一實施例中，對應多上下文筐之編碼後的資料流，可以被插入由許多不同的編碼器之指令所排序(ordered)的流中。在一實施例中，編碼後的資料被排序，使得各編碼器被固定地供給資料，即使編碼後的資料被傳送至資料流中的解碼器。須注意本發明能以全部型式的資料包括影像資料來操作。

藉由使用小型的簡單編碼器，其能便宜地重複設置於積體電路中，編碼後的資料能以平行的方式被快速地解碼。在一實施例中，編碼器被實施於硬體中，使用場可程式閘陣列(FPGA)晶片或標準的格子應用特定積體電路(ASIC)晶片。平行化的組合及簡單的位元流產生器，可允許以超過習知解碼器的速度來使編碼後的資料解碼，而仍保持或增加習知解碼系統的壓縮效率。

多資料流之通道排序：

有許多不同的設計問題會影響系統性能。以下將敘述一些部份的這些問題。然而，圖2B與2C(及2D)所示之實施例是使用多碼流。可以想像具有平行通道的系統，其能適應此實施例：多電話線、一磁碟機上之多磁頭等。在某些應用中，只有一通道是有用或方便的。的確，如果需要多通道，由於各別的碼流之叢發特性(bursty nature)，頻帶寬度之利用性會較差。

在一實施例中，碼流被連結並被接觸地送至解碼器。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (15)

序文標頭(preface header)含有指標(pointer)作為各資料流之位元位置的開始。圖3指出此資料之排列的一個實施例。參照圖3，三個指標301-303分別指出在碼流1、2、3的連結碼中之開始位置。完整的壓縮資料檔在一緩衝器中對解碼器是有用的。因需要，碼字是從適當的位置經由適當的指標被取還(retrieved)。然後指標被更新(updated)成在此碼流中的下一個碼字。

須注意此方法需要整個編碼圖框被儲存在解碼器，而為了實際用途亦可儲存在編碼器。如果需要一即時系統或低叢發資料流，則可以使用兩個圖框緩衝器，以觸排(banking)在編碼器與解碼器。

資料順序對碼字順序：

須注意解碼器可以解碼一給定的定性順序之碼字。使用平行編碼，碼流所要求之順序是定性的。於是，如果來自平行碼流之碼字能以正確的順序被插入在編碼器，則單一的碼流是足夠的。碼字是根據剛好及時(just-in-time)而以相同的順序被傳送至解碼器。在編碼器，解碼器之模型決定碼字順序，並將碼字封包成單一流。此模型可以是一實際的解碼器。

當資料為可變長度時，將資料傳送至平行解碼元件會發生一問題。欲解開(unpacking)可變長度碼字，需要使用位元移相器以對準碼字。當實施於硬體中時，位元移相器通常是昂貴且/或緩慢的。位元移相器之控制是視特殊

五、發明說明(16)

碼字的尺寸而定。此控制反饋迴路可避免快速地執行可變長度移位。如果解開流之處理是在單一位元移相器中執行，其速度不夠快而無法趕上多解碼器，則無法實現以單一流來反饋多解碼器之優點。

本發明所提出之解決方式，是分離將編碼資料分布至平行編碼器之問題與解碼時可變長度碼字的對準之問題。在各獨立的碼流中之碼字被封包成固字長度的字，稱為插入字。在通道之解碼器端，使用快速固線式資料線及一簡單的控制電路，這些插入字可以被分布至平行解碼器單元。

使插入字長度大於最大碼字長度是方便的，所以至少欲完成一碼字之足夠的位元係包含於各插入字中。插入字可含有許多碼字及部份的碼字。圖4指出一組平行碼流的例子之插入。

依據在解碼器之指令，這些字被插入。各獨立解碼器接收整個插入字。在各解碼器現在局部地做成位元移位操作，而保持系統之平行性。須注意在圖4中，各插入字中之第一碼字為一組中之最低其餘碼字。例如，第一插入字是來自碼流1，從最低碼字(亦即#1)開始。接著是碼流2中之第一插入字，然後才是碼流3中之第一插入字。然而，在已排序的插入字中之並未完全包含的下一個最低碼字為#7。所以，在流中之下一個字為碼流2之第二插入字。

在另一個實施例中，隨後的一組插入字(例如從流1中之碼字#8開始的碼字、從流2中之碼字#7開始的碼字、從

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(17)

流3中之碼字#11開始的碼字)插入於插入流中的順序，是根據前一組插入字的第一碼字(例如從流1中之碼字#1開始的碼字、從流2中之碼字#2開始的碼字、從流3中之碼字#4開始的碼字)，並被排序而成從具有最低數目第一碼字之插入字至具有最高數目第一碼字之插入字。所以，在此情形中，由於從碼字#1開始的插入字是第一個，流1中之下一個插入字為欲被插入於插入流中之第二群插入字的第一個，接著為流2之下一個插入字，然後才是流3中之下一個插入字。須注意在第二群插入字被插入於插入流之後，流2中之下一個插入字將會是插入於流中之下一個插入字，因為碼字#7為第二組插入字的最低碼字(接著為流1中之碼字#8，然後才是流3中之碼字#11)。

使用實際的解碼器作為資料流之模型，可以說明欲產生插入流的全部設計選擇及延遲。對於具有編碼器及解碼器之雙工系統而言，這並不是昂貴的。須注意並不能歸納為任意平行組的可變長度(或不同尺寸)資料字是假設成定性的順序。

平行解碼之位元流產生器與編碼器之型式：

本發明可以使用現存的編碼器，例如Q-編碼器或B-編碼器，作為平行地重複設置之位元流產生元件。然而，亦可以使用其它的碼及編碼器。本發明所使用之編碼器及其相關的碼為簡單的編碼器。

在本發明中，使用具有簡單碼而非複雜碼的位元流產

五、發明說明 (18)

生器，例如 Q-編碼器所使用之算術碼，或 B-編碼器所使用之多狀態碼，可提供許多優點。簡單碼之優點在於硬體實施會更快速且簡單，需要比複雜碼更少的矽。

本發明之另一個優點在於：可以改善編碼效率。使用有限量的狀態資訊之碼，無法理想地符合對於各概率之 Shannon 一致性極限。習知的硬體實施碼可允許單一位元流產生器來處理多概率或上下文，會有降低編碼效率之限制。移除多上下文或多概率等級所需之限制，則可以使用更接近符合 Shannon 一致性極限之碼。

R-碼：

本發明之一個實施例所使用的碼(與編碼器)稱為 R-碼。R-碼為合適碼，其將可變數目的相同輸入符號轉換成一碼字。在一實施例中，R-碼被參數化使得許多不同概率可由單一的解碼器設計來處理。此外，本發明之 R-碼可以由簡單、高速的硬體來解碼。

在本發明中，R-編碼器使用 R-碼來執行編碼或解碼。在一實施例中，R-編碼器為一組合的位元流產生器與概率估計模組。例如，在圖 1 中，R-編碼器可以包括概率估計模組 102 與位元流產生器 103 之組合，及概率估計模組 105 與位元流產生器 106 之組合。

碼字代表最大可能符號(MPS)之運轉(run)。MPS 代表二進位判定之結果超過 50% 的概率。另一方面，最小可能符號代表二進位判定之結果小於 50% 的概率。須注意當此

五、發明說明 (19)

兩個結果為相等的概率時，那一個標示為MPS或LPS並不重要，只要編碼器與解碼器可以作成相同的標示。在壓縮檔中的結果位元順序是表示於表1中，給定的參數稱為MAXRUN。

表 1 - 位元產生編碼

碼字	意指
0	MAXRUN 個連續 MPS
1N	N 個連續 MPS, 接著 LPS, $N < \text{MAXRUN}$

欲編碼時，在運轉中之MPS的數目是由一簡單計數器所計數。如果此計數等於MAXRUN計數值，則碼字0被送入碼流中，且計數器被復置(reset)。如果遇到LPS，則碼字1接著位元N被送入碼流中，其可以獨特地敘述在LPS之前的MPS符號之數目(須注意有許多方式可指定N位元以敘述運轉長度)。再度使計數器復置。須注意N所需之位元的數目是與MAXRUN之值無關。而且須注意亦可以使用1之補數的碼字。

欲解碼時，如果碼流中之第一位元為0，則MAXRUN之

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (20)

值置於MPS計數器中，且LPS指示被清除。然後，位元0被拋棄。如果第一位元為1，則以下的位元被檢查以取出位元N，且適當的計數(N)置於MPS計數器中並設定LPS指示器。然後含有碼字1N之碼流位元被拋棄。R-碼是由表1之規則所產生的。須注意一給定的R-碼 $R_x(k)$ 的定義是由MAX-RUN來定義。例如：

$$\text{MAXRUN for } R_x(k) = x \cdot 2^{k-1},$$

於是

$$\begin{aligned} \text{MAXRUN for } R_2(k) &= 2 \cdot 2^{k-1}, \\ \text{MAXRUN for } R_3(k) &= 3 \cdot 2^{k-1}, \\ \text{etc.....} \end{aligned}$$

須注意R-碼為Golomb碼之子集。亦須注意Rice碼只使用 $R_2(\cdot)$ 碼。本發明之R碼允許使用 $R_2(k)$ 及 $R_3(k)$ 碼，如果需要的話可以使用其它的 $R_n(k)$ 碼。在一實施例中，使用 $R_2(k)$ 及 $R_3(k)$ 碼。須注意 R_n 存在於 $n=2$ 且 n 等於任意奇數（例如 R_2 、 R_3 、 R_5 、 R_7 、 R_9 、 R_{11} 、 R_{13} 、 R_{15} ）。在一實施例中，對於 $R_2(k)$ 碼而言，運轉計數 r 被編碼於 N 中；運數計數 r 是以 k 個位元來敘述，使得 $1N$ 是以 $k+1$ 個位元來表示。而且在一實施例中，對於 $R_3(k)$ 碼而言，位元

(k-1)

(k-1)

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (21)

N 可以含有 1 位元以表示如果 $n < 2$ 或 $n \geq 2$ ，且以 $k-1$ 或 k 個位元來指出運轉計數 r ，使得變數 N 是分別由總數為 k 或 $k+1$ 個位元來表示。在其它的實施例中， N 之 1 的補數可以使用於碼字中。在此情形中，MPS 易於產生含有許多 0 之碼流，而 LPS 易於產生含有許多 1 之碼流。

表 2、3、4、5 敘述使用於本發明之一實施例中的某些有效的 R 碼。須注意在本發明中亦可以使用其它的運轉長度碼。用於 $R2(2)$ 之另一種運轉長度碼的例子表示於表 6 中。表 7 與 8 指出使用於實施例中的碼之例子。

表 2 - $R2(0)$

未編碼資料	碼字
0	0
1	1

表 3 - $R2(1)$

未編碼資料	碼字
00	0
01	10
1	11

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (22)

表 4 - R 3 (1)

未編碼資料	碼字
000	0
001	100
01	101
1	11

表 5 - R 2 (2)

未編碼資料	碼字
0000	0
0001	100
001	101
01	110
1	111

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (23)

表 6 - 另一種 R 2 (2) 表 7 - 另一種 R 3 (2) 碼

另一種方式	R 2 (2)
0000	0
0001	111
001	101
01	110
1	100

最好	R 3 (2)
000000	0
000001	1000
00001	1010
0001	1001
001	1011
01	110
1	111

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (24)

表 8 - 另一種 R 2 (2) 碼

最好	R 2 (2)
0000	0
0001	100
001	110
01	101
1	111

R碼之概率估計模組：

在一實施例中，R2(0)不執行編碼：輸入0被編碼成0且輸入1被編碼成1(反之亦然)，概率等於50%是最好的。目前較佳實施例之R2(1)碼，概率最好接近0.707(亦即70.7%)，且R3(1)最好為0.794概率(79.4%)。R2(2)碼最好為0.841概率(84.1%)。以下表9敘述幾近最佳運轉長度碼，這裡概率歪斜(probability skew)是由以下的公式來定義：

$$\text{概率斜歪} = - \log_2 (LPS)$$

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (25)

表 9

概 率	概 率 歪 斜	最 佳 Golomb 碼
. 500	1. 00	R2(0)
. 707	1. 77	R2(1)
. 841	2. 65	R2(2)
. 917	3. 59	R2(3)
. 958	4. 56	R2(4)
. 979	5. 54	R2(5)
. 989	6. 54	R2(6)
. 995	7. 53	R2(7)
. 997	8. 53	R2(8)
. 999	9. 53	R2(9)

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (26)

須注意當由概率歪斜所指出的概率範圍涵蓋非常均勻的空間時碼是最佳的，雖然最佳概率在較高的k值與在較低的k值並不會相差很多。

參照R碼為最佳的概率。實際上，只有R2(2)符合一致性曲線。實際考慮什麼範圍的概率為一特殊的R-編碼器，優於一給定的等級中之其它全部的R-編碼器。下列表提供R2碼之等級及R2與R3碼之等級的範圍。

對於從0至12的R2碼之等級而言，其範圍是在以下的表10中。例如，當只有使用R2碼， $0.50 \leq \text{概率} \leq 0.6180$ 時R2(0)是最好的。同樣地，當 $0.6180 \leq \text{概率} \leq 0.7862$ 時R2(1)是最好的。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (27)

表 1 0 - R 2 碼 0 至 1 2

碼	概 率
R2(0)	0.6180
R2(1)	0.7862
R2(2)	0.8867
R2(3)	0.9416
R2(4)	0.9704
R2(5)	0.9851
R2(6)	0.9925
R2(7)	0.9962
R2(8)	0.9981
R2(9)	0.9991
R2(10)	0.9995
R2(11)	0.9998
R2(12)	—

對於 R2 與 R3 碼，解決方式是在以下的表 11 中。例如，
當使用 R2 與 R3 碼， $0.6180 \leq \text{概 率} \leq 0.7549$ 時 R2(1) 是最好
的。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (28)

表 1 1 - R 2 與 R 3 碼

長度小於或等於 1 3 位元

碼	概 率
R2(0)	0.6180
R2(1)	0.7549
R3(1)	0.8192
R2(2)	0.8688
R3(2)	0.9051
R2(3)	0.9321
R3(3)	0.9541
R2(4)	0.9655
R3(4)	0.9754
R2(5)	0.9826
R3(5)	0.9876
R2(6)	0.9913
R3(6)	0.9938
R2(7)	0.9956
R3(7)	0.9969
R2(8)	0.9978
R3(8)	0.9984
R2(9)	0.9989
R3(9)	0.9992
R2(10)	0.9995
R3(10)	0.9996
R2(11)	0.9997
R3(11)	0.9998
R2(12)	—

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (29)

對於固定 k 而言 $R2(k)$ 稱為運轉長度碼。然而，固定的 k 只有對於一範圍的概率而言是最好的。須注意當幾近最佳概率而編碼時，依據本發明之 R -碼以幾乎相等的頻率來使用碼字 0 與 1N。換句話說，一半的時間，本發明之 R -編碼器輸出一碼，而另一半的時間， R -編碼器則輸出另一碼。藉由檢查碼字 0 與 1N 之數目，可以做成是否使用最佳碼的判定。亦即，如果正在輸出太多個碼字 1N，則運轉長度太長；另一方面，如果正在輸出太多個碼字 0，則運轉長度太短。

Langdon 所使用之概率估計模型檢查各碼字之第一個位元，以決定源概率是在目前估計之上或之下。參照 G.G. Langdon, "An Adaptive Run-Length Coding Algorithm", IBM Technical Disclosure Bulletin, Vol 26, No. 7 B, Dec. 1983。根據此判定， k 被增加或減小。例如，如果看到表示 MPS 之碼字，則概率估計太低。所以，依據 Langdon，對於各碼字 0， k 被增加 1。如果在 LPS (例如碼字 1N) 之後看到表示小於 MAXRUN MPS 之碼字，則概率估計太高。所以，依據 Langdon，對於各碼字 1N， k 被增加 1。

本發明允許比對於每一個碼字使 k 簡單增加或減少 1 更為複雜的概率估計。本發明包括一概率估計模組狀態，其決定欲使用之碼。許多狀態可以使用相同的碼。碼被指定給使用一狀態表或狀態機器之狀態。

在本發明之一個實施例中，概率估計改變每一個碼字輸出之狀態。於是，概率估計模組視碼字是以 0 或 1 開始而

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (30)

定來增加或減小概率估計。例如，如果碼字"0"被輸出，則發生MPS概率之估計的增加。另一方面，如果碼字"1"被輸出，則MPS概率之估計被減小。

習知技術之Langdon編碼器只使用 $R2(k)$ 碼，且對於各碼字增加或減小 k 。本發明則是使用 $R2(k)$ 及 $R3(k)$ 碼，連同狀態表或狀態機器，以允許調整適應速率來應用。亦即，如果有小量的固定資料，則適應必須更快以產生最佳的編碼，且如果有更大量的固定資料，則適應時間必須更長使得可以選擇編碼以達成其餘資料之更好的壓縮。須注意在會發生狀態之可變數目改變的情形，應用之特定特徵亦會影響適應速率。由於 R -碼之特性， R -碼之估計很簡單且只需要少量的硬體，卻仍非常具有威力。圖33指出編碼效率(相對於一致性使碼長度正常化)對MPS概率之圖形。圖33指出本發明之某些 R -碼如何覆蓋概率空間。作為一個例子，圖33指出MPS概率約為0.55， $R2(0)$ 碼之效率為1.01(或較差1%)一致性極限(entropy limit)。相反地， $R2(1)$ 碼具有效率為1.09(或較差9%)一致性極限。此例子指出對於特別低概率之情形而言，使用錯誤碼會引起編碼效率之8%損失。

$R3(k)$ 碼之併入允許更多的概率空間能以更大的效率被覆蓋。依據本發明之概率估計狀態表的例子係表示於圖5中。參照圖5，概率估計狀態表指出與表中之個別狀態有關的碼及狀態計數器。須注意此表包括正與負狀態。所示之表具有37個正狀態與37個負狀態，包括零狀態。負狀態

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (31)

表示與正狀態不同的MPS。在一個實施例中，當MPS為1時可使用負狀態，當MPS為0時可使用正狀態，或反之亦然。須注意圖5所示之表只是一個例子，其它的表可以具有較多或較少的狀態及不同的狀態配置。

開始時，編碼器是在狀態0(亦即沒有碼)，其為概率估計等於0.50之 $R2(0)$ 碼。在各碼字被處理之後，狀態計數器視碼字之第一個位元而定被增加或減小。在一實施例中，碼字0增加狀態計數器之數量(magnitude)；以1開始之碼字則減小狀態計數器之數量。所以，各碼字以狀態計數器來引起狀態之改變。換句話說，概率估計模組改變狀態。然而，連續狀態可以與相同的碼結合。在此情形中，可以完成概率估計而不需要改變各碼字之碼。換句話說，對於各碼字改變狀態；然而，此狀態是以固定的次數被映射(mapped)成相同概率。例如，狀態5至-5全部使用 $R2(0)$ 碼，而狀態6至11及-6至-11均使用 $R2(1)$ 碼。使用本發明之狀態表，能以非線性之方式而允許概率估計停留在相同的編碼器。

須注意對於較低的概率而言，包括更多個具有相同R-碼之狀態。這是因為當以低概率來使用錯誤碼時，效率之損失很大。運轉長度碼狀態表之特性，是在各碼字之後轉移於狀態之間。在一狀態表中，其被設計成隨著狀態中之各變化而改變碼，當接於較低概率的狀態之間，此碼是接於兩個碼之間，其中一個碼是非常接近一致性極限，而另一個碼則是遠離一致性極限。於是，一障礙penalty(編碼

五、發明說明 (32)

後的資料位元之數目)會導致狀態間之轉變。習知的概率估計模組，例如 Langdon 之概率估計模組，會由於此障礙而損失性能。

在較高概率之運轉長度碼中，在錯誤碼中之障礙並不是很大。所以，在本發明中，額外的狀態是以較低的概率被加入，使得兩個正確碼之間的連接改變被增加，藉以降低編碼失效。

須注意在某些實施例中，編碼器可以具有啓始的概率估計狀態。換句話說，編碼器能以預定的一個狀態開始，例如狀態 18。在一實施例中，可以使用不同的狀態表，使得第一少數符號可以使用某些狀態，以允許快速適應，而對於其餘的符號可以使用第二狀態表而減緩適應，以允許概率估計之細微調整。以此方式，編碼器能以較快的編碼處理來使用更有效的碼。在另一個實施例中，碼流可以指定各上下文之啓始概率估計。在一實施例中，並沒有以固定數目(例如 1)來做成增加或減少。而是能以一可變數目來增加概率估計狀態，此可變數目是依據已遇到的資料之量或資料中之改變量(穩定性)。此種表的例子為以下所述之表 21-25。

如果狀態表是對稱的，如同圖 5 之範例表所示，只有一半的狀態(包括 0 狀態)需要被儲存或實施於硬體中。在一實施例中，狀態數目是利用對稱性而以正負號大小(1s)補數形式來儲存。以此方式，藉著取補數之絕對值而使用此表來決定狀態，其檢查正負號以決定 MPS 為 1 或 0。因為

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (33)

使用狀態之絕對值來標示此表，且狀態補數之絕對值的計算是不重要的，所以允許用於增加或減少狀態所需之硬體可以降低。在另一實施例中，爲了更高的硬體效率，能以固線式或可程式的狀態機器來取代狀態表。碼轉換器之固線式狀態爲狀態表的一種實施方式。

平衡的平行一致性編碼系統：

本發明提供一種平衡的平行一致性編碼系統。此平行一致性編碼系統包括即時編碼及執行於高速/低價硬體中的即時編碼。本發明可以使用於許多無損耗的編碼應用，包括但不限於：可寫光碟或磁碟資料之壓縮/解壓縮、計算機網路資料之即時壓縮/解壓縮、多功能機器(例如影印機、傳真機、掃描器、印表機等)中之壓縮圖框儲存中的影像資料的即時壓縮/解壓縮、及音頻資料的即時壓縮/解壓縮。

指定編碼器之性能時須注意，其是以直線前進之方式設計，且編碼器可以達成在非常快速的編碼資料通道中所給定的原始資料之某種速率。然而在許多應用中，目的在於使編碼器可以有效地利用編碼資料通道。編碼資料通道利用會受到原始資料介面之最大猝發速率、編碼器速度、及資料上所達成之壓縮影響。在某些局部量的資料，其是視編碼器中之緩衝量而定，必須考慮這些效果的影響。最好具有一種編碼器，其可以有效地利用編碼資料通道，而仍保持速度與高壓縮，並仍可容納最大猝發速率。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (34)

以下敘述描述本發明之編碼器。亦敘述能與編碼器一起使用之解碼器。

本發明中之即時編碼：

圖6為本發明之編碼系統的方塊圖。在一實施例中，本發明之編碼器執行即時編碼。參照圖6，編碼系統600包括一編碼器602，耦合至上下文模型(CM)&狀態記憶603，響應原始資料601用於產生碼字資訊604之形式的碼資訊。碼字資訊604被重序單元606接收，此重序單元606耦合至重序記憶器607。響應碼字資訊604，重序單元606配合重序記憶器607而產生編碼資料流608。須注意編碼系統600並不限於碼字之操作，在其它的實施例中可以利用本發明之揭示而操作各別的類比波形、可變長度位元圖型、通道符號、英文字母、事件等。

編碼器602包括上下文模型(CM)、概率估計機器(PEM)及位元流產生器(BG)。編碼器602中之上下文模型及PEM(概率估計機器)大致是與解碼器中者相同(除了資料流的方向)。編碼器602之位元產生器是與解碼器位元產生器類似，且將敘述如下。編碼器602之編碼結果為零或更多的位元(表示原始資料)之輸出。在一實施例中，位元流產生器之輸出亦包括一個或一個以上的控制信號。這些控制信號提供一控制路徑給位元流中之資料。在一實施例中，碼字資訊可以包含運轉指示之開始、運轉指示之結束、碼字、及辨識運轉計數(是否其為上下文或概率等級)之各碼字的

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (35)

索引(index)。本發明之位元流產生器的一個實施例將敘述如下。

重序單元606接收由編碼器602之位元流產生器所產生的控制信號與位元，並產生編碼資料。在一實施例中，由重序單元606所輸出之編碼資料包含插入字流。

在一實施例中，重序單元606執行兩種功能。重序單元606將碼字從編碼器所產生的運轉結束移動至解碼器所需之運轉開始，並將可變長度碼字組合成固定長度插入字，且以解碼器所需之適當順序而將它們輸出。

重序單元606使用一暫時重序計憶器607。在一實施例中，於一工作站上執行編碼，暫時重序計憶器607之尺寸可以超過100百萬位元組(Megabytes)。在本發明之平衡系統中，暫時重序計憶器607是非常小(例如約為1仟位元組)而且是固定的。於是，在一實施例中，使用固定量的記憶體來執行即時編碼，即使如此會增加解碼器所需之記憶體或位元率(例如當在運轉完成之前要作成輸出時)。以下將敘述本發明之解碼器可以決定重序單元的有限記憶體之效果，例如使用隱含(implicit)、顯示(explicit)或內流(instream)發信(signaling)。重序單元606含用於重序之有限的記憶體，但是需要的記憶是無限的。必須考慮用於運轉的結束至運轉佇列的開始及用於插入字重序之有限記憶體的效果。

在一實施例中，本發明之編碼系統(及對應的解碼系統)使用單一積體電路晶片來執行編碼(或解碼)。在另一

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (36)

實施例中，單一積體電路包含本發明之編碼器系統（包括其編碼器、解碼器及記憶體）。可以加入各別的外部記憶體來幫助編碼。多晶片模組或積體電路可以包含編碼/解碼硬體及記憶體。

本發明之編碼系統可增加有效的頻帶寬度至因數 N 。如果所達成的壓縮是小於 $N:1$ ，則編碼資料通道將被完全地使用，但是所達成的有效頻帶寬度增加只等於壓縮率。如果所達成的壓縮大於 $N:1$ ，則可達成有效頻帶寬度具有可寫的額外頻帶寬度。在此兩種情形中，所達成的壓縮必須超過由編碼系統中出現的緩衝量所定義的資料之局部區域。

本發明之編碼器的位元產生器：

圖7指出本發明之編碼器位元產生器的一個實施例。位元產生器701被耦合以接收概率等級及未編碼位元（例如MPS或LPS指示）作為輸入。響應這些輸入，位元產生器701輸出多信號。兩個輸出為控制信號其指示運轉之啓始及運轉之結束（各碼字代表一運轉），分別為開始信號711與結束信號712。對於運轉而言在相同時間開始及結束是可能的。當一運轉開始或結束時，“索引”輸出713包含未編碼位元之概率等級（或上下文）的指示。在一實施例中，索引輸出713代表系統之觸排指示與位元之概率等級的結合，其中各概率等級折返於記憶器的數個觸排中。當運轉結束時，使用碼字輸出714以從位元產生器701輸出一碼字

五、發明說明 (37)

。 記憶器 702 被耦合至位元產生器 701，並含有一給定的概率等級之運轉計數。於位元產生期間，位元產生器 701 使用索引(例如概率等級)而從記憶器 702 讀取。在從記憶器 702 讀取之後，位元產生器 701 執行位元產生如下。首先，如果運轉計數等於零，則開始信號 711 被確定為指示一運轉之開始。然後，如果未編碼位元等於 LPS，則結束信號 712 被確定指示運轉之結束。而且如果未編碼位元等於 LPS，則碼字輸出 714 被設定以指示碼字為 1N 碼字且運轉計數被清除，例如設定為零(由於其為運轉結束)。如果未編碼位元並不等於 LPS，則運轉計數被增加且一測試決定是否運轉計數等於碼之最大運轉計數。如果如此，則結束信號 712 被確定，碼字輸出 714 被設定為零且運轉計數被清除(例如運轉計數被設定為零)。如果此測試決定運轉計數並不等於碼之最大值，則運轉計數被增加。須注意索引信號 713 代表所接收的概率等級作為一輸入。

在本發明中，執行 1N 碼字之產生，使得不需要額外資訊而可以決定其長度。表 12 指出解碼器與編碼器之碼字 R 3(2) 之 1N 碼字表示。解碼器預期 "1N" 碼字中的 "1" 位元為 LSB，且 "N" 計數部份是以適當的 MSB...LSB 順序。在解碼器順序中，不知道使用特別碼的情形下，無法區分可變長度碼字與零填料(zero padding)。在編碼器順序中，碼字被反向(reversed)，大部份的 "1" 位元之位置指示 "1N" 碼字之長度。欲產生編碼器順序的碼字，計數值之補數必

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (38)

須被反向。其達成方式是藉由使13-位元計數反向，然後將它移位使其與LSB對齊。如以下所詳細敘述，位元包封單元使碼字反向而成解碼器順序。然而，碼字之此反向並不會引起位元包封單元606之複雜性的增加，由於它必須執行移位。

表 1 2 - R 3 (2) 碼字之 " 1 N " 碼字表示

未編碼 資料	碼字	計數 值 反相	解碼器順序	編碼器順序
			(計數值加底線)	
000000	0		00000000000000	00000000000000
000001	1000	00	0000000000 <u>00</u> 01	000000000010 <u>00</u>
00001	1010	01	0000000000 <u>01</u> 01	000000000010 <u>10</u>
0001	1001	10	0000000000 <u>10</u> 01	000000000010 <u>01</u>
001	1011	11	0000000000 <u>11</u> 01	000000000010 <u>11</u>
01	110	0	0000000000 <u>00</u> 11	000000000011 <u>00</u>
1	111	1	0000000000 <u>01</u> 11	000000000011 <u>01</u>

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (39)

對於 R3 碼而言，產生 "N" 碼字亦需要在接著 "1" 之後的位元指出表示一短的或長的計數。

藉著使用記憶器之多觸排，本發明允許疊流 (pipelining)。例如，在多阜記憶器的情形中，對於未編碼位元而言記憶器想到一讀取操作，而對於先前的未編碼位元而言記憶器想到一寫入操作。

可變 AHDL 取樣設計：

本發明之編碼器位元產生器的一個實施例包含 FPGA。此設計處理全部的 R2 及 R3 到 R2(12)。以下列出 AHDL (可變硬體敘述語言) 源碼。

如圖 13 所示，此設計包含多部份。首先，"ENCBG" 1301 為此設計的主要部份，其具有邏輯可處理運轉之開始、結束與連續。第二，使用 "KEXPAND" 1302 而將概率等級展開成最大運轉長度、可變長度罩、及 R3 碼之第一長碼字的長度。"KEXPABD" 1302 是與具有相同名稱的解碼器功能相同。第三，"LPSCW" 1303 部份在概率等級附近取一計數值與資訊作為輸入，並產生適當的 "1N" 碼字。

此設計使用兩個疊流級 (pipeline stage)。於第一疊流級期間，計數被增加，概率等級被展開，且執行長 R3 碼字之減法及比較。全部其它的操作則是在第二疊流級期間執行。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (40)

encbg.tdf

TITLE "Bit Generator for the encoder";

INCLUDE "kexpand.inc";

INCLUDE "lpscw.inc";

SUBDESIGN encbg

(

k[3..0],

r3,

bit,

count_in[12..0],

clk

: INPUT;

start_run,

end_run,

index[4..0],

count_out[12..0],

codeword[12..0]

: OUTPUT;

)

VARIABLE

k_q[3..0],

r3_q,

k_qq[3..0],

r3qq,

bit_q,

bit_q,

count_in_q[12..0],

start_run,

end_run,

start_run_q,

index[4..0],

count_out[12..0],

count_plus[12..0],

max_r1[12..0],

codeword[12..0]

: DFF;

kexpand_

: kexpand;

lpscw_

: lpscw;

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (41)

```

BEGIN
    lpscw_.clk      = clk;
    k_q[].clk       = clk;
    r3_q.clk        = clk;
    k_qq[].clk      = clk;
    r3_qq.clk       = clk;
    bit_q.clk       = clk;
    bit_qq.clk      = clk;
    count_in_q[].clk = clk;
    start_run.clk   = clk;
    end_run.clk     = clk;
    start_run_q.clk = clk;
    index[].clk     = clk;
    count_out[],clk = clk;
    count_plus[].clk = clk;
    max_rl[].clk    = clk;
    codeword[].clk  = clk;

    k_q[]           = k[];
    r3_q            = r3;
    k_qq[]          = k_q[];
    r3_qq           = r3_q;
    bit_q           = bit;
    bit_qq          = bit_q;
    count_in_q[]    = count_in[];
    count_plus[]    = count_in_q[] + 1;
    start_run       = start_run_q;

    stat_run_q      = (count_in_q[] == 0);
    index[0]        = r3_qq;
    index[4..1]     = k_qq[];

    kexpand_.k_reg[] = k_q[];
    kexpand_.r3_reg  = r3_q;
    lpscw_.r3        = r3_q;
    lpscw_.k_q[]     = k_q[];
    lpscw_.r3_q      = r3_qq;
    lpscw_.count[]   = count_in_q[];
    lpscw_.mask[]    = kexpand_.mask[];
    lpscw_.r3_split[] = kexpand_.r3split[];
    lpscw_.maxrl_q[] = max_rl[];
    max_rl[]         = kexpand_.maxrl[];

    IF (bit_qq) THEN                                     % LPS %
        end_run      = VCC;
        count_out[]  = 0;
        codeword[]   = lpscw_.cw[];

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (42)

```

    ELSIF (count_plus[] == max_r1[]) THEN
        end_run = VCC;
        count_out[] = 0;
        codeword[] = 0;
    ELSE
        end_run = GND;
        count_out[] = count_plus[];
        codeword[] = 0;
    END IF;
END;

```

lpscw.tdf

SUBDESIGN lpscw
(

```

    r3,
    k_q[3..0],
    r3_q,
    count[12..0],
    mask[11..0],
    r3_split[10..0],
    maxr1_q[12..0],
    clk
                                : input;
    cw[12..0]
                                : output;
)

```

VARIABLE

```

    temp[12..0] : NODE;
    temp_rev[12..0] : NODE;
    temp_sh[12..0] : NODE;
    split[11..0] : NODE;
    r3_long : DFF;
    count_minus[11..0] : DFF;
    mask_q[11..0] : DFF;
    count_q[12..0] : DFF;

```

BEGIN

```

    r3_long.clk = clk;
    count_minus[].clk = clk;
    mask_q[].clk = clk;
    count_q[].clk = clk;
    split[10..0] = r3_split[];
    split[11] = GND;
    r3_long = (r3) AND (count[11..0] .> split[]);
    count_minus[] = count[11..0] - split[];
    mask_q[] = mask[];
    count_q[] = count[];

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

訂

五、發明說明 (43)

% ————— pipeline stage ————— %

```
IF (r3_long) THEN
    temp[11..0] = (count_minus[]) XOR mask_q[];
ELSE
    temp[11..0] = count_q[11..0] XOR mask_q[];
END IF;
temp[12] = GND;
```

```
temp_rev[0] = temp[12];
temp_rev[1] = temp[11];
temp_rev[2] = temp[10];
temp_rev[3] = temp[9];
temp_rev[4] = temp[8];

temp_rev[5] = temp[7];
temp_rev[6] = temp[6];
temp_rev[7] = temp[5];
temp_rev[8] = temp[4];
temp_rev[9] = temp[3];
temp_rev[10] = temp[2];
temp_rev[11] = temp[1];
temp_rev[12] = temp[0];
```

CASE k_q[] IS

```
WHEN 0 => temp_sh[] = 0;
WHEN 1 => temp_sh[0] = temp_rev[12];
           temp_sh[12..1] = 0;
WHEN 2 => temp_sh[1..0] = temp_rev[12..11];
           temp_sh[12..2] = 0;
WHEN 3 => temp_sh[2..0] = temp_rev[12..10];
           temp_sh[12..3] = 0;
WHEN 4 => temp_sh[3..0] = temp_rev[12..9];
           temp_sh[12..4] = 0;
WHEN 5 => temp_sh[4..0] = temp_rev[12..8];
           temp_sh[12..5] = 0;
WHEN 6 => temp_sh[5..0] = temp_rev[12..7];
           temp_sh[12..6] = 0;
WHEN 7 => temp_sh[6..0] = temp_rev[12..6];
           temp_sh[12..7] = 0;
WHEN 8 => temp_sh[7..0] = temp_rev[12..5];
           temp_sh[12..8] = 0;
WHEN 9 => temp_sh[8..0] = temp_rev[12..4];
           temp_sh[12..9] = 0;
WHEN 10 => temp_sh[12..10] = 0;
           temp_sh[12..10] = 0;
WHEN 11 => temp_sh[10..0] = temp_rev[12..2];
           temp_sh[12..11] = 0;
WHEN 12 => temp_sh[11..0] = temp_rev[12..1];
           temp_sh[12] = GND;
```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (44)

```

END CASE;

IF (NOT r3_q) THEN                                % R2 %
    cw[] = temp_sh[] OR maxrl_q[] ;
ELSIF (NOT r3_long) THEN                          % R3 SHORT %
    cw[11..0] = temp_sh[12..1] OR maxrl_q[11..0] ;
    cw[12] = GND;
ELSE                                              % R3 LONG %
    cw[12..1] = temp_sh[12..1] OR
                (maxrl_q[11..0] AND NOT mask_q[11..0]) ;
    cw[0] = temp_sh[0] ;
END IF;
END;

```

kexpand.tdf

TITLE "decoder, k expand logic" ;

SUBDESIGN kexpand

```

(
    k_reg[3..0] ,
    r3_reg

                                : input;

    maxrl[12..0] ,
    mask[11..0] ,
    r3split[10..0]

                                : output;
)

```

BEGIN

```

TABLE
    k_reg[], r3_reg    => maxrl[], mask[], r3split[] ;

    0,    0    => 1,          0,    X;
    1,    0    => 2,          1,    X;
    1,    1    => 3,          1,    1;
    2,    0    => 4,          3,    X;
    2,    1    => 6,          3,    2;
    3,    0    => 8,          7,    X;
    3,    1    => 12,         7,    4;
    4,    0    => 16,         15,   X;
    4,    1    => 24,         15,   8;

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

外

五、發明說明 (45)

5,	0	= >	32,	31,	X;
5,	1	= >	48,	31,	16;
6,	0	= >	64,	63,	X;
6,	1	= >	96,	63,	32;
7,	0	= >	128,	127,	X;
7,	1	= >	192,	127,	64;
8,	0	= >	256,	255,	X;
8,	1	= >	384,	255,	128;
9,	0	= >	512,	511,	X;
9,	1	= >	768,	511,	256;
10,	0	= >	1024,	1023,	X;
10,	1	= >	1536,	1023,	512;
11,	0	= >	2048,	2047,	X;
11,	1	= >	3072,	2047,	1024;
12,	0	= >	4096,	4095,	X;

END TABLE;

END;

本發明之重序單元：

圖8為重序單元的一個實施例之方塊圖。參照圖8，重序單元606包含一運轉計數重序單元801及一位元包封單元802。運轉計數重序單元801將碼字從由編碼器所產生的運轉結束移動至解碼器所需要的運轉開始，而位元包封單元802將可變長度碼字與固定長度插入字結合，並將它們以解碼器所需的適當順序輸出。

可使用一"snooper"解碼器來重序任意解碼器，其中一解碼器包括於一編碼器中，並提供要求資料做成真實解碼器所需之碼字的順序。欲支援一snooper解碼器，對於各流必須獨立地完成運轉計數之重序。對於可容易地模組

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

訂

五、發明說明 (46)

化之解碼器而言，可以使用多時標記佇列(multiple time stamped queue)或單一合併佇列以允許重序。在一實施例中，使用佇列狀資料可以完成重序各碼字，而且是與多編碼資料流之使用無關。給定如何執行重序之敘述如下。

在編碼器中所執行的第一重序操作是欲重序各運轉計數，使得運轉計數在運轉開始時(當解碼器需要解碼器)被指定。需要此重序，因為編碼器並沒有決定運轉計數(及碼字)是什麼，直到運轉結束。於是，從編碼資料所產生的結果運轉計數被重序，使得解碼器可以適當地解碼運轉計數而成資料流。

參照圖 8，本發明之重序單元 606 包含運轉計數重序單元 801 及位元包封單元 802。運轉計數重序單元 801 被耦合以接收多輸入，此多輸入包括開始信號 711、結束信號 712、索引信號 713 及碼字 714。連同圖 9 之運轉計數重序單元，將更詳細地敘述這些信號。響應這些輸入，運轉計數重序單元 801 產生碼字 803 及信號 804。信號 804 指出什麼時間要復置運轉計數。碼字 803 被位元包封單元 802 接收。響應碼字 803，位元包封單元 802 產生插入字 805。

運轉計數重序單元 801 及位元包封單元 802 將進一步詳細敘述如下。

運轉計數重序單元：

如前所述，解碼器在資料被需要的碼字編碼的開始時接收碼字。然而，編碼器並不知道碼字之辨識，直到被碼

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (47)

字編碼的資料結束。

運轉計數重序單元801之一個實施例的方塊圖敘述於圖9中。所敘述的實施例容納四個插入流，各插入字為16位元，且碼字之長度為1至13位元。在此情形中，重序單元300可以被疊流(pipelined)以處理全部的流。此外，使用連接運轉計數與概率等級之編碼器，使得可以在任何時間作用之運轉計數的最大數是很小，且對於此實施例而言假設為25。須注意本發明並不限於四個插入流、16位元的插入字、或1至13位元的碼字長度，而可以使用更多或更少的流，具有比16位元更多或更少的插入字，且碼字長度可以從1位元延伸至超過13位元。

參照圖9，耦合一指標記憶器901以接收索引輸入713，並產生一位址輸出其被耦合至多工器(MUX)902的一個輸入。MUX902之兩個其它的輸入被耦合以接收來自頭計數器903以頭指標之形式的位址，及來自尾計數器904以尾指標之形式的位址。MUX902之輸出為一位址被耦合並使用以存取一碼字記憶器908。

索引輸入713亦被耦合而作為MUX905之輸入。MUX905之另一個輸入被耦合至碼字輸入714。MUX905之輸出被耦合至有效偵測模組906的輸入及資料匯流排907。資料匯流排907被耦合至控制模組909的輸出。開始輸入711及結束輸入712被耦合至控制模組909的各別輸入。有效偵測模組906之輸出包含碼字輸出803及信號804(圖8)。運轉計數重序單元801亦包含控制器邏輯(本發明中並未示出)，以配

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(48)

合運轉計數重序單元801之不同零件的操作。

索引輸入713辨識一運轉以反復操作。在一實施例中，索引指出25個概率等級的其中之一。在此一情形中，需要5個位元以表示索引。須注意如果使用概率等級之多觸排，則將需要額外的位元以指定特別的觸排。在一實施例中，索引輸入辨識運轉計數之概率等級。碼字輸入714為碼字，當運轉結束發生或者為"不管(don't care)"。開始輸入711及結束輸入712為控制信號其表示運轉是否開始、結束、或兩者均發生。當運轉是由單一未編碼位元組成時，運轉同時開始及結束。

運轉計數重序單元801響應其輸入信號而重序由位元產生器所產生的運轉計數。碼字記憶器908儲存重序期間的碼字。在一實施例中，碼字記憶器908是大於在一次可以作用的運轉計數之數目。如此會導致較好的壓縮。如果碼字記憶器是小於在一次可以作用的運轉計數之數目，如此將會實際地限制作用的運轉計數之數目成為記憶器中所能保持的數目。在可提供良好壓縮的系統中，常常發生當具有長的運轉計數之碼字的資料正被計算時，具有短的運轉計數之許多碼字將會開始(或許結束)。如此需要具有大的記憶器以避免在完成之前輸出長的運轉。

指標記憶器901儲存在運轉之中間的概率等級之碼字記憶器位置的位址，及在隨機存取方式之碼字記憶器908的位址。對於可以是在運轉之中間的各概率等級，指標記憶器901具有碼字記憶器908中的位址之儲存位置。對於特

五、發明說明 (49)

別的概率等級一旦完成運轉，則使用此概率等級之指標記憶器901中所儲存的位址，以存取碼字記憶器908，且完成的碼字在此位置被寫入碼字記憶器908。直到此時，碼字記憶器908中的位置含有一有效進入(valid entry)。於是，指標記憶器901儲存各運轉計數之有效碼字的位置。

頭計數器903及尾計數器904亦提供位址以存取碼字記憶器908。使用頭計數器903及尾計數器904，允許碼字記憶器908可被定位址而作為佇列或圓形緩衝器(例如[FIFO]記憶器)。尾指標904含有碼字記憶器908中下一個有用位置的位址，以允許碼字插入於碼字記憶器908中。頭計數器903含有欲被輸出的下一個碼字之碼字記憶器908中的位址。換句話說，頭計數器903含有欲從碼字記憶器908刪除的下一個碼字之碼字記憶器位址。使用指標記憶器901中各可能索引(例如概率等級)之位置，以想起當運轉開始時尾指標904是在何處，使得當運轉結束時適當的碼字可以被置於碼字記憶器908的此位置中。

控制模組909產生一有效信號作為碼字記憶器中所儲存的部份資料，以指出是否一進入儲存有效的碼字資料。例如，如果有效位元是在邏輯1，則碼字記憶器位置包含有效資料。然而，如果有效位元是在邏輯0，則碼字記憶器位置包含無效資料。有效偵測模組907決定是否每次從碼字記憶器809讀取一碼字時，記憶器位置包含一有效碼字。在一實施例中，有效偵測模組907偵測是否記憶器位置具有一有效碼字或一特殊無效碼。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (50)

當開始一新運轉時，一無效資料進入被置於碼字記憶器908中。此無效資料進入作用為碼字記憶器908中所儲存的資料流中之空間保存器(space holder)，使得當運轉已完成時，運轉之碼字可在正確位置中被儲存於記憶器中(以確保至模型及解碼器的適當順序)。在一實施例中，無效資料進入包括經MUX905之索引及來自控制模組909之無效指示(例如無效位元)。碼字記憶器908中儲存無效進入之位址是由尾指標904所給定，且隨後儲存於指標記憶器901中作為提醒碼字記憶器908中的運轉計數之位置。資料提醒出現於碼字記憶器908中的頭指標903與尾指標904之間，作為完成運轉計數(例如，重序運轉計數)。無效記憶器位置的最大數目為0至1-1，這裡1為運轉計數之數目。當碼字在運轉結束完成時，運轉計數使用指標記憶器901中所儲存的位址而被填入於碼字記憶器908中。

當運轉開始時，運轉之索引被儲存於碼字記憶器908中，所以如果碼字記憶器908是滿的，但是運轉尚未完成，則使用索引連同信號804以復置對應的運轉計數。除了將碼字或指引(indice)儲存於碼字記憶器908之外，使用一位元(下文中稱為"有效"位元)以指示兩種型式的資料何種被儲存。

如果沒有開始或結束運轉，則運轉計數重序單元是閒置的(idle)。如果開始運轉且並沒有結束運轉，及如果記憶器是滿的，則一碼字從碼字記憶器908被輸出。所輸出的碼字為此概率等級之在包含於頭指標903的位址所儲存

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (51)

的碼字。然後，如果開始運轉且並沒有結束運轉（不管記憶器是否滿），則索引輸入 713 經由 MUX 905 在尾指標 904 所標示的位址被寫入碼字記憶器 908。然後尾指標 904 在索引輸入 713 所標示的位址被寫入指標記憶器 901（例如，在概率等級之指標記憶器 901 中的位置）。在寫入尾指標 904 之後，尾指標 904 被增加。

如果結束運轉且並沒有開始運轉，則儲存在對應索引（概率等級）之指標記憶器 901 中的位址被讀出，並被使用作為碼字記憶器中之位置以將完成的碼字儲存在碼字輸入 714 上。

如果開始運轉且結束運轉（亦即，同時開始並結束運轉），且記憶器是滿的，則碼字從碼字記憶器 908 被輸出。然後，如果開始運轉並結束運轉（不管是否記憶器滿），則碼字輸入 714 在尾指標 904 所指定的位址被寫入碼字記憶器 908。然後尾指標 904 被增加以得到下一個有用的位置（亦即，增加 1）。

在本發明中，運轉計數重序單元 801 能以不同次數輸出碼字。在一實施例中，碼字可以被輸出，當它們是有效或無效。當無效時如果存在記憶器滿的情況且運轉並未完成，則可輸出碼字。無效碼字亦可以被輸出以保持最小速率（亦即速率控制）。而且，無效碼字可以被輸出至平碼字記憶器 908，當全部資料已進行運轉計數重序，或當運轉計數重序單元由於復置操作而跳至碼字記憶器 908 的中間。須注意在此情形中，解碼器必須知道編碼器正以此種方

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (52)

式操作。

如上所述，當碼字記憶器 908 是滿的，則輸出一碼字。一旦記憶器是滿的，當做成一輸入（亦即，開始一新碼字）至碼字記憶器 908 時，做成來自碼字記憶器 908 之輸出。須注意對進入之更新並不會引起來自碼字記憶器 908 之輸出，當存在記憶器滿的狀況。亦即，在將結果碼字寫入其先前指定的記憶器位置之後的運轉完成，並不會引起發生記憶器滿輸出。同樣地，當運轉結束且指標記憶器 901 中的對應位址與頭計數器 903 中的位址是相同的，則碼字可以馬上被輸出，然後頭計數器 903 可以被增加而不需存取碼字記憶器 908。在一實施例中，在尾指標已被增加之後，當尾指標 904 等於頭指標 903 時，發生記憶器滿狀況。所以，一旦尾指標 904 已被增加，運轉計數重序單元 801 中的控制器邏輯比較尾指標 904 與頭指標 903，如果相同的話，則控制器邏輯決定碼字記憶器是滿的且應該輸出一碼字。在另一個實施例中，可以在記憶器滿之前來輸出碼字。例如，如果由頭定位址的部份佇列包含有效碼字，則它可以被輸出。如此需要佇列之開始被重覆地檢查以決定其中的碼字之狀態。須注意在檔案之編碼結束時，碼字記憶器 908 是空的。

使用運轉計數重序單元 801，在由頭指標 903 所指定的位址，首先藉由從碼字記憶器 908 讀取一值（例如資料）而輸出一碼字。使用控制器邏輯而控制並配合碼字之輸出。有效偵測模組 906 執行測試而決定是否此值為一碼字。換

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (53)

句話說，有效偵測模組 906 決定碼字是否有效。在一實施例中，有效偵測模組 906 藉由檢查儲存有各進入之有效位元，而決定進入之有效。如果此值為一碼字（亦即，碼字是有效的），則此值被輸出而作為一碼字。另一方面，如果此值並不是一碼字（亦即，碼字是無效的），則可以輸出任意碼字，其具有至少與目前運轉計數一樣長的 MPS 之運轉。"0" 碼字為一碼字其正確地表示運轉計數而可以被輸出。在輸出已做成之後，頭指標 903 被增加至碼字記憶體 908 中的下一個位置。另外，使用具有最短可允許的運轉長度之 "1N"，可允許解碼器只檢查是否一碼字已在發射 LPS 之前被強迫輸出。

在一實施例中，運轉計數重序單元 801 以二時鐘循環時間來操作。在第一時鐘循環，輸入被接收進入運轉計數重序單元 801。在第二時鐘循環，發生從碼字記憶體 908 輸出。

雖然每當頭指標 903 定位址一有效碼字時，可以輸出碼字，最好在某些實施中當緩衝器是滿的時只輸出一碼字。如此使得系統在許多碼字具有固定的延遲而非可變延遲。如果記憶體 908 能保持預定數目的碼字，在當運轉開始並被輸入時與當其被輸出時之間的時間，此延遲為此數目的碼字，由於並沒有做成輸出直到其是滿的。於是，在碼字中有一固定延遲。須注意重序延遲在其它方面仍是可變的，例如編碼或原始資料的量。藉著允許記憶體 908 在產生一輸出之前被填滿，輸出在每循環產生一碼字。

五、發明說明 (54)

須注意如果碼字記憶器位置被標示為無效，則未使用的位元可以被使用而儲存其是那一個運轉計數的辨識（亦即，必須填入位置的上下文筐或概率等級被儲存於其中）。此資訊可用於處理記憶器是滿的情形。特別是，可以使用資訊以指出此特別運轉長度之碼字沒有完成的位元產生器，及其現在必須完成的位元產生器。在此情形中，已做成決定以輸出一無效碼字，由於記憶器滿狀況而發生。於是，當系統復置運轉計數時，資訊指出當以位元產生器及運轉計數來看，系統正欲再度開始。

關於索引輸入，當使用概率等級之觸排時為了疊流原因，索引可以包括觸排辨識號(bank identifier)。亦即，對於一特別概率等級可以有許多運轉計數。例如，對於80百分比的碼可以使用兩個運轉計數，而在其它的碼則使用一個運轉計數。

由於碼字是可變長度，它們必須被儲存於碼字記憶器908中，其方式為允許它們的長度被決定。雖然可以明確地儲存尺寸，此並不會減小記憶體使用。對於R-碼，儲存零的值於記憶器中可以指出一位元"0"碼字，且"1N"碼字可以被儲存使得可以使用一優先編碼器以決定距離第一個"1"位元之長度。

如果碼字記憶器908為多阜(例如雙阜)，此設計可以被疊合以處理每一個時間循環一個碼字。因為碼字記憶器908中的任意位置可以從多阜被存取，碼字記憶器908中的位置可以被寫入，例如當一無效或碼字正在被儲存，而另

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (55)

一部份可以被讀取，例如當一碼字正在被輸出。須注意在此一情形中，多工器必須被修改以支援多資料與位址匯流排。

每當編碼器輸出一"0"碼字，且復置運轉計數因為碼字記憶器是滿的，則解碼器必須做相同的操作。如此需要解碼器對模型編碼器的碼字記憶器佇列。以下將討論其是如何完成。

須注意欲節省CMOS實施中的功率，對於"1N"碼字而言計數器可以被除能(disable)，當"0"碼字對於無效運轉被輸出。這是因為正在解碼的"1N"碼字是有效的，而只有一"0"碼字會是無效的。

根據上下文之另一個實施例：

圖10為運轉計數重序單元之另一個實施例的方塊圖，其重序依據上下文所接收的資料(與概率等級相對)。運轉計數重序單元1000使用R-碼而執行重序。參照圖10，重序單元1000包括指標記憶器1001、頭計數器1002、尾計數器1003、資料多工器(MUL)1004、位址MUX1005、計算長度方塊1006、有效偵測方塊1007、及碼字記憶器1008。碼字記憶器1008於重序期間儲存碼字。指標記憶器1001儲存上下文筐的碼字記憶器位置之位址，此上下文筐是在運轉的中間。頭計數器1002與尾計數器1003允許碼字記憶器1008被定位址而成爲一佇列或圓形緩衝器，除了正在被指標記憶器1001以隨機存取方式而定位址之外。對於R-碼，儲存一

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (56)

零值於記憶器中可以指出一位元"0"碼字，且"1N"碼字可以被儲存使得一優先編碼器可以被使用以決定距離第一個"1"位元之長度。計算長度模組1006以類似優先編碼器之方式來操作。如果使用其它的可變長度碼，則有更多有用的記憶器可將"1"位元加至碼字之開始，加入 $\log_2$ 位元以明確地儲存長度。運轉計數重序單元1000亦包括後級的控制器邏輯以配合並控制零件1001至1008的操作。

運轉計數重序單元1000的操作是非常類似根據概率估計之運轉計數重序單元。如果開始一新的運轉，則包括上下文筐之無效進入在由尾指標1003所指示的位址，被寫入碼字記憶器1008。然後尾指標1003位址在目前運轉計數的上下文筐之位址被儲存於指標記憶器1001中。然後尾指標1003被增加。當完成一運轉時，對應運轉計數之指標記憶器1001中的指標從指標記憶器1001被讀取，且在由此指標所指出的位置碼字被平行地寫入碼字記憶器1008。如果開始或結束一運轉，且如果由頭指標1002之位址所指出的碼字記憶器1008中之位置並沒有包含無效資料，則由頭定位址的碼字被讀取並被輸出。然後頭指標1002被增加。對於當一運轉同時開始及結束時之情形，碼字在由尾指標1003所指出的位址被寫入碼字記憶器1008，然後尾指標1003被增加。

類似地，當運轉結束且指標記憶器1001中的對應位址與頭計數器1002中的位址相同時，碼字可以馬上地被輸出，且頭計數器1002中的值可以被增加而不需存取碼字記憶

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (57)

器 1008。

對於"以上下文"運轉計數系統，每一個上下文需要指標記憶器 1001 中的記憶位置，所以 BG 與 PEM 狀態之寬度可以被延伸以實施此記憶器。指標記憶器 1001 之寬度等於碼字記憶器位址所需之尺寸。

碼字記憶器 1008 中的位置之數目可以被特別實施中的設計者所選擇。此記憶器之有限尺寸會降低壓縮效率，所以有價格/壓縮之考量(各一半一半)。碼字記憶器之寬度等於最大的碼字加上有效/無效指示的一個位元。

在以下的表 13 中指出使用 R2(2)碼的一個例子，其將用於說明重序。表 14 指出欲被重序的資料(0=MPS，較可能的符號；1=LPS，較不可能的符號)，以上下文來標記。只有兩種上下文。未編碼的位元數指出未編碼位元時鐘循環中的時間。指出運轉的開始與結束，且碼字表示在運轉的結束。

表 13 - R 2 (2) 碼

原始	碼字
0000	0
0001	100
001	110
01	101
1	111

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (58)

表 1 4 - 欲被編碼之範例資料

未編碼 位元數	資料	上下文	運轉之開 始 / 結束	碼字
1	0	0	S	
2	0	1	S	
3	0	0		
4	1	1	E	101
5	0	0		
6	0	1	S	
7	0	0	E	0
8	1	1	E	101
9	0	0	S	
10	0	1	S	
11	0	0		
12	0	1		
13	0	0		
14	0	1		
15	1	0	E	100
16	0	1	E	0

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (59)

範例資料之重序操作係表示於表15中。使用具有四個位置0-3之碼字記憶器，其在此例子中大到足以不會產生超限(overflow)。各列指出系統在操作之後的狀態，其為碼字之輸出或某上下文的運轉之開始或結束。使用一"x"來表示記憶器位置，其為"不管(don't care)"。對於某些未編碼位元，運轉開始或結束，使得運轉計數重序閒置的單元。對於結束運轉之編碼位元，可以輸出一個或以上的碼字，其會引起系統狀態的某些改變。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (60)

表 1 5 - 重序操作之範例

未編碼 位元數	輸入	poi指標		指標記憶器		碼字記憶器				輸出
		頭	尾	0	1	0	1	2	3	
1	開始0	0	1	0	x	無效	x	x	x	
2	開始1	0	2	0	1	無效	無效	x	x	
3	(重序單元閒置)									
4	結束1,101	0	2	0	x	無效	101	x	x	
5	(重序單元閒置)									
6	開始1	0	3	0	2	無效	101	無效	x	
7	結束0,0	0	3	x	2	0	101	無效	x	
		1	3	x	2	x	101	無效	x	0
		2	3	x	2	x	x	無效	x	101
8	結束,101	2	3	x	x	x	x	101	x	
		3	3	x	x	x	x	x	x	101
9	開始0	3	0	3	x	x	x	x	無效	
10	開始1	3	1	3	0	無效	x	x	無效	
11	(重序單元閒置)									
12										
13										
14										
15	結束0,100	3	1	x	0	無效	x	x	100	
		0	1	x	0	無效	x	x	x	100
16	結束1,0	0	1	x	x	0	x	x	x	
		1	1	x	x	x	x	x	x	0

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (61)

參照表 15，尾指標與尾指標被啓始爲零，表示在碼字記憶器中不含有任何東西(例如佇列)。指出具有兩個儲存位置之指標記憶器，其中一個儲存位置是用於上下文。各位置在位元數目一之前具有"不管"值。指出具有四個碼字之深度的碼字記憶器，全部啓始爲"不管"值。

響應位元數 1 所接收的資料，頭指標指向碼字記憶器位置 0。由於解碼器將預測資料，下一個有用碼字記憶器位置 0 被指定給碼字，且一無效值被寫入記憶器位置 0。因爲上下文是零，指定給碼字之碼字記憶器位置的位址，被儲存在零上下文(指標記憶器位置 0)之指標記憶器位置中。於是，一"0"被儲存在指標記憶器位置 0 中。尾指標被增加至下一個碼字記憶器位置 1。

響應對應位元數 2 之資料，尾計數器保持指向第一記憶器位置(由於並沒有輸出引起增加計數)。由於對應第二上下文(上下文 1)的資料，下一個碼字記憶器位置被指定給碼字而作爲由尾指標所指出的碼字記憶器位置 1，且一無效值被寫入此位置。位址(碼字位置 1)被寫入對應上下文 1 之指標記憶器位置。亦即，第二碼字記憶器位置的位址被寫入指標記憶器位置 1。然後尾指標被增加計數。

響應對應位元數 3 的資料，重序單元是閒置的，由於運轉並沒有正在開始或結束。

響應對應位元數 4 的資料，對於上下文 1 指示運轉的結束。所以，碼字"101"被寫入指定給上下文 1 之碼字記憶器位置(碼字記憶器位置 1)，由上下文 1 之指標記憶器位置所

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (62)

表示。頭指標與尾指標保持相同，且上下文1之指標記憶器位置中的值將不會再度被使用，所以它是"不管"。

響應對應位元數5的資料，重序單元是閒置的，由於運轉並沒有開始或結束。

響應對應位元數6的資料，發生位元2之如上述的相同型式操作。

響應對應位元數7的資料，發生上下文0之碼字的運轉結束。在此情形中，碼字"0"被寫入碼字記憶器位置(碼字記憶器位置0)，由上下文0之指標記憶器位置所表示(指標記憶器位置0)。然後指標記憶器位置上的值將不會再度被使用，由於它是"不管"。而且由頭指標所指出的碼字記憶器位置包含有效資料。所以，有效資料被輸出，且頭指標被增加計數。增加頭指標會使它指向包含一有效碼字的另一個碼字記憶器位置。所以，此碼字被輸出且頭指標再度被增加計數。須注意在此例子中，當碼字為致能時，它們被輸出，與當碼字記憶器是完全滿的情形相對。

依據上述，連續地發生未編碼位元之處理。須注意碼字記憶器位置並沒有用於特殊的上下文，使得來自任意上下文的碼字，在資料檔的整個編碼中，可以平行地被儲存於碼字記憶器位置中。

位元包封單元：

位元包封是表示於圖4中，指出在位元包封之前與之後，由重序單元所處理的資料。回到參照圖4，指出第十

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (63)

六個可變長度碼字，數字1至16指出解碼器的使用順序。各碼字被指定給三個編碼流的其中之一。各編碼流中的資料被打破成固定長度字(稱為插入字)。須注意單一可變長度碼字可以打破成兩個插入字。在此例子中，插入字被排序以單一插入流，使得特殊插入字中第一可變長度碼字(或部份碼字)的順序，決定插入字的順序。可以執行其它型式的排序。插入多編碼流的優點為：可以使用單一編碼資料通道以傳送資料，且對於各流而言能以平行或疊合的方式來移位可變長度。

本發明之位元包封單元802從運轉計數重序單元801接收可變長度碼字，並將它們包封成插入字。位元包封單元802包含邏輯以執行可變長度碼字及合併佇列型式重序單元的處理，以正確的順序而輸出固定長度插入字。在一實施例中，以每一個時鐘循環一個碼字的速率，從運轉計數重序單元接收碼字。位元包封單元802之一個實施例的方塊圖表示於圖11中。在以下的實施例中，使用四個插入流，各插入字為16位元，且碼字之長度從1改變至13位元。在一實施例中，單元位元包封單元被疊合(pipelined)以處理全部的流。如果本發明之位元包封單元802使用雙卓記憶器(或暫存器檔)，它可以每一個時鐘循環輸出一個插入字。這會比保持其它的編碼器更快。

再度參照圖11，位元包封單元802包括包封邏輯1101、流計數器1102、記憶器1103、尾指標1104、及頭計數器1105。包封邏輯1101被耦合以接收碼字，且被耦合至流計

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (64)

數器 1102。流計數器 1102 亦被耦合至記憶體 1103。尾指標 1104 與頭計數器 1105 亦被耦合至記憶體 1103。

流計數器 1102 保持追蹤與目前輸入碼字有關的插入流。在一實施例中，流計數器 1102 重複地計數 0 至 $N-1$ 的流，這裡 N 為流之數目。一旦流計數器 1102 到達 $N-1$ ，它開始再度從 0 計數。在一實施例中，流計數器 1102 為二位元計數器，並從 0 計數至 3 (對於四個插入流)。在一實施例中，流計數器 1102 被啓始為零 (例如經由整體復置)。

包封邏輯 1101 合併目前的輸入碼字與先前的輸入碼字以形成插入碼字。各碼字之長度會改變。所以，包封邏輯 1101 包封這些可變長度碼字成固定長度字。由包封邏輯 1101 所產生的插入碼字依序被輸出至記憶體 1103，並被儲存在記憶體 1103 中，直到適當時間而將它們輸出。在一實施例中，記憶體 1103 為一靜態隨機存取記憶體 (SRAM) 或具有十六對四之 16 位元字的暫存器檔。

插入碼字被儲存於記憶體 1103 中。在本發明中，記憶體 1103 之尺寸大到足以處理兩種情形。一種情形為正常操作情形，其中一插入流具有最小長度碼字，且其它的插入流具有最大長度碼字。此第一種情形需要 $3 \times 13 = 39$ 個記憶體器位置。另一種情形為啓始情形，其中一流再度具有最小長度或短的碼字，而其它的流具有最大長度或長的碼字。對於第二種情形而言，雖然 $2 \times 3 \times 13 = 78$ 個記憶體器位置是足夠的，PEM 之操作允許 56 之較緊界限。

與流計數器 1102 及尾指標 1104 配合的記憶體 1103 執行

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (65)

重序。流計數器 1102 指出正在由記憶器 1103 所接收的目前碼字流。各插入流是與至少一尾指標有關。尾指標 1104 與頭計數器 1105 執行碼字之重序。當插入字 N-1 中的資料包含下一個碼字的開始時，解碼器所要求的插入字 N 之後的每一流具有兩個尾指標。其原因是一個尾指標決定記憶器 1103 中的位置以儲存來自一給定插入流的下一個插入字。另一個尾指標決定記憶器中的位置以儲存在下一個之後的插入字。如此允許當知道插入字 N-1 的解碼器要求時間，可以指定插入字 N 的位置。在一實施例中，指標為八個 6 位元暫存器（每一流有兩個尾指標）。

在一實施例中，在編碼的開始，尾指標 1104 被設定，使得第一組八個插入字（每流兩個）依順序被儲存於記憶器 1103 每流一個。在啓始之後，每當包封邏輯 1101 對於一特殊碼流開始一新的插入字，"下一個"尾指標被設成下一個尾指標之後的值，碼流的下一個尾指標之後被設定至下一個有用記憶器位置。於是，各流有兩個尾指標。在另一個實施例中，各流只使用一尾指標，並指出下一個插入字欲被儲存於記憶器 1203 的何處。

使用頭計數器 1105 以決定下一個插入字欲從位元包封單元 802 被輸出的記憶器位置。在所述的實施例中，頭計數器 1105 包含 6 位元計數器，其被增量以一次輸出整個插入字。

記憶器 1103，除了被使用於重序之外，亦可以被使用作為編碼器與通道之間 FIFO（先進先出）緩衝。最好使此記

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明(66)

憶器大於重序所需的記憶器，使得可以使用FIFO幾乎滿的信號以拖延編碼器，當通道無法跟上編碼器。每循環一位元之編碼器無法每循環產生一插入字。當編碼器與通道良好地匹配時，通道將無法每循環接收一插入字，且某些FIFO緩衝是需要的。例如，當壓縮為2:1或更大時，對於2:1有效頻帶寬度展開的良好匹配設計而言，一通道可以每32循環接收一16位元插入字。

本發明之包封邏輯：

包封邏輯之方塊圖是表示於圖12中。參照圖12，包封邏輯1101包含尺寸單元1201、一組累積器1202、移相器1203、MUX1204、一組暫存器1205、及OR閘邏輯1206。尺寸單元1201被耦合以接收碼字，並被耦合至累積器1202。累積器及碼字被耦合至移相器1203。移相器1203被耦合至MUX1204及OR閘邏輯1206。MUX1204亦被耦合至OR閘邏輯1206。

在一實施例中，碼字被輸入在一13位元的匯流排上，未使用的位元被歸零。這些被歸零的未使用位元是與"1N"碼字的"1"相鄰，使得尺寸單元1201中的優先編碼器可以被使用，以決定碼字"1N"的長度，並產生"0"碼字之尺寸。

累積器1202包含多累積器，每一插入流各一個。各插入流之累積器保持已在目前插入字中的位元之數目。在一實施例中，各累積器包含4位元加法器(具有進位)及各流

五、發明說明 (67)

所使用的4位元暫存器。在一實施例中，加法器之輸出為累積器的輸出。在另一實施例中，暫存器之輸出為累積器之輸出。使用從尺寸單元所接收的碼字之尺寸，累積器決定位元之數目，以移位而連結目前的碼字進入暫存器，此暫存器包含此流之目前插入字。

根據累積器之目前值，移相器1203對齊目前碼字，使得其適當地跟隨此插入字中的任意先前碼字。於是，編碼器中的資料被移位成解碼器順序。移相器1203之輸出為28位元，其處理13位元的碼字必須附加至目前插入字中的15位元，使得來自目前碼字的位元結束在正被輸出的28位元中的更高12位元。須注意移相器1203是以無反饋的方式操作，於是可以疊合。在一實施例中，移相器1203包含一筒形移相器。

暫存器1205儲存位元於目前插入字中。在一實施例中，各插入流之16位元暫存器保持目前插入字中的先前位元。

開始時，由移相器1203接收一碼字流，而尺寸單元1201指出碼字之尺寸給對應流的累積器。累積器具有零之啓始值，此零是經由整體復置而設定。由於累積器值為零，碼字並沒有被移位，然後使用具有對應流之暫存器的內容OR邏輯1206。然而，在某些實施例中，1N碼字必須被移位以適當地對齊，即使是在一插入字的開始。此暫存器已被啓始成零，所以OR操作之結果是將碼字放入OR邏輯1206的輸出之最右邊的位元位置，並經由MUX1204被反饋

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (68)

至用於儲存的暫存器，直到來自流的下一個碼字。於是，啓始移相器1203操作爲一經過路徑。須注意第一碼字中的位元之數目現在被儲存於累積器中。當接收此流之下一個碼字，累積器中之值被送至移相器1203，且碼字被移位至此數目的位元之左邊，用於與插入字中之先前輸入位元結合。許多零被放置在移位後的字中之其它位元位置。來自對此流的暫存器之位元，使用OR邏輯1206而與來自移相器1203之位元結合。如果累積器沒有產生進位指示(例如信號)，則需要更多的位元以完成目前的插入字，且由OR操作所產生的資料經由MUX1204被儲存送回至暫存器。在一實施例中，MUX1204包含2:1多工器。當累積器產生一進行時，來自OR邏輯1206之16位元的資料爲一完成的插入字並被輸出。MUX1204使暫存器在第一個16之後載入額外的位元(例如從移相器1203所輸出的28位元之上12位元)，並將其餘部份填以零。

插入字之輸出與MUX1204的控制包含來自累積器之進位信號。在一實施例中，多工器1204包含十六個2:1多工器，其中四個具有輸入爲零。

重序選擇：

本發明提供用於執行重序資料的多重選擇。例如，在具有多碼流的系統中，碼流必須被重序成圖4所示的插入字。本發明提供許多方式來完成重序成插入字。

將資料重序成插入字的一個方法是使用如圖25所示的

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (69)

snooper解碼器。參照圖25，多運轉計數重序單元2501A-n被耦合以接收與碼字流對齊的碼字資訊。各產生一碼字輸出及一尺寸輸出。各別的位元包封邏輯(1101)單元，例如位元封包單元2502A-n，被耦合以接收來自其中一個運轉計數重序單元2501A-n之碼字及尺寸輸出。位元包封邏輯單元2502a-n輸出插入字，其被耦合至MUX2503與snooper解碼器2504。解碼器2504提供一由MUX2503所接收之選定控制信號，並指示MUX2503那一個插入字欲被輸出進入碼流。

各編碼後的資料流具有一運轉計數重序單元，包含圖8中的運轉計數重序單元801。各位元封包單元將可變長度碼字組合成固定尺寸的插入字，每一個字為8、16或32位元。各位元封包單元如前所述含有暫存器及移位電路。解碼器2504包含全操作的解碼器(包括BG、PEM與CM)。其可存取來自全部位元封包單元之插入字(經由圖25所示之各別的匯流排或一共用的匯流排)。每當解碼器2504從其中一個位元封包單元被選定一插入字時，此字被傳送於碼流中。由於在接收端的解碼器會要求與相同的snooper解碼器相同的資料，插入字是以適當的順序被傳送。

具有snooper解碼器之編碼器在半雙工系統中是非常吸引的，由於snooper解碼器可以使用作為正常的解碼器。snooper解碼器方式的優點在於其可應用於任何定性解碼器。以下敘述另一種解決方式而不依附snooper解碼器，使用較簡單模型的解碼器以降低硬體價格。對於解碼相

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (70)

同的時鐘循環中之多碼字的解碼器，可將解碼器模組成比使用 snooper 解碼器之解碼器本身較少的硬體。以下將敘述對於每循環一碼字的解碼存在較簡單的模型。

用於重序疊合解碼器系統之資料的另一種技術，其在每循環解碼一碼字，是根據只有對於需要模組解碼器要求的編碼資料之資訊，要知道碼字的順序（考慮全部碼字，並各編碼資料流之碼字獨立地）。如果時間標記（time stamp）是與各碼字有關，當其進入運轉計數重序單元，每一個位元封包後的插入字具有最舊的時間標記，其與欲輸出的下一個插入碼字有關。

在圖 26 中以方塊圖指出用於範例的編碼器重序單元。參照圖 26，編碼系統是與圖 25 所述相同，除了時間標記資訊是由各運轉計數重序單元 2501A-n 所接收。此時間標記資訊亦向前送至位元封包單元 2502A-n。位元封包單元 2502A-n 提供插入字至 MUX2503，及其有關的時間標記至邏輯 2601。邏輯 2601 提供一控制信號至 MUX2503 以選定插入字 Z 輸出至碼流。

在此實施例中，snooper 解碼器是以簡單的比較來取代，其決定那一個位元封包單元 2502a-n 具有最舊時間標記之碼字。此一系統出現至 MUX2503 作為具有時間標記之多佇列。邏輯 2601 簡單地選定於不同的佇列之間。各運轉計數重序單元 2503A-n 之邏輯當運轉開始時只輕微地改變（由運轉計數重序單元 801）以寫入一時間標記。設置各運轉計數重序單元 2501A-n 以儲存碼字記憶器中的時間標記。

五、發明說明 (71)

儲存具有足夠位元以將編碼資料流中之各碼字標號的時間標記是充分的，但是在某些實施例中，則使用較少的位元。

以下簡短地敘述使用具有時間標記之多佇列的步驟。此敘述對熟於此技藝人仕而言是可辨識的。有許多編碼器操作。對於一運轉是由相同的碼字來開始及結束的情形而言是無法簡化。對於各編碼符號可以檢查操作（雖然在實際上並不需要全部的檢查）。假設插入字之尺寸為32位元。

```

if (no current codeword for context) {
    place time in Queue (used to determine next Queue)
    place context pointer in Queue
    place invalid data in Queue
    point context to Queue entry
    increment Queue tail
}
if (already a codeword and MPS) {
    increment context runcount
}

```

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (72)

```

if (MAXRUN or LPS) {
    place correct data in Queue
    (context pointer unneeded)
    zero pointer & runcount in context memory
    update probability estimate in context memory
}

if (valid data at head of next queue) {
    place 32 bits of data on output
    clear Queue entry
    Increment Queue head
}

while (any queue is almost full) {
    find the next Queue which must place data on the output
    while (less than 32 bits of valid data) {
        use context pointer to find context
        zero pointer & run count in context memory
        place MAXRUN code word in Queue data
    }
}

```

解碼器操作是類似的，雖然碼字不需要被儲存於佇列中。仍然需要儲存佇列中的碼字之時間標記。

上面所討論的時間標記之功能是用於儲存碼字之順序資訊。表示相同觀念之等效方式是經由使用全部碼字之單一佇列，亦即一合併佇列。在合併佇列的系統中，如圖 27 所示，對於全部插入流使用單一的運轉計數重序單元 2701。運轉計數重序單元 2701 產生碼字、尺寸及流輸出至位元封包單元 2502A-n 輸出插入字至 MUX2503，及位置資訊至邏輯 2702，其發信至 MUX 以輸出插入字作為部份的碼流。

對任意流而言，運轉計數重序記憶體儲存各碼字之插入流 ID。各插入流具有其自己的頭指標。當位元封包單元需要更多的資料時，使用對應的頭指標以取得需要形成一新插入字一樣多的碼字。此可以包含檢視許多碼字記憶體位置以決定何者為部份的適當流。另一種方式，此可以包

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (73)

含檢視額外欄位的碼字記憶器以實施一鏈接的列表。

本發明中插入流的另一種方法是使用具有固定流指定的合併佇列。此方法使用單一尾指標合併佇列情形中，所以不需要時間標記。而且，在先前的情形中使用多頭指標，所以輸出來自特殊流的資料不會過頭。欲完成此，是依據以下規則來執行將碼字指定給插入流，對於 N 流：碼字 M 被指定給流 M 模組 N 。須注意依據此方法插入流可以具有來自任意上下文筐或概率等級之碼字。如果流之數目為 2 的次方，則可以藉著拋棄某些較明顯的位元而計算 M 模組 N 。例如，假設碼字重序記憶器是以 12 位元來定位址，且使用四個插入流。尾指標為 12 位元長，且兩個較不明顯的位元辨識下一個碼字之編碼流。四個具有 10 位元之頭指標，各被隱含地指定給兩個最不明顯位元之四種可能組合。尾指標及頭指標均被增量而成正常的二進位計數器。

在解碼器中，移相器具有暫存器以儲存插入字。此移相器可表現適當對齊的編碼資料至位元產生器。當位元產生器使用某些編碼資料時，其通知移相器。此移相器表現來自下一個插入流的適當對齊資料。如果編碼資料流的數目為 N ，則移相器具有 $N-1$ 個時鐘循環以移位出使用後的資料，要求將再度使用特殊插入流之前的另一個插入碼字。

本發明之解碼器：

本發明包括一解碼器，其支援包含有限重序記憶器之即時編碼器。在一實施例中，解碼器亦包括減少的記憶器

五、發明說明 (74)

需求及複雜性，藉著保持各概率等級而非各上下文筐之運轉計數。

本發明之解碼器系統的一個實施例：

圖 14A 指出本發明之解碼器硬體系統的一個實施例之方塊圖。參照圖 14A，解碼器系統 1400 包括先前先出 (FIFO) 結構 1401、解碼器 1402、記憶器 1403、上下文模型 1404。解碼器 602 包括多個解碼器。耦合編碼資料 1410 以被 FIFO 結構所接收。FIFO 結構 1401 被耦合以供給編碼後的資料給解碼器 1402。解碼器 1402 被耦合至記憶器 1403 及上下文模型 1402。上下文模型 1404 亦被耦合至記憶器 1403。上下文模型 6.04 的一個輸出包含解碼資料 1411。

在系統 1400 中，輸入於 FIFO 結構 1401 的編碼資料 1410 被重序並插入。FIFO 結構 1401 包含適當順序的資料。此流被傳送至解碼器 1402。解碼器 1402 需要來自串聯的這些流之資料及定性順序。雖然解碼器要求編碼資料之順序並非不重要的，其並不是隨機的。藉由在編碼器而非解碼器將碼字重序成此順序，編碼資料可插入於單一流中。在另一實施例中，編碼資料 1410 可以包含單一流的非插入資料，各上下文筐、上下文等級或概率等級之資料被附加至資料流上。在此情形中，FIFO 1410 被一儲存區域 1410 所取代以在將資料送至解碼器 1402 之前，接收全部的編碼資料，所以可以將資料適當地分段。

當由 FIFO 1401 接收編碼資料時，上下文模型 1404 決定

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (75)

目前的上下文筐。在一實施例中，上下文模型 1404 根據先前的圖素且 / 或位元來決定目前的上下文筐。雖然未示，線緩衝可以包含於上下文模型 1404 中。此線緩衝提供必需的資料或模板，藉此上下文模型 1404 可決定目前的上下文筐。例如，上下文是根據目前圖素附近的圖素值之情形，可以使用線緩衝以儲存在被使用以提供特定上下文附近的這些圖素之圖素值。

響應上下文筐，解碼器系統 1400 取得來自目前的上下文筐之記憶體 1403 的解碼器狀態。在一實施例中，解碼器狀態包括概率估計模型 (PEM) 狀態及位元產生器狀態。PEM 狀態決定使用那一個碼來解碼新碼字。位元產生器狀態保持在目前筐中的位元之記錄。響應由上下文模型 1404 所提供的位址，此狀態從記憶體 1403 被提供至解碼器 1402。此位址存取記憶體 1403 中之位置，此記憶體儲存對應上下文筐之資訊。

一旦從記憶體 1403 取得目前上下文筐之解碼器狀態，系統 1400 決定下一個未壓縮位元並處理解碼器狀態。然後解碼器 1402 解碼新的碼字，如果需要的話，且 / 或更新運轉計數。PEM 狀態被更新，如果需要的話，連同位元產生器狀態被更新。然後解碼器 1402 將新的編碼器狀態寫入記憶體 1403。

圖 14B 指出本發明之解碼器的一個實施例。參照圖 14B，解碼器包括移位邏輯 1431、位元產生器邏輯 1432、"新 k" 邏輯 1433、PEM 更新邏輯 1434、新碼字邏輯 1435、PEM 狀

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (76)

態對快取邏輯 1436、碼對罩邏輯 1437、碼對 MaxPL；罩；
R 3Split展開邏輯 1438、多工器 1440、及運轉計數更新邏
輯 1441。移位邏輯 1431被耦合以接收編碼資料輸入 1443以
及狀態輸入 1442(來自記憶器)。移位邏輯 1431之輸出亦被
耦合而作為位元產生器邏輯 1432、"新 k"產生邏輯 1433、
PEM 更新邏輯 1434之輸入。位元產生器邏輯 1432亦被耦合
以接收狀態輸入 1442，並產生解碼資料輸出至上下文模型
。新-k邏輯 1433產生一輸出，此輸出被耦合至碼對罩邏輯
1437之輸入。PEM更新邏輯 1434亦被耦合至狀態輸入 1442
，並產生狀態輸出(至記憶器)。狀態輸入 1442亦被耦合至
新碼字邏輯 1435與 PEM 狀態對碼邏輯 1436之輸入。耦合 PEM
狀態對碼邏輯 1436之輸出以被展開邏輯 1438所接收。展開
邏輯 1438之輸出被耦合至解碼邏輯 1438及運轉計數更新邏
輯 1441。解碼邏輯的另一個輸入被耦合至碼對罩 1437之輸
出。解碼邏輯 1439之輸出被耦合至 MUX1440的一個輸入。
MUX1440之另一個輸入被耦合至狀態輸入 1442。MUX1440之
選定輸入被耦合至新碼字邏輯 1435之輸出。MUX1440與展
開邏輯之輸出被耦合至具有碼對罩邏輯 1437之輸出的運轉
計數更新邏輯 1441之兩個輸入。運轉計數更新邏輯 1441之
輸出包括於記憶器之狀態輸出中。

移位邏輯 1431移位來自碼資料流的資料。根據編碼資
料輸入及狀態輸入，位元產生器邏輯 1432產生解碼資料至
上下文模型。新-k邏輯 1433亦用移位後的資料及狀態輸入
以產生 k 的新值。在一實施例中，新-k邏輯 1433使用 PEM

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (77)

狀態及編碼資料的第一位元以產生k的新值。根據新的k值，碼對罩邏輯1437產生下一個碼字之RLZ罩。下一個碼字之RLZ罩被送至解碼邏輯709及運轉計數更新邏輯1441。

PEM更新邏輯1434更新PEM狀態。在一實施例中，PEM狀態使用現在狀態而被更新。更新後的狀態被送至記憶器。新碼字邏輯1435決定是否需要一新碼字。PEM狀態對碼邏輯1436使用狀態輸入1442而決定用於解碼之碼。此碼被輸入至展開邏輯1438以產生最大運轉長度、目前罩、及R3分裂值(split value)。解碼邏輯1439將碼字解碼以產生運轉計數輸出。MUX1440選定來自解碼邏輯1439或狀態輸入1442之輸出至運轉計數更新邏輯1441。運轉計數更新邏輯1441更新運轉計數。

本發明之解碼系統1400包括解碼器1430是以疊合方式來操作。在一實施例中，本發明之解碼系統600決定上下文筐、估計概率、解碼碼字，並以疊合方式來產生來自運轉計數之全部位元。解碼系統之疊合結構的一個實施例是敘述於圖15A。參照圖15A，是以六個級標號為1-6來表示本發明之疊合解碼處理的一個實施例。

在第一級中，決定目前上下文筐(1501)。在第二級中，於決定上下文筐之後，從記憶器取得上下文筐之目前解碼器狀態，發生記憶器讀取。如上所述，解碼器狀態包括PEM狀態及位元產生器狀態。

在本發明之疊合解碼處理的第三級中，產生一解壓縮位元(1503)。如此允許對於上下文模型而言有一位元是有

五、發明說明(78)

用的。於第三級期間發生兩個其它的操作。PEM狀態被轉換成一種碼型式(1504)，且在第三級中亦發生決定是否一新碼字必須被解碼(1505)。

於第四級期間，解碼系統處理一碼字且/或更新運轉計數(1506)。數個副操作(sub-operations)包含於處理碼字及更新運轉計數中。例如，一碼字被解碼以決定下一個運轉計數，或者對於目前碼字而言更新運轉計數(1506)。如果需要的話，當解碼新的碼字時，可以從輸入FIFO取得更多的編碼資料。發生於第四級中的另一個副操作為PEM狀態(1507)之更新。最後，在解碼疊合的第四級中，使用新的PEM狀態以決定運轉長度零碼字(敘述於後)是用於下一個碼，如果目前碼字之運轉計數為零(1508)。

於本發明之解碼疊合的第一級期間，具有更新的PEM狀態之解碼器狀態被寫入記憶器(1509)，且移位開始下一個碼字(1510)。在第六級中，完成移位至下一個碼字(1510)。

本發明之疊合解碼實際上是以是否開始解碼處理的判定來開始。此判定是根據是否有足夠的資料可表現至本發明之解碼器。如果沒有足夠的資料來自FIFO，則停止解碼系統。在另一情形中，當輸出解碼資料至一周邊裝置，其不能接收來自解碼器之全部資料輸出當其正被產生時，則會停止解碼系統。例如，當解碼器正在提供輸出至一視頻顯示介面及其有關的視頻電路時，視頻可以是非常緩慢，使得解碼器需要被停止以允許視頻跟上。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(79)

一旦做成決定開始解碼處理，則由上下文模型來決定目前上下文筐。在本發明中，目前上下文筐是藉著檢查先前資料而加以確定。此先前資料可以被儲存於線緩衝器中，且可以包括來自目前線且/或先前線之資料。例如，在上下文模板中，對於一給定位元而言，來自線緩衝器之位元可以使用相對於先前資料之模板來設計，使得目前資料之上下文筐依據正被檢查的先前資料是否符合模板而被選定。這些線緩衝器可以包括位元移位暫存器。對於一 n 位元影像的各位元平面可以使用一模板。

在一實施例中，藉著於下一個疊合級期間，將一位址輸出至記憶器而選定上下文筐。此位址可以包括一預定數目的位元例如三個位元以辨識位元平面。藉由使用三個位元，可以辨識圖素資料中的位元位置。用於決定上下文之模板亦可以由一部份的位址來表示。用於辨識位元平面之位元及辨識模板之位元可以被組合，以產生在記憶器中之特定位置之位址，其包含由這些位元所定義的上下文筐之狀態資訊。例如，藉由使用三個位元以決定特殊圖素中的位元位置及模板中的各先前圖素中之相同位置中的先前位元，可以產生13位元的上下文位址。

使用由上下文模型所產生的位址，記憶器(例如RAM)被存取以得到狀態資訊。此狀態包括PEM狀態。PEM狀態包括目前概率估計。因為一個以上的狀態使用相同的碼，PEM狀態並沒有包括概率等級或碼指定，而是表中之索引，此表例如圖5中之所示之表。而且當使用例如圖5所示

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (80)

之表時，PEM狀態亦提供最大可能符號(MPS)作為一種方式用於辨識是否目前PEM狀態是定位在表之正或負側上。位元產生器狀態可以包括計數值及是否出現LPS的指示。在一實施例中，目前運轉之MPS值亦被包括，以解碼下一個碼字。在本發明中，位元產生器狀態被儲存於記憶體中，以降低運轉計數所需之空間。如果系統中用於各上下文之計數器的空間之價格很低，位元產生器狀態並不需要被儲存於記憶體中。

一旦完成第四級，新的位元產生器狀態及PEM狀態被寫入記憶體。而且在第五級中，編碼資料流被移位至下一個碼字。在第六級中完成此移位操作。

圖14C為本發明之FIFO結構1401之一個實施例的方塊圖，指出兩個解碼器之插入字緩衝。須注意使用本發明之技術可以支持任意數目的解碼器。如圖所示，輸入資料及FIFO是夠寬足以容納兩個插入字。FIFO1401包含FIFO1460、暫存器1461-62、MUX1463-1464及控制區塊1465。兩個輸入碼字被耦合作為輸入插入字。FIFO之輸出被耦合至暫存器1461-1462之輸入。MUX1463之輸入被耦合暫存器1461與1462之輸出。控制區塊1465被耦合以提供控制信號至FIFO1460、暫存器1461與1462、及MUX1463與1464。插入字為提供至兩個解碼器之輸出資料(輸出資料1與2)。各解碼器使用一要求信號以指出，目前的字已被使用，且將需要下一個新的字。來自解碼器之要求信號被耦合至控制區塊1465的輸入。控制區塊1465亦輸出一FIFO

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (81)

要求信號以從記憶器要求更多的資料。

啓始地，FIFO及暫存器1461與1462被填入資料，且控制單元1465中的有效正反器被設定。每當發生一要求時，控制區塊1465依據表16所示之邏輯來提供記憶器。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (82)

表 1 6

二者有效	要求 1	要求 2	多工器 1	多工器 2	下一個二者有效	FIFO 及暫存器致能
0	0	0	X*	X	0	0
0	0	1	X	REG 1462	1	1
0	1	0	REG 1462	X	1	1
0	1	1	REG 1462	FIFO	0	1
1	0	0	X	X	1	0
1	0	1	X	REG 1461	0	0
1	1	0	REG 1461	X	0	0
1	1	1	REG 1461	REG 1462	1	1

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(83)

"X表示'不管'"

圖 15B 指出本發明之解碼器的不同觀念。參照圖 15B，可變長度(編碼)資料被輸入至解碼器。此解碼器輸出固定長度(解碼)資料。此輸出亦被反饋作為延遲反饋，其被接收作為解碼器之輸入。在本發明之解碼器中，於解碼中所使用的可變長度移位是根據解碼資料，其在某些延遲之後是有用的。此反饋延遲並不會降低延遲容許解碼器之產量。

輸入可變長度資料被分成固定長度插入字，例如圖 4 所述。解碼器使用固定長度字，如以下圖 16A 所述。解碼器及延遲模組成一疊合解碼器，如圖 15 與 32 所述，或多個平行解碼器如圖 2A-2D 所述。於是，本發明提供一延遲容許解碼器。本發明之延遲容許解碼器允許平行地處理可變長度資料。

習知技術的解碼器(例如 Huffman 解碼器)並非延遲容許。要求由解碼全部的先前碼字所決定的資訊，以執行欲解碼下一個碼字所需的可變長度移位。另一方面，本發明提供延遲容許解碼器。

解碼系統中的移位：

本發明之解碼器提供移位邏輯以將插入字移位至用於解碼之適當的位元產生器。本發明之移相器並不需要任何特殊型式的"以上下文"或"以概率"平行。解碼器將碼字 M 指定給流 M 模組 N(在 C 語言中之 $M\%N$)，假設其中 N 為流之數

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(84)

目。在本發明中，來自目前流之編碼資料出現，直到一碼字被要求。然後只有資料被轉換至下一個流。

圖16指出本發明之解碼器的移相器之一個實施例。移相器1600被指定給四個資料流。此允許各移位操作有四個時鐘循環。插入字為16位元，且最長的碼字為13位元。參照圖16，移相器1600包含四個暫存器1601-1604，其被耦合以接收來自插入編碼資料的輸入。各暫存器1601-1604之輸出被耦合作為MUX1605之輸入。MUX1605之輸出被耦合至桶形移相器1606之輸入。桶形移相器1606之輸出被耦合作為暫存器1607、MUX暫存器1608-1610、及尺寸單元1611之輸入。尺寸單元1611之輸出被耦合至一累積器1612。累積器1612之輸出被反饋並耦合至桶形移相器1606。暫存器1607之輸出被耦合作為MUX暫存器1608之輸入。MUX暫存器1608之輸出被耦合作為MUX暫存器1609之輸入。MUX暫存器1609之輸出被耦合作為MUX暫存器1610之輸入。MUX暫存器1610之輸出為對齊的編碼資料。在一實施例中，暫存器1601-1604為16位元暫存器，桶形移相器1606為32位元對13位元桶形移相器，且累積器1612為一4位元累積器。

暫存器1601-1604接受來自FIFO之16位元字，並將它們輸入至桶形移相器1606。至少未解碼的資料之32位元在全部次數被提供至桶形移相器1606。四個暫存器1601-1604以編碼資料的兩個16位元字啓始化而開始。此允許總是有至少一個新碼字可用於各流。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (85)

對於R-碼，碼字尺寸單元1611決定"0"或"1N"碼字出現，如果其為"1N"碼字，部份的目前碼字在"1"之後有多少個位元。尺寸單元提供相同的功能，此尺寸單元敘述於圖12中。對於其它的碼，決定碼字之尺寸是習知的。

移相器1600包含一FIFO，含有四個暫存器，其中三個具有多工輸入。暫存器1607-1610之各暫存器容納至少一個碼字，所以暫存器與多工器之寬度為13位元，以容納最長的可能碼字。各暫存器亦具有一控制正反器與其相連（為了避免妨礙本發明而未示），其表示是否一特殊暫存器包含一碼字或是否等待桶形移相器1606以提供一碼字。

FIFO將不會是空的。每個時鐘循環只能使用一個碼字，且每個時鐘循環只能移位一碼字。欲執行移位之延遲被補償，由於系統只開始四個碼字於前面。當各碼字被移位成對齊的編碼資料輸出時，暫存器1607-1610中之其它的碼字向下移位。當留在FIFO中的碼字被儲存於暫存器1620中時，桶形移相器1606使碼字經由MUX1605而從暫存器1601-1604被讀取，以填入暫存器1607-1609。須注意FIFO可以被指定以下一個碼字來重填暫存器1607，只要其碼字被移位進入暫存器1608。

桶形移相器1606、碼字尺寸計算器1611及累積器1612選定處理可變長度移位。累積器1602具有四個暫存器，各編碼資料流具有一個，其包含各資料流之目前碼字的對齊。累積器1612為四位元累積器，用於控制桶形移相器1606。累積器1612以來自碼字尺寸單元1611之輸入值來增加其

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (86)

值。當累積器 1612 溢流時 (例如每次移位計數為 16 或更大) , 暫存器 1601-1604 被定時而移位。各其它的 16 位元移位引起一新的 32 位元字從 FIFO 被要求。累積器 1611 之輸入為碼字之尺寸, 其是由目前碼字的前一個或兩個位元與目前碼來決定。須注意在某些實施例中, 暫存器 1601-1604 在解碼可以開始之前, 必須以編碼資料來啓始。

當系統要求一碼字時, FIFO 中之暫存器被定時, 所以碼字被移至輸出。當桶形移相器 1606 準備傳送一新碼字時, 其被多工化成 FIFO 中之第一空暫存器。

在此實施例中, 來自位元產生器之下一個碼字信號, 在做成決定以轉換流之前被接收。

如果來自位元產生器之下一個碼字信號無法保證在轉換流的決定之前被接收, 則可以使用一向前看系統例如圖 16B 中所示。參照圖 16B, 使用向前看之移相器 1620 是以方塊圖形式來表示。移相器 1620 包括一移相器 1600, 其產生目前編碼資料及下一個編碼資料之輸出。目前編碼資料被耦合至碼字處理邏輯單元 1621 之輸入及碼字處理單元 1624 之輸入。下一個編碼資料被耦合至碼字處理邏輯單元 1622 之輸入。處理邏輯單元 1621 與 1622 之輸出均被耦合至 MUX 1623 之輸入。MUX 1623 之輸出被耦合至碼字處理邏輯 1624 之另一個輸入。

使用碼字之邏輯被分成兩個部份: 碼字預先處理邏輯及碼字處理邏輯。兩個相同的疊合預先處理單元 1621-1622 在插入流可以被移位之前操作。預先處理單元 1612-

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(87)

1622之其中一個可產生適當資訊如果流被轉換，而另一個則產生資訊如果流沒有被轉換。當流被轉換時，適當的碼字預先處理之輸出被MUX1623多工化而送至碼字處理邏輯1624，其以適當的碼字來完成操作。

晶片外記憶器及上下文模型：

在一實施例中，最好使用多晶片用於外部記憶器或外部上下文模型。在這些實施例中，最好降低產生一位元與使位元可用於上下文模型之間的延遲，其中使用多個積體電路。

圖17指出系統的一個實施例之方塊圖，具有外部上下文模型晶片1701與含有各上下文之記憶器的編碼器晶片1702。須注意只有指出與編碼器晶片中之上下文模型有關的單元；對熟於此技藝者很明顯編碼器晶片1702包含位元產生、概率估計等等。參照圖17，編碼器晶片1702包含零順序上下文模型1703、上下文模型1704與1705、選定邏輯1706、記憶器控制1707及記憶器1708。零順序上下文模型1703與上下文模型1704-1705產生輸出，其被耦合至選定邏輯1706的輸入。選定邏輯1706的另一個輸入被耦合至外部上下文模型晶片1701的輸出。選定邏輯1706之輸出被耦合至記憶器1708的輸入。記憶器1708之輸入亦耦合至記憶器控制1707的輸出。

選定邏輯1706允許使用外部上下文模型或內部上下文模型(例如零順序上下文模型1703、上下文模型1704、上

五、發明說明 (88)

下文模型 1705)。選定邏輯 1706 允許使用上下文模型 1703 之內部零順序部份，即使當使用外部上下文模型。零順序上下文模型 1703 提供一位元或更多的位元，而外部上下文模型晶片 1701 提供其餘的部份。例如，先前位元可以馬上被反饋，並從零順序上下文模型 1703 取還，而先前位元則走至外部上下文模型 1701。以此方式，時間臨界資訊保持在晶片上。此消除最近產生的位元之晶片外通信延遲。

圖 18 為具有外部上下文模型 1801、外部記憶器 1803 與編碼器晶片 1802 的系統之方塊圖。參照圖 18，某些記憶器位址線被外部上下文模型驅動，而其它的部份則由解碼器晶片 1802 中之“零順序”上下文模型所驅動。亦即，來自立即通過的解碼循環之上下文是由零順序上下文模型所驅動。此允許解碼器晶片提供來自立即通過的上下文資訊，其具有最小通信延遲。上下文模型晶片 1802 進行其餘的上下文資訊，只使用過去的解碼位元，所以允許通信延遲。在許多情形中，來自立即通過的上下文資訊為零順序 Markov 狀態，且來自過去的上下文資訊為更高順序的 Markov 狀態。圖 18 所示的實施例消除實施外部上下文模型晶片 1802 中之零順序模型所固有的通信延遲。然而，由於解碼器晶片 1802 與記憶器 1803，會仍有一上下文筐決定送至位元產生延遲。

須注意可以使用其它的記憶器架構。例如，可以使用具有一晶片中之記憶器與另一晶片中之編碼器及上下文模型的系統。系統亦可以包括具有內部記憶器之編碼器晶片

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(89)

，其使用於某些上下文，且外部記憶器使用於其它的上下文。

使用記憶器之位元產生器：

圖19指出具有使用記憶器之疊合位元產生器的解碼器。參照圖19，解碼器1900包含上下文模型1901、記憶器1902、PEM狀態對碼區塊1903、疊合位元產生器1905、記憶器1904及移相器1906。上下文模型1901之輸入包含來自疊合位元流產生器1905的解碼資料。移相器1906之輸入被耦合至記憶器1902之輸入。記憶器1902之輸出被耦合至PEM狀態對碼1903。PEM狀態對碼1903之輸出與來自對齊編碼資料之輸出被耦合至位元產生器1905的輸入。記憶器1904使用雙向匯流排亦被耦合至位元產生器1905。位元產生器1905之輸出為解碼資料。

上下文模型1901響應在其輸入上之編碼資料而輸出一上下文筐。使用上下文筐作為一位址以存取記憶器1902，根據上下文筐以得到概率狀態。概率狀態是被PEM狀態對碼模組1903所接收，其響應概率狀態而產生概率等級。然後使用概率等級作為一位址以存取記憶器1904而得到運轉計數。然後由位元產生器1905使用此運轉計數以產生解碼資料。

在一實施例中，記憶器1902包含 1024×7 位元記憶器（其中1024為不同上下文的數目），而記憶器1904包含 25×14 位元記憶器（其中25為不同運轉計數之數目）。

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明(90)

由於位元產生器狀態(運轉計數等等)是與概率等級有關而非上下文筐，所以在有一位元可用上下文模型之前，會有一額外的疊合延遲。因為更新一位元產生器狀態需要多個時鐘循環(位元產生器狀態記憶器會再訪延遲)，對於各概率等級將使用多個位元產生器狀態。例如，如果疊合為六個時鐘循環，則位元產生器在每個概率等級將具有六個進入。使用一計數器以選定適當的記憶器位置。即使每個概率等級具有多個進入，記憶器之尺寸將典型地少於上下文之數目。能以SRAM之多觸排或多阜暫存器檔來實施記憶器。

由於運轉計數可以是與多個上下文有關，一個實施例必須升級一個或以上的上下文之概率估計狀態。在一實施例中，使運轉結束的上下文之PEM狀態被更新。

在可以被再讀取之前，並非要求一運轉計數之讀取、修改及寫入，只要修改一完成則可以再度使用一運轉計數。

圖32指出本發明之一個實施例中的解碼操作的時間圖。參照圖32，敘述一個三循環解碼操作。信號名稱被列在時間圖的左行上。於任意循環期間的一個信號之有效是以循環(或其部份)期間的條(bar)表示。在某些情形中，表示用於產生信號或供給有效信號之單元或邏輯，與虛線盒中的有效信號指示相鄰。經常亦提供所揭示的特定元件與單元之例子。須注意延伸進入另一個循環的信號部份，表示信號只對於信號延伸進入其它循環的時間周期是有效。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(91)

而且，指出某些信號是對於一個以上的循環有效。一個例子為暫時運轉計數信號在第二循環結束的一點上是有效，然後再度於第三循環期間是有效的。須注意此表示信號只是正在暫存在循環的結束。依附關係之列表亦表示於以下的表17中，提出來自相同或先前的時鐘循環之依附關係至目前時間，在此時間信號被指定為有效。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(92)

表 17

名稱	單元	相依**
暫存器檔 1	CM	{先前位元, CM移位暫存器}
狀態對碼	CM	暫存器檔 1
桶形移位	SH	{累積器暫存器, 未對齊的編碼資料暫存器}
尺寸	SH	桶形移相器輸出(對齊的編碼資料)
acc(累積器)	SH	尺寸{先前的累積器暫存器值}
暫存器檔 2	BG	{K, R3暫存}
α (需要的碼)	BG	暫存器檔 2
碼至(罩, MaxRL, R3split)	BG	{K, R3暫存}
產生器位元	BG	暫存器 2 桶形移相器輸出(對齊的編碼資料) 碼至(罩, MaxRL, R3split) {暫存器檔 1, 暫存MPS}
解碼	BG	桶形移相器輸出(對齊的編碼資料) 碼至(罩, MaxRL, R3split)
PEM表	PEM	{K, R3暫存}
β (PEM更新)	PEM	{暫存: PEM表輸出, LPS表示, 連續}
-1(運轉計數更新)	BG	{暫存: 需暫存之碼字、運轉計數、LPS表示連續}
γ (連續LPS表示更新)	BG	{暫存: 需要的碼字、運轉計數、LPS表示, 連續}

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (93)

*CM=上下文模型，SH=移相器，BG=位元產生器，PEM=概率估計機器。

**[italics]意指先前時鐘循環的依附關係。

++在一實施例中，用於更新PEM之組合邏輯是執行於"PEM表"步驟中，"PEM更新"只是一多工器操作。

隱含發信：

在某些實施例中，解碼器必須模組編碼器之有限的重序緩衝器。在一實施例中，此模組化是以隱含發信來完成。

如前面關於編碼器之說明，當在編碼器中開始一碼字時，空間被保留在碼字之適當的緩衝器中，其順序為碼字必須放置在通道上。當緩衝器中的最後空間是保留給一新的碼字時，某些碼字被放置在壓縮位元流中，無論是否已完成決定。

當必須完成一疊合碼字時，可以選擇一碼字，其是短的且正確地指定目前所接收符號。例如，在R-編碼器系統中，如果需要永久地完成一碼字，用於具有128最大運轉長度之運轉碼中的一系列100MPS，然後可以使用128MPS之碼字，由於正確地指定前100個符號。

另外，可以使用指定LPS之後的100MPS之碼字。當已完成碼字時，其可以從重序緩衝器被移除並被加至碼系統。此可以允許先前完成的碼字被放置於碼流中。如果強迫一部份碼字之完成，導致從滿緩衝器移除一碼字，然後可

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (94)

以繼續編碼。如果緩衝器仍是滿的，則下一個碼字再度必須被完成且加至碼流。此處理繼續，直到滿的緩衝器不再是滿的。解碼器可以模組編碼器用於隱含發信，使用一計數器用於各位元產生器狀態資訊記憶器位置。

在一實施例中，各運轉計數(此例子中的概率等級)具有一計數器，其是與編碼器中的頭或尾計數器相同的尺寸(例如6位元)。每次一新運轉計數開始時(取得一新的碼字)，對應的計數被載入碼字記憶器之尺寸。每次一運轉開始(取得一新碼字)，則全部計數器被增數。到達零之任意計數會使對應的運轉計數被清除(且LPS表示旗標被清除)。

有限記憶器之發信選擇：

本發明中之即時編碼要求解碼器處理MPS的運轉，此MPS並非跟在LPS之後且並非最大運轉長度。當編碼器開始PMS之運轉時其會發生，但是並不具有足夠的有限重序記憶器，以等待直到運轉完成。此情況要求欲被解碼之新碼字，下一次使用此上下文筐，且此情況必須被發信至解碼器。以下敘述修改解碼器的三種方式。

當緩衝器是滿的，被強迫送出的概率等級或上下文筐之運轉計數必須被復置。欲有效地實施，將上下文筐或概率等級儲存於碼字記憶器中是有用的。由於對於運轉只需要不具有相關的碼字，所以可以分享記憶器以用於儲存碼字。須注意在某些系統中，並非強迫送出一不完整的碼字

五、發明說明 (95)

，位元可以被迫送入碼字之上下文 / 概率等級，其仍然在緩衝器中，當緩衝器是滿的。解碼器偵測此，並使用對應（錯誤）的上下文筐或概率等級。

流發信是使用碼字來發信解碼器。在一實施例中， $R2(k)$ 及 $R3(k)$ 碼定義被改變以包括並非跟在LPS之後的MPS之非最大長度運轉。此實施可以藉由將一位元加至應發生最低概率的碼字。如此允許非最大長度運轉計數之獨特的可解碼預先固定(prefix)。表18指出可允許流發信之 $R2(2)$ 碼的更換。此方法之缺點為R-碼解碼邏輯必須被改變，且每一次發生具有最低概率之碼字時會有一壓縮價格。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (96)

表 1 8

原始資料	碼 字
0000	0
0001	1000
001	101
01	110
1	111
000	100100
00	100101
0	10011

在某些實施例中，解碼器使用時間標記來執行隱含發信。一計數器追蹤目前的「時間」，藉著每次要求一碼字時增量。而且，每當開始一碼字時，目前的「時間」被儲

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (97)

存於與碼字相關的記憶器中。在第一次使用碼字之後，將目前的「時間」與對應的儲存「時間」值加上編碼器之重序緩衝器的尺寸比較。如果目前的「時間」較大，則產生一隱含信號以要求一新的碼字。於是，已模擬編碼器中的有限重序記憶器。在一實施例中，使用足夠的位元用於「時間」值，以允許全部的碼字被標以數字。

欲降低所需的記憶器，時間標記所使用的位元之數目被保持在最小值。如果時間標記使用小數目的位元，使時間值被重新使用，則必須小心注意在計數器開始重新使用時間之後的全部舊時間標記。讓 N 大於位元產生器狀態記憶器或佇列之位址位元的數目。可以使用 $N+1$ 位元的時間標記。位元產生器狀態記憶器必須支援多存取，或許為每解碼位元有兩個讀取及兩個寫入。使用一計數器以循環全部的位元產生器狀態記憶器，對於各解碼後的位元增量一次。太舊的記憶器位置被清除，使得當其在將來被使用時可取得一新的碼字。如此保證全部的時間標記在時間值被重新使用之前被檢查。

如果位元產生器狀態記憶器小於佇列，則所需的計數速率（時間標記計數器）及記憶器頻帶寬度可以降低。這是因為各時間標記（每位元產生器狀態記憶器一個）必須被檢查，只要每數目的循環需要使用整個佇列。而且將時間標記儲存於不同的記憶器中會降低所需的記憶器頻帶寬度。在使用部份運轉之「0」碼字的系統中，對於「1N」碼字並不需要檢查時間標記。在使用部份運轉之「1N」碼

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (98)

字的系統中，只有在產生LPS之前必須檢查時間標記。

在某些實施例中，於解碼期間隱含發信是以一佇列來實施。此方法可用於半雙工系統中，用於編碼之硬體於解碼期間是有用的。佇列之操作幾乎是與解碼期間相同。當要求一碼字時，其索引被放置於佇列中並被標記為無效。當來自碼字之資料完成時，其佇列進入被標記為有效。當資料被取出佇列以留出空間給新碼字時，如果取出的資料被標記為無效，來自索引之位元產生器狀態資訊被清除。此清除操作需要位元產生器狀態記憶器能夠支援額外的寫入操作。

相反地隱含發信與緩衝器溢流互通作為壓縮資料。一個例子是具有輔助的上下文筐，其對於每個正常上下文筐使用一次，或著對每個被解碼的碼字使用一次。來自輔助上下文筐之解碼後的位元指出是否發生需要新碼字的情況，且對於對應的正常上下文筐必須解碼一新碼字。在此情形中，此特殊上下文的碼字必須被適當地重序。由於此上下文之使用為重序單元所習知的一種功能（典型地，對於各碼字使用一次），重序輔助上下文所需之記憶器可以隱含地被定界線或模組化。而且，可以限定此輔助上下文所允許之可能碼。

隱含發信模組編碼器之有限的緩衝器，其被解碼以產生一信號時，其指出必須解碼一新碼字。在一實施例中，編碼器之有限尺寸重序緩衝器被直接模組化。在半雙工系統中，由於編碼器之重序電路於解碼期間是有用的，其可

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明(99)

以被使用以產生用於解碼器之信號。

隱含發信如何正確地完成是視編碼器如何辨識及處理滿的緩衝器情況而定。對於使用具有固定配置之合併佇列的系統而言，多頭標記之使用可允許選擇那一種"緩衝器滿"方式。給定編碼器之一種設計，可以設定合適的模型。

以下提供編碼器操作及一模型用於具有固定流指定的、合併佇列之解碼器，與概率系統平行。對於此例子，假設重序緩衝器具有156位置，使用4個插入流，且各插入字為16位元。當緩衝器包含256位元時，一進入必須於第257個進入可以放置於佇列中之前，被送出至位元包封(例如位元封包單元)。如果需要的話，可較早地強迫輸出進入。

在某些系統中，移除緩衝器中的第一進入，需要移除足夠的位元以完成整個插入碼字。所以，如果1位元的碼字是可能的，移除碼字0將需要亦移除16位元的插入字之碼字4、8、12...、52、56、60。欲保證全部這些緩衝器進入具有有效的進入，強迫一進入被填入，因為記憶器是滿的，可以在距離新碼字進入的位置64、192位置($256-16 \times 4=192$)執行。

在解碼器中，各概率有一計數器。當使用新碼字以開始運轉時，此計數器被載入192。只要由任何概率來使用新碼字，則決定全部的計數器。如果任何計數器到達零，則此概率之運轉長度被設定為零(且LPS表示旗標被清除)。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (100)

使用多 RAM 觸排 (多卓記憶器，以快速記憶器模擬等等) 是方便的，各編碼資料流有一觸排。此允許全部的位元封包單元可以同時接收資料，所以讀取特殊流的多碼字並不會禁止其它流的讀取。

在其它的系統中，多位元封包單元必須仲裁單一記憶器，根據儲存於緩衝器中的碼字順序。在這些系統中，從一緩衝器移除進入並不會完成一插入字。各位元封包單元典型地依序接收一插入字的某些部份。各位元封包單元接收至少一數目的位元，其等於最短的碼字長度 (例如 1 位元)，及至少一數目的位元，其等於最長的碼字長度 (例如 13 位元)。插入字無法被送出，直到它們完成，且必須是以啓始的順序被送出。在此例子中，一位元封包單元必須緩衝 13 個插入字，其為以最大長度碼字可以完成之插入字的最大數目，而另一流具有一插入字待解決，其正接收最小長度碼字。

對於硬體實施而言，可以執行兩個寫入與兩個讀取之系統是優於每一個碼字需要記憶器的兩個寫入及一個讀取之系統。如果對於具有四個流之範例系統而言這是最好的，則位元封包單元 1 與 2 可以分享一個記憶器讀取循環，且位元封包單元 1 與 3 可以分享另一個讀取循環 (或是任何其它的任意組合)。雖然此將不會降低緩衝所需要的尺寸，其將可允許較高的轉換速率進入位元封包單元。如此允許位元封包單元可以更理想地使用編碼資料通道之容量。具有固定尺寸記憶器之系統：

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (101)

包含每一個概率等級具有多位元產生器狀態之系統的一個優點為：系統可以支援有耗損的編碼，當一固定尺寸的記憶器溢流時。對於圖框緩衝器之影像壓縮而言這將是有用的，且其它的應用只能儲存有限量的編碼資料。

對於具有固定尺寸記憶器的系統而言，各概率之多位元產生器狀態各指定給一部份的資料。例如，各八個狀態可以被指定給八個位元資料的特殊位元平面。在此情形中，移相器亦被指定給各部份的資料，與依序提供下一個碼字之移相器相反。須注意資料並不需要被位元平面分割。而且，在編碼器中，沒有執行插入，各部份的資料只是被位元包封。記憶器被配置給各部份的資料。

編碼資料之記憶器管理出現於將全部資料儲存於固定尺寸記憶器中的系統，及以最大可允許頻帶寬度在一通道中傳送資料的系統。在這些系統中，最好是適度變差對有損耗的系統。對於具有不同重要性的資料，使用不同流的資料，使得較不重要的流可以被儲存或著不被傳送，當並沒有足夠的儲存或頻帶寬度時。

當使用記憶器時，編碼資料被必須被儲存所以可以被存取，使得較不重要的資料流可以被拋棄，而不會喪失解碼重要資料流的能力。由編碼資料為可變長度，可以使用動態記憶器配置。圖31指出用於三個編碼資料流之動態記憶器配置的例子。一暫存器槽3100(或其它儲存)容納各流之指標加上用於指出下一個自由記憶器位置之另一個指標。記憶器3101被分割成固定尺寸頁。

五、發明說明 (102)

啓始地，指定給一流之各指標係指向記憶器之頁的開始，及記憶器之下一個可用頁的自由指標。來自特殊流之編碼資料被儲存在由對應的指標定出位址的記憶器位置。然後此指標被增量至下一個記憶器位置。

當指標到達目前頁之最大值時則發生以下情形。下一個自由頁之開始的位址是以目前頁來儲存。可以使用部份的編碼資料記憶器或分開的記憶器或暫存器檔。目前的指標被設定至下一個自由頁。此自由指標被增量。這些動作配置一新頁的記憶器至一特殊流，並提供鏈結使得於解碼期間可以決定配置之順序。

當記憶器中之全部頁是在使用中，且來自一流有更多的資料，此資料是比記憶器中的資料更為重要，則可以做成以下三件事情的其中之一。在全部的三種情形中，指定給較不重要的資料流之記憶器，被重新指定給較重要的資料流，且不再有來自較不重要的資料流之資料被儲存。

首先，目前正被較不重要的資料流所使用的頁只被指定給較重要資料。由於大部份典型的一致性編碼器使用中間狀態資訊，先前儲存於此頁中的全部較不重要的資料會遺失。

第二，目前正被較不重要的資料流所使用的頁只被指定給較重要資料。不像先前的情形，指標被設定至頁之結束，當更多重要的資料被寫入於頁中，對應的指標被增量。如此具有優點為可保留較不重要的資料在頁的開始處，如果較重要的資料流並不需要整頁。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (103)

第三，並非較不重要的資料之目前頁被重新指定，而是較不重要的資料之任意頁可以被重新指定。如此需要全部頁之編碼資料被獨立地編碼，其可以降低所達成的壓縮。其亦需要對應全部頁之開始的未編碼資料被辨識。由於可以拋棄任意頁的較不重要資料，在適度變差對有損耗編碼中的較大彈性是有用的。

第三種方式在影像的區域上可達成固定速率的壓縮之系統中是非常吸引人的。一指定數目的記憶器頁可以被配置給一區域的影像。較不重要的資料是否被保持是視在特殊區域中所達成的壓縮而定。指定給一區域之記憶器可以不是被完全使用，如果所需要的無損耗壓縮小於指定記憶器的量。在一區域的影像上達成固定速率的壓縮可以支援對影像區域之隨機存取。

從兩端將資料寫入各頁之能力可以被使用以較佳地利用系統中之可用的記憶器之總量。當全部頁被配置時，在尾端具有足夠自由空間之頁可以被配置而從尾端使用。使用頁之兩端的能力可以與保持位置之追縱的價格平衡，在此位置符合兩種型式的資料。這是不同於其中一種資料型式是較不重要且可以被重寫的情形。

現在考慮一流其中資料被傳送於通道中而非被儲存於記憶器中。使用固定尺寸頁的記憶器，但是每一流只需要一頁（或者兩頁，如果需要交替以提供通道之緩衝的話，使得當寫入至其中一個時，另一個可以被讀取而輸出）。

當一頁的記憶器是滿的，其被傳送於通道中，且只要此頁

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (104)

被傳送則可以馬上重新使用記憶器配置。在某些應用中，記憶器之頁尺寸可以為使用於多包封尺寸的通道中之資料包封的尺寸。

在有些通信系統中，例如 ATM (非同步傳送模型)，優先順序可以被指定給包封。ATM 具有兩種優先順序位準：優先與次等。次等包封只有當有足夠的頻帶寬度時才被傳送。使用一臨界以決定那些流為優先而那些流為次等。另一種方法為使用在編碼器之臨界，而不傳送比臨界更不重要的流。

各碼之分開的位元產生器：

圖 20 為具有各碼之分開的位元產生器之系統的方塊圖。參照圖 20，解碼系統 2000 包含上下文模型 2001、記憶器 2002、PEM 狀態對碼區塊 2003、解碼器 2004、位元產生器 2005A-n、及移相器 2006。上下文模型 2001 之輸出被耦合至記憶器 2002 之輸入。記憶器 2002 之輸出被耦合至 PEM 狀態對碼區塊 2003 之輸入。PEM 狀態對碼區塊 2003 之輸出被耦合至解碼器 2004 之輸入。解碼器 2004 之輸出被耦合作為位元產生器 2005A-n 之致能信號。位元產生器 A-n 亦被耦合以接收從移相器 2006 輸出之編碼資料。

上下文模型 2001、記憶器 2002 及 PEM 狀態對碼區塊 2003 是與圖 19 中之對應元件類似地操作。上下文模型 2001 產生一上下文筐。記憶器 2002 根據上下文筐而輸出一概率狀態。此概率狀態被 PEM 狀態對碼區塊 2003 所接收，其對

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (105)

於各概率狀態產生一概率等級。解碼器 2004 在解碼概率等級時致能其中一個位元產生器 2005A-n (須注意解碼器 2004 為 M 對 2^M 解碼器電路，類似於習知的 74×138 個 3:8 解碼器，其並非一致性編碼解碼器)。須注意由各碼具有一分開的位元產生器，某些位元產生器可以使用 R-碼以外的碼。特別是，可以使用概率接近 60% 之碼，以更理想地堆疊概率空間於 $R2(0)$ 與 $R2(1)$ 之間。例如，表 19 敘述此種碼。

表 19

未編碼資料	碼字
000	00
001	01
01	10
1	11

如果需要達成理想的速度，可以做成一個或更多的位元之預先解碼，以保證解碼資料可快速地使用。在某些實施例中，欲避免需要在每一時鐘循環更新大的運轉計數，

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (106)

長碼之運轉計數與碼字解碼被分開。

$R2(0)$ 之位元產生器是不複雜的。每次要求一位元時則要求一碼字。所產生的位元簡單地為碼字(以MPS來XOR操作)。

短的運轉長度之碼，例如 $R2(1)$ 、 $R3(1)$ 、 $R2(2)$ 及 $R3(2)$ ，是以如下之方式被處理。一碼字中的全部位元被解碼並被儲存於一狀態機器中，其包含小計數器(分別為1、2或3位元)及LPS表現位元。計數器及LPS表現位元操作如同一R-碼解碼器。

對於較長的碼，例如 $k > 2$ 之 $R2(k)$ 與 $R3(k)$ ，位元產生器被分割成如圖21所示的兩個單元。參照圖21，指出 $k > 2$ 之 $R2(k)$ 碼的位元產生器結構，具有一短的運轉單元2101及一長的運轉單元2102。須注意雖然此結構是與 $R2(k > 2)$ 碼一起使用，其操作將類似 $R3(k > 2)$ 碼(且對熟於此技藝者是明顯的)。

短的運轉單元2101被耦合以接收致能信號及碼字 $[0..2]$ 作為位元產生器之輸入，及一"全零"信號與一"計數零"信號(指出計數零)，二者均來自長的運轉單元2102。響應這些輸入，短的運轉單元2101輸出一解碼位元及下一個信號指示，其發信一需要的新碼字。短的運轉單元2101亦產生一計數致能信號、一計數負載信號及一計數最大信號至長的運轉單元2102。長的運轉單元2102亦被耦合以接收碼字 $[k...3]$ 作為位元產生器之輸入。

短的運轉單元2101處理長度4之運轉，且類似一 $R2(2)$

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (107)

位元產生器。在一實施例中，對於全部 $R2(k>2)$ 而言，短的運轉單元 2101 是相同的。長的運轉單元之目的在於決定何時輸出運轉之最後 1-4 位元。長的運轉單元 2102 具有輸入、AND 邏輯及計數器其隨著 k 而改變尺寸。

圖 22 指出長的運轉計數單元 2102 之實施例。參照圖 22，長的運轉單元 2102 包含 AND 邏輯 2201 被耦合以接收碼字 $[k...3]$ 並輸出 "全 1" 信號作為邏輯 1，如果碼字中的全部位元為 1，藉以指出目前的碼字為 1N 碼字且運轉計數小於 4。NOT 邏輯 2202 亦被耦合以接收碼字並使其反相。NOT 邏輯 2202 之輸出被耦合至位元計數器 2203 之輸入。位元計數器 2203 亦被耦合以接收計數致能信號、計數負載信號及計數最大信號。響應輸入，位元計數器 2203 產生一計數零信號。

在一實施例中，計數器 2003 為 $k-2$ 位元計數器，且用以將長的運轉計數分開成 4MPS 之運轉及可能某些餘數。計數致能信號指出 4MPS 已被輸出且計數器必須被減量。當解碼 "1N" 碼字時使用計數負載信號，並使計數器被載入位元 k 至 3 的碼字位元之補數。當解碼 "0" 碼字時使用計數最大信號，並使計數器載入其最大值。計數零輸出信號指出計數器為零。

圖 23 指出短的運轉計數單元 2101 之實施例。參照圖 23，短的運轉計數單元包含控制模組 2301、二位元計數器 2302 及三位元計數器 2303。控制模組 2301 接收致能信號、碼字 $[0...2]$ 、及來自長的運轉計數單元之計數零信號與

五、發明說明 (108)

全部 1 信號。使用二位元計數器以計數 MPS 之 4 位元運轉，其為較長的運轉之部份。使用 R2(2) 計數器及 LPS 位元 (總數三位元) 2303，以在運轉之結束產生 1-4 位元。致能輸入指出一位元應被產生在位元輸出上。當沒有被確認時計數零輸入指出 4MPS 的運轉應被輸出。每當 MPS 計數器 2302 到達零時，計數致能輸出被確認。當計數零輸入被確認時，使用 R2(2) 計數器 LPS，或者解碼一新碼字且下一個輸出被確認。

當新碼字被解碼時，是由碼字輸入來決定所執行的動作。如果輸入為 "0" 碼字，則使用 MPS 計數器 2302 且確認計數最大輸出。對於 "1N" 碼字，碼字之前三個位元被載入 R 2(2) 計數器及 LPS 2303，且確認計數負載輸出。如果確認為全部 1 輸入，則使用 R2(2) 計數器及 LPS 2303 以產生位元；或者使用 MPS 計數器直到確認計數零輸入。

為了使系統工作良好，來自各別的系統之碼的數目很小，典型地為 25 或更小。為了快速操作，位元及下一個碼字輸出所需的多工器之尺寸與用於致能特殊位元產生器之解碼器必須被限制。而且，為了高速操作，來自移相器之碼字的扇出必須不能太長。

各碼之分開的位元產生器允許疊合。如果全部碼字導致至少二位元，碼字之處理可以疊合於兩個循環而非一個循環。如此會使解碼器之速度加倍，如果位元產生器為系統之限制部份。一種完成方式為：運轉長度零碼字 (只指出 LPS 之碼字) 之後跟隨著一位元，此位元為下一個未編

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (109)

碼位元。這些稱為 $RN(k)+1$ 碼且總是至少編碼二位元。須注意 $R2(0)$ 碼字或某些其它的短碼字並不需要被疊合以加速。

分開的位元產生器本身可以使用於隱含發信。使用以下方式可以完成以有限記憶體器來編碼之隱含發信。當使用尺寸為 512 佇列時，各位元產生器具有一計數器其為佇列位址之尺寸例如 9 位元。位元產生器每次使用一新碼字時，計數器被載入最大值。每次位元產生器要求一碼字時，全部位元產生器之計數器被減量。每次計數器到達零時，對應的位元產生器之狀態被清除（例如，MPS 計數器、 $R2(2)$ 計數器與 LPS 及長的運轉計數計數器被清除）。因為即使特殊的位元產生器並沒有被致能亦可以發生清除，所以舊的計數不會有問題。

各上下文筐之記憶器的啓始：

在各上下文筐之記憶器保持概率估計資訊的情形中，要求額外的記憶器頻帶寬度以非常快速地啓始解碼器（例如記憶器）。快速地啓始解碼器會發生問題，當解碼器有很多上下文且它們需要被清除。當解碼器支援很多上下文（1K 或更多）且記憶器無法全部被清除時，則需要非常大的時鐘循環來清除記憶器。

欲快速地清除上下文，本發明之某些實施例使用外位元，在下文中稱為啓始狀態位元，其被儲存於各上下文。於是，對於各上下文而言，外位元儲存有 PEM 狀態（例如 8

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (110)

位元)。

各上下文筐之記憶器及啓始控制邏輯是表示於圖24。參照圖24，上下文記憶器2401被耦合至一暫存器2402。在一實施例中，暫存器2402包含一位元暫存器，其指出啓始狀態筐之目前的合適狀態。暫存器2402被耦合至XOR邏輯2403的一個輸入。XOR邏輯2403的另一個輸入被耦合至記憶器2401的輸出。XOR邏輯2403之輸出為有效信號，且被耦合至上下文筐信號與計數器2405之輸出。控制邏輯2404之輸出被耦合至MUX2406-2407之選定輸入，且被耦合至計數器2405的輸入。控制邏輯2404的另一個輸出被耦合至MUX2408的選定輸入。MUX2406之輸入被耦合至上下文筐指示及計數器2405的輸出。MUX2406之輸出被耦合至記憶器2401。MUX2407之輸入被耦合至新PEM狀態及零。MUX2407之輸出被耦合至記憶器2401外的記憶器2401之輸入，且啓始PEM狀態被耦合至MUX2408的輸入。MUX2408之輸出為PEM狀態輸出。

每次發生解碼操作時(亦即各資料設定，不是各解碼位元)，暫存器2402中之值成為補數。XOR邏輯2403比較存取的記憶器位置之有效與暫存器值，以決定對於此解碼操作而言存取記憶器位置是否有效。它是使用XOR邏輯2403以檢查啓始狀態位元是否符合暫存器2402中之適當狀態。如果記憶器2401中之資料並非有效，則控制邏輯2404使資料被狀態對碼邏輯所忽略，而使用啓始PEM狀態取代。其是使用MUX2408而完成。當一新PEM狀態被寫入記憶器，啓

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (111)

始位元被設定為暫存器之目前值，使得當再度存取時其將被視為有效。

每一個上下文筐記憶器進入在另一個解碼操作可以開始之前，必須使其啓始狀態位元設定為暫存器之目前值。計數器 2405 步驟經過全部記憶器位置以確保它們被啓始。每當使用上下文筐，其 PEM 狀態入沒有被更新，則可以使用未使用的寫入循環以測試或更新指定給計數器 2405 之記憶器位置。在解碼操作完成之後，如果計數器 2405 沒有到達最大值，則其餘的上下文在開始下一個操作之前被啓始。使用以下邏輯來控制操作。

```

write_it = false;
counter = 0;
all_initialized = false;
while (counter < maximum context bin+1)
    read PEM state from context memory
    if ( (counter == context bin read) and (write_it) )
        write_it = false
        counter = counter + 1
    if ( (PEM state changed)
        write new PEM state
    else if (write_it)
        write initial PEM state to memory location "counter"
        counter = counter + 1
    else
        read memory location "counter"
        if (initialized bit in read location is in wrong state)
            write_it = true
        else
            counter = counter + 1
all_initialized = true;
while (decoding)
    read PEM state from context memory
    if (PEM state changed)
        write new PEM state

```

具有快速配合之 PEM：

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (112)

使用於本發明中的 PEM 可以包括一配合圖，以不管有用資料的量而可允許快速配合。藉由如此，本發明允許解碼可更快速地配合啓始，且當更多的資料有用時可更緩慢地配合，作為用於提供更多精確的估計之手段。此外，PEM 可以被固定於欄位可程式規劃陣列 (FPGA) 或 PEM 狀態表 / 機器的 ASIC 實施。

以下的表 20-25 敘述許多概率估計狀態機器。某些表並不是使用 R3 碼或者不是使用長碼，以降低硬體價格。全部的表除了表 20 是使用 "快速配合" 特殊狀態，其在編碼開始時可以快速配合，直到第一個 LPS 發生。這些快速配合狀態在所示的表中被啓始。例如，參照表 21，當解碼開始時，目前的狀態為狀態 0。如果一 MPS 發生，則解碼器傳送至狀態 35。只要 MPS 發生，解碼器傳送狀態 35 的上面，最後傳送至狀態 28。如果在任何時間發生 LPS，解碼器傳送快速配合狀態至一狀態，此狀態代表資料之正確概率狀態，該資料已被接收。

須注意對於各表，在某一數目的 MPS 已被接收之後，解碼器轉移快速配合狀態。在理想的實施例中，一旦已存在快速配合狀態，沒有機構可以將它們送回，除了重新開始解碼處理。在其它的實施例中，藉著允許更快的配合，可以設計狀態表以重新進入這些快速配合狀態，本發明允許解碼器可更快速地到達更多的斜碼，藉以可從改良的壓縮而得到優點。須注意對於一特殊表，藉著改變表進入為目前的狀態 0，可以去除快速配合，使得表轉移只有一狀

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (113)

態向上或向下，視資料輸入而定。

對於全部的表，各狀態之資料為此狀態之碼，此狀態為正更新(up)上的下一個狀態及負更新(down)上的下一個狀態。星號表示MPS必須在一負更新上被改變之狀態。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (114)

表 20

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
0	r2(0)	1	0*
1	r2(0)	2	0
2	r2(0)	3	1
3	r2(0)	4	2
4	r2(0)	5	3
5	r2(0)	6	4
6	r2(1)	7	5
7	r2(1)	8	6
8	r2(1)	9	7
9	r2(1)	10	8
10	r2(1)	11	9
11	r2(1)	12	10
12	r3(1)	13	11
13	r3(1)	14	12
14	r3(1)	15	13
15	r2(2)	16	14
16	r3(2)	17	15
17	r2(3)	18	16

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
18	r3(3)	19	17
19	r2(4)	20	18
20	r3(4)	21	19
21	r2(5)	22	20
22	r3(5)	23	21
23	r2(6)	24	22
24	r3(6)	25	23
25	r2(7)	26	24
26	r3(7)	27	25
27	r2(8)	28	26
28	r3(8)	29	27
29	r2(9)	30	28
30	r3(9)	31	29
31	r2(10)	32	30
32	r3(10)	33	31
33	r2(11)	34	32
34	r3(11)	34	33

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

5 *轉換至 MPS

五、發明說明 (115)

表 21

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
0	r2(0)	35	35*
1	r2(0)	2	1*
2	r2(0)	3	1
3	r2(0)	4	2
4	r2(0)	5	3
5	r2(0)	6	4
6	r2(1)	7	5
7	r2(1)	8	6
8	r2(1)	9	7
9	r2(1)	10	8
10	r2(1)	11	9
11	r2(1)	12	10
12	r3(1)	13	11
13	r3(1)	14	12
14	r3(1)	15	13
15	r2(2)	16	14
16	r3(2)	17	15
17	r2(3)	18	16
18	r3(3)	19	17
19	r2(4)	20	18
20	r3(4)	21	19
21	r2(5)	22	20

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
22	r3(5)	23	21
23	r2(6)	24	22
24	r3(6)	25	23
25	r2(7)	26	24
26	r3(7)	27	25
27	r2(8)	28	26
28	r3(8)	29	27
29	r2(9)	30	28
30	r3(9)	31	29
31	r2(10)	32	30
32	r3(10)	33	31
33	r2(11)	34	32
34	r3(11)	34	33
35	r2(0)	36	1*
36	r2(1)	37	2
37	r2(2)	38	4
38	r2(3)	39	6
39	r2(4)	40	10
40	r2(5)	41	16
41	r2(6)	42	19
42	r2(7)	43	22
43	r2(8)	28	25

* 轉換至 MPS

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (116)

表 22

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
0	r2(0)	35	35*
1	r2(0)	2	1*
2	r2(0)	3	1
3	r2(0)	4	2
4	r2(0)	5	3
5	r2(0)	6	4
6	r2(1)	7	5
7	r2(1)	8	6
8	r2(1)	9	7
9	r2(1)	10	8
10	r2(1)	11	9
11	r2(1)	12	10
12	r2(1)	13	11
13	r2(2)	14	12
14	r2(2)	15	13
15	r2(2)	16	14
16	r2(2)	17	15
17	r2(3)	18	16
18	r2(3)	19	17
19	r2(4)	20	18
20	r2(4)	21	19
21	r2(5)	22	20

* 轉換 MPS

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
22	r2(5)	23	21
23	r2(6)	24	22
24	r2(6)	25	23
25	r2(7)	26	24
26	r2(7)	27	25
27	r2(8)	28	26
28	r2(8)	29	27
29	r2(9)	30	28
30	r2(9)	31	29
31	r2(10)	32	30
32	r2(10)	33	31
33	r2(11)	33	32

35	r2(0)	36	1*
36	r2(1)	37	2
37	r2(2)	38	4
38	r2(3)	39	6
39	r2(4)	40	10
40	r2(5)	41	16
41	r2(6)	42	19
42	r2(7)	43	22
43	r2(8)	28	25

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (117)

表 23

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
0	r2(0)	35	35*
1	r2(0)	2	1*
2	r2(0)	3	1
3	r2(0)	4	2
4	r2(0)	5	3
5	r2(0)	6	4
6	r2(1)	7	5
7	r2(1)	8	6
8	r2(1)	9	7
9	r2(1)	10	8
10	r2(1)	11	9
11	r2(1)	12	10
12	r3(1)	13	11
13	r3(1)	14	12
14	r3(1)	15	13
15	r2(2)	16	14
16	r3(2)	17	15
17	r2(3)	18	16
18	r3(3)	19	17
19	r2(4)	20	18
20	r3(4)	21	19
21	r2(5)	22	20

* 轉換 MPS

目前狀態	碼	下一個 態狀 上	下一個 態狀 下
22	r2(5)	23	21
23	r2(6)	24	22
24	r2(6)	25	23
25	r2(7)	26	24
26	r2(7)	27	25
27	r2(8)	28	26
28	r2(8)	29	27
29	r2(9)	30	28
30	r2(9)	31	29
31	r2(10)	32	30
32	r2(10)	33	31
33	r2(11)	34	32
34	r2(11)	34	33
35	r2(0)	36	1*
36	r2(1)	37	2
37	r2(2)	38	4
38	r2(3)	39	6
39	r2(4)	40	10
40	r2(5)	41	16
41	r2(6)	42	19
42	r2(7)	43	22
43	r2(8)	28	25

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (118)

表 24

目前狀態	碼	下一個 態狀向 上	下一個 態狀向
0	r2(0)	35	35*
1	r2(0)	2	1*
2	r2(0)	3	1
3	r2(0)	4	2
4	r2(0)	5	3
5	r2(0)	6	4
6	r2(1)	7	5
7	r2(1)	8	6
8	r2(1)	9	7
9	r2(1)	10	8
10	r2(1)	11	9
11	r2(1)	12	10
12	r3(1)	13	11
13	r3(1)	14	12
14	r3(1)	15	13
15	r2(2)	16	14
16	r3(2)	17	15
17	r2(3)	18	16
18	r3(3)	19	17
19	r2(4)	20	18
20	r3(4)	21	19
21	r2(5)	22	20

目前狀態	碼	下一個 態狀向 上	下一個 態狀向 下
22	r3(5)	23	21
23	r2(6)	24	22
24	r3(6)	25	23
25	r2(7)	26	24
26	r2(7)	27	25
27	r2(7)	27	26

（請注意：本圖之主要部分係由本局代印）

35	r2(0)	36	1*
36	r2(1)	37	2
37	r2(2)	38	4
38	r2(3)	39	6
39	r2(4)	40	10
40	r2(5)	41	16
41	r2(6)	42	19
42	r2(7)	25	22

5 * 轉換 MPS

五、發明說明 (119)

表 25

目前狀態	碼	下一個 態上	下一個 態下
0	r2(0)	35	35*
1	r2(0)	2	1*
2	r2(0)	3	1
3	r2(0)	4	2
4	r2(0)	5	3
5	r2(0)	6	4
6	r2(1)	7	5
7	r2(1)	8	6
8	r2(1)	9	7
9	r2(1)	10	8
10	r2(1)	11	9
11	r2(1)	12	10
12	r2(1)	13	11
13	r2(2)	14	12
14	r2(2)	15	13
15	r2(2)	16	14
16	r2(2)	17	15
17	r2(3)	18	16
18	r2(3)	19	17
19	r2(4)	20	18
20	r2(4)	21	19
21	r2(5)	22	20

目前狀態	碼	下一個 態上	下一個 態下
22	r2(5)	23	21
23	r2(6)	24	22
24	r2(6)	25	23
25	r2(7)	26	24
26	r2(7)	27	25
27	r2(7)	28	26
28	r2(7)	28	27

35	r2(0)	36	1*
36	r2(1)	37	2
37	r2(2)	38	4
38	r2(3)	39	6
39	r2(4)	40	10
40	r2(5)	41	16
41	r2(6)	42	19
42	r2(7)	25	22
43	r2(8)	28	25

* 轉換 MPS

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (120)

將快速配合加至概率估計只能幫助編碼之開始。可以使用其它的方法於編碼期間來改良配合，當上下文筐之統計，以比前述PEM狀態表可以追縱更快的速度來改變。

整個編碼期間用於保持快速配合的一個方法是將加速項加至PEM狀態更新。此加速可以藉著重覆各碼一固定數目的次數(例如8)，而併入於PME狀態表中。然後加速項M(例如一正整數)當更新時可以從目前的狀態被加或減。當M為1時，系統是與沒有加速的情形相同地操作，且發生最慢的配合。當M大於1時，發生較快的配合。啓始地，M可以被設定為大於1之值以提供啓始快速配合。

本發明用於更新M之值的一種方法是根據連續碼字的數目。例如，如果連續地發生一預定數目的碼字，則M之值被增加。例如，如果四個連續碼字為"0" "0" "0" "0"或"1N" "1N" "1N" "1N"，則M之值被增加。另一方面，可以使用切換"0"與"1N"之圖形以增加M之值。例如，如果四個連續碼字為"0" "1N" "0" "1N"或"1N" "0" "1N" "0"，則M被減少。

另一種減速的方法是使用各碼被重複S次的狀態表，其中S為一正整數。S為反向加速參數。當S為1時，配合是快速的，且當S為更大時，則配合會更慢。S之值可以被啓始地設定為1以提供啓始快速配合。使用與上述類似的一種方法，S之值可以被更新，當四個連續碼字為"0" "0" "0" "0"，或"1N" "1N" "1N" "1N"。在此一情形中，S之值被減少。相反地，如果四個連續碼字為"0" "1N" "0" "1N"或"1N" "0" "1N" "0"，則S之值被增加。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (121)

連續碼字的定義可以有數種意思。在"以上下文"的系統中，連續碼字可視為一上下文筐中的連續碼字。在"以概率"的系統中，連續碼字可以視為一概率等級中的連續碼字。另外，在其它的系統中，連續碼字可以視為整體的連續碼字(不管是上下文筐或概率等級)。對於這三種例子，保持碼字之歷史所需之儲存位元分別為 $3x$ 上下文筐之數目、 $3x$ 概率等級之數目與3。各上下文筐之保持加速可以提供最好的配合。由於較差的追縱常常是由於未編碼資料之整體改變，決定整體地加速亦可提供良好的配合。

系統應用：

任何壓縮系統之一個優點在於可降低一組資料之儲存需求。本發明之平行系統可以取代由無損耗的編碼系統所實行的目前應用，且可以應用於可操作音頻、上下文、資料庫的系統、可執行的電腦、或其它的數位資料、信號或符號。範例的無損耗編碼系統包括傳真壓縮、資料庫壓縮、位元映像圖形影像之壓縮、及影像壓縮標準例如JPEG與MPEG中的變換係數之壓縮。本發明允許小的有效硬體實施，及非常快的軟體實施，使其可以良好地選擇，即使是用於不需要高速之應用。

本發明對於習知技術之真實優點在於能以非常高的速度操作，特別是用於解碼。以此方式，本發明可以完全使用昂貴的高速通道，例如高速計算機網路、衛星及陸地上的廣播通道。圖28指出此一系統，其中廣播資料或高速計

五、發明說明 (122)

算機網路將資料供給至解碼系統 2801，其平行地解碼資料以產生輸出資料。目前的硬體一致性(例如 Q-編碼器)將會使這些系統的產量變慢。全部這些系統是以高價格來設計，以具有高的頻帶寬度。計數器的多產會使解碼器減緩產量。本發明之平行系統不僅可以完成這些高頻帶寬度，亦可以實際地增加有效的頻帶寬度，因為資料能以壓縮的方式來傳送。

本發明之平行系統亦可應用於，在適當快速的通道如 ISDN、CD-ROM、SCSI 以外，得到更有效的頻帶寬度。此一頻帶寬度匹配系統是表示於圖 29 中，其中來自信號源例如 CD-ROM、Ethernet、小計算機標準介面(SCSI)或其它類似源之資料被耦合至解碼系統 2901，其接收資料並將資料解碼以產生一輸出。這些通道仍然比某些目前的編碼器更快。常常這些通道是用於接收資料源，其需要比通道更大的頻帶寬度，例如即時視頻或以計算機為基礎的多媒體。本發明之系統可以扮演頻帶寬度匹配之角色。

本發明之系統為一種用於即時視頻系統如高畫質電視(HDTV)與 PMEG 視頻標準之一致性編碼器部份的良好選擇。此一系統是表示於圖 30 中。參照圖 30，即時視頻系統包括解碼系統 3001，其被耦合以壓縮影像資料。系統 3001 將資料解碼並輸出至有損耗解碼器 3002。有損耗解碼器 3002 可以為 HDTV 或 MPEG 解碼器之傳送、色彩轉換及次取樣部份。監視器 3003 可以是一電視或視頻監視器。

在閱讀以上敘述之後，本發明之許多修改及其它對熟

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (123)

於此技藝之人仕而言是明顯的，須瞭解文中所述及指出的特殊實施例並不是用於限制本發明。所以，較佳實施例之細節部份並不是要限制申請專利範圍之範圍，其本身只指出本發明之較重要的特徵。

於是已敘述了用於平行地解碼及編碼資料的方法與裝置。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

四、中文發明摘要(發明之名稱：

編碼及解碼資料之方法及裝置)

敘述用於解壓縮及壓縮資料之方法及裝置。本發明提供一種編碼器，使用於具有解碼由編碼器所產生的資訊之解碼器的壓縮系統。本發明編碼器包括一編碼器其響應資料而產生碼字資訊。此編碼器亦包括一重序單元，其響應來自編碼器之碼字資訊而產生一編碼資料流。此重序單元包含一運轉計數重序單元，用於將碼字排列成解碼順序，及一位元封包單元可將可變長度碼字組合成固定長度插入字，而以解碼器所要求的順序而輸出固定長度插入字。

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

英文發明摘要(發明之名稱：

)

訂

線

六、申請專利範圍

1. 一種用於編碼資料流的方法，包含以下步驟：響應資料流而產生代表資料流之碼字資訊，其中碼字資訊包含許多碼字，且來自資料流之資料所產生的多碼字係被平行地處理；

響應碼字資訊而產生編碼資料，其中產生編碼資料之步驟包含碼字資訊中各該許多碼字被輸出，使得正被輸出之碼字的順序是根據由各碼字所代表的部份資料流之開始。

2. 如申請專利範圍第1項之方法，其中各該許多碼字在各運轉開始時被輸出。

3. 如申請專利範圍第1項之方法，其中產生編碼資料之步驟進一步包含將可變長度碼字組合成固定長度資料結構。

4. 如申請專利範圍第3項之方法，其中各固定長度資料結構包含一插入字。

5. 如申請專利範圍第1項之方法，進一步包含輸出編碼資料，使得碼字是以解碼順序被排序。

6. 如申請專利範圍第1項之方法，進一步包含排序碼字的步驟。

7. 如申請專利範圍第1項之方法，其中產生碼字資訊之步驟包含以下步驟：

產生碼字之概率狀態；

根據概率狀態來選擇位元產生碼；

存取記憶器以得到與概率等級有關的運轉計數。

六、申請專利範圍

8. 一種用於編碼資料流之編碼系統，該編碼系統包含：

一編碼器，被耦合以接收資料流，而產生包括碼字之碼字資訊，且其中從正被平行地處理的資料，由編碼器來產生多碼字；及

一重序單元，被耦合至編碼器以響應碼字資訊而產生編碼資料，其中重序單元重序由編碼器所產生的碼字成解碼器順序，使得根據由各碼字所代表的部份資料流之開始而輸出碼字之順序。

9. 如申請專利範圍第8項之編碼系統，其中重序單元儲存碼字，以方式可指出碼字尺寸。

10. 如申請專利範圍第9項之編碼系統，其中1N碼字被重序，使得大部份明顯"1"位元指出各碼字之長度。

11. 如申請專利範圍第8項之編碼系統，其中重序單元包含：

至少一運轉計數重序單元，在各運轉開始時，產生碼字資訊中的各該許多碼字；及

至少一位元封包單元，將可變長度碼字組合成固定長度資料結構。

12. 如申請專利範圍第11項之編碼系統，其中各固定長度資料結構包含一插入字。

13. 如申請專利範圍第8項之編碼系統，其中重序單元使碼字重序。

14. 如申請專利範圍第13項之編碼系統，進一步

六、申請專利範圍

包含一記憶器，被耦合至重序單元以一順序來儲存碼字。

1 5 . 如申請專利範圍第 8 項之編碼系統，其中編碼器包含：

一上下文模型；

一概率估計機器，被耦合至上下文模型；及

一位元產生器結構，被耦合至概率估計機器，以響應資料流而產生碼字。

1 6 . 如申請專利範圍第 1 5 項之編碼系統，其中編碼器進一步包含儲存許多運轉計數之記憶器，且使用來自概率估計機器之概率等級而存取記憶器，以提供其中一個許多運轉計數至位元產生器，而輸出部份碼字資訊。

1 7 . 如申請專利範圍第 1 5 項之編碼系統，其中位元產生器結構包含：

一位元產生器，響應索引及 MPS/LPS 指示而提供碼字資訊；及

一記憶器，被耦合至位元產生器，以提供一運轉計數至位元產生器，其中位元產生器根據索引而讀取記憶器，並根據從讀取記憶器所得到的資料而執行位元產生。

1 8 . 如申請專利範圍第 1 7 項之編碼系統，其中碼字資訊包含第一信號其指出 MPS/LPS 指示是否包含運轉之開始、第二信號其指出 MPS/LPS 指示是否包含運轉之結束、及一碼字輸出。

1 9 . 如申請專利範圍第 1 1 項之編碼系統，其中位元封包邏輯將插入字排序，以產生編碼資料流作為插入流

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

，此插入流的排序是依據各流之各插入字中的第N個先前碼字。

2 0 . 如申請專利範圍第 1 1 項之編碼系統，其中重序單元包含一 snooper 解碼器，用於選定插入字輸出至碼流。

2 1 . 如申請專利範圍第 2 0 項之編碼系統，其中重序單元包含許多運轉計數重序單元，被耦合至許多位元封包單元，其中各許多位元封包單元產生插入字，且 snooper 解碼器從許多插入字選定一插入字作為輸出至碼流。

2 2 . 如申請專利範圍第 1 1 項之編碼系統，其中碼字資訊包括一時間標記，且重序單元進一步包含邏輯以根據相關的時間標記而輸出插入字。

2 3 . 如申請專利範圍第 2 2 項之編碼系統，其中重序單元包含許多運轉計數重序單元，被耦合至許多位元封包單元，且該邏輯根據相關的時間標記而使插入字被輸出。

2 4 . 如申請專利範圍第 2 3 項之編碼系統，其中插入字是根據最舊的時間標記而被輸出。

2 5 . 如申請專利範圍第 1 1 項之編碼系統，其中單一佇列供給碼字至許多位元封包單元，且該許多位元封包單元產生插入字，輸出作為部份的碼流。

2 6 . 如申請專利範圍第 2 5 項之編碼系統，其中單一佇列包括單一運轉計數重序單元，且該邏輯決定欲輸出作為碼流的下一個插入字。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

27. 如申請專利範圍第11項之編碼系統，其中單一佇列供給碼字至單一位元封包單元。

28. 一種使用於壓縮系統之編碼系統，此壓縮系統具有一解碼器用於解碼由編碼器所產生的資訊，該編碼包含：一編碼器，用於響應資料而產生碼字資訊；

一重序單元，被耦合至編碼器，其中重序單元響應碼字資訊而產生一編碼資料流，且重序單元包含一運轉計數重序單元，用於將各碼字放置在與其對應資料之開始處，及一位元封包單元被耦合以接收來自運轉計數重序單元之碼字，以將可變長度碼字組合成許多固定長度插入字，並以解碼器所要求的順序來輸出許多固定長度插入字。

29. 如申請專利範圍第28項之編碼系統，其中重序單元進一步包含一記憶器以在重序期間來儲存碼字。

30. 如申請專利範圍第28項之編碼系統，其中編碼器進一步包含一上下文模型、一耦合至上下文模型之概率估計機構、及一耦合至概率估計機構之位元流產生器。

31. 如申請專利範圍第30項之編碼系統，其中運轉計數重序單元進一步包含：

第一記憶器以儲存碼字；

第一指示器及第二指示器，用於定位址第一記憶器作為一佇列，其中第一指示器指向指定為第一記憶器之輸出的第一進入，且第二指示器指向指定為第一記憶器中之下一個有用而未指定的儲存位置之第二進入。

32. 如申請專利範圍第31項之編碼系統，進一步

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

包含一指標記憶器，用於儲存對應第一記憶器中的位置之位址資訊，其目前被指定用於各索引之碼字的儲存。

3 3 . 如申請專利範圍第 3 2 項之編碼系統，其中各索引指出一概率等級。

3 4 . 如申請專利範圍第 3 2 項之編碼系統，其中各索引指出至少一上下文。

3 5 . 如申請專利範圍第 2 8 項之編碼系統，其中重序單元進一步包含一碼字記憶器佇列用於儲存碼字、頭指標用於指出欲輸出的至少一碼字、及尾指標用於指出將碼字插入於碼字記憶器佇列中的至少一記憶器位置。

3 6 . 如申請專利範圍第 3 5 項之編碼系統，其中碼字記憶器佇列中的各碼字進入包括一有效指示，且當頭指標指定碼字之位址，各碼字之有效指示指出其有效時，一碼字從碼字記憶器佇列被輸出。

3 7 . 如申請專利範圍第 2 8 項之編碼系統，其中位元封包單元包含位元封包邏輯，以從重序單元接收碼字，並將碼字合併於許多流的插入字中。

3 8 . 如申請專利範圍第 3 7 項之編碼系統，其中位元封包邏輯進一步包含許多累積器及許多暫存器，其中各許多暫存器被連接至各許多累積器及許多流的其中一個，各暫存器用於儲存其相關流的插入字，且各累積器指出用於儲存流之目前碼字的相關暫存器中的下一個位置，使得許多流的其中之一的各碼字被附加至與該許多流的其中之一有關的暫存器之上下文，根據相關的累積器中之值。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

39. 如申請專利範圍第38項之編碼系統，其中進一步包含一移相器被耦合至許多累積器，響應至少來自許多累積器之一信號而使碼字移位，其中碼字根據累積器值而被移位，以附加至許多暫存器中的上下文，而將至少兩個碼字之部份包封成各插入字。

40. 如申請專利範圍第37項之編碼系統，進一步包含一重序記憶器，用於將插入字以由解碼器所指定的順序來儲存。

41. 如申請專利範圍第40項之編碼系統，進一步包含對應許多插入資料流的許多指標，其中各許多指標指出各許多流之下一個插入字的重序記憶器中之位置。

42. 如申請專利範圍第28項之編碼系統，進一步包含許多運轉計數重序單元，其中各運轉計數重序單元是與其中一個編碼資料流相連，且進一步包含許多位元封包單元，用於產生各編碼資料流之插入字，及一解碼器用於從許多位元封包單元來選擇插入字作為編碼後的輸出。

43. 如申請專利範圍第28項之編碼系統，進一步包含許多運轉計數重序單元，其中各運轉計數重序單元是與其中一個編碼資料流相連，並產生許多碼字及與各許多碼字有關的時間標記，且進一步包含許多位元封包單元用於產生各編碼資料流之插入字，及邏輯用於根據各插入字中的碼字之時間標記而選擇各插入字。

44. 如申請專利範圍第43項之編碼系統，其中邏輯選擇包含具有最舊的時間標記之碼字的插入字。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

4 5 . 如申請專利範圍第 2 8 項之編碼系統，其中位元封包單元包含許多位元封包單元，被耦合以接收來自許多流的碼字，且進一步包含邏輯以根據下一個流而從各許多位元封包單元來選擇插入字而輸出。

4 6 . 如申請專利範圍第 2 8 項之編碼系統，其中重序單元進一步包含一有限記憶器。

4 7 . 一種用於處理資料之編碼系統，包含：

一索引產生器，用於根據資料而產生索引；及

一狀態表被耦合，而根據索引來提供一概率估計，其中此狀態表包括第一許多狀態及第二許多狀態，其中各狀態對應一碼，且當轉移於第一許多狀態中的狀態之間時，對應於第一許多狀態的不同碼之間的轉移，比當轉移於第二許多狀態中的狀態之間時，對應於第二許多狀態的不同碼之間的轉移更快。

4 8 . 如申請專利範圍第 4 7 項之編碼系統，其中對於一預定數目的索引只使用第一許多狀態。

4 9 . 如申請專利範圍第 4 7 項之編碼系統，其中對於啓始地索引狀態表之預定數目的索引，只使用第一許多狀態。

5 0 . 如申請專利範圍第 4 7 項之編碼系統，其中各第一許多狀態是與 R2 碼有關。

5 1 . 如申請專利範圍第 4 8 項之編碼系統，其中第一許多狀態包括至少轉移至第二許多狀態，使得狀態表從預定數目的索引之後的第一許多狀態，轉移至第二許多狀

六、申請專利範圍

態。

5 2 . 如申請專利範圍第 4 7 項之編碼系統，其中各第一許多狀態是與不同的碼有關。

5 3 . 如申請專利範圍第 4 7 項之編碼系統，其中響應一最低可能符號，狀態表從許多第一狀態的其中之一轉移至許多第二狀態的其中之一。

5 4 . 如申請專利範圍第 4 7 項之編碼系統，其中響應一最大可能符號，狀態機器增加狀態。

5 5 . 一種用於處理資料之編碼系統，包含：

一索引產生器，用於根據資料而產生索引；及

一狀態表被耦合，而根據索引來提供一概率估計，其中此狀態表包括許多狀態，其中各狀態對應一碼，狀態表中之每一個碼被重複預定數目的次數；

其中根據可以修改的加速項而發生狀態表之狀態間的轉移，使得於第一時間周期的狀態之間的轉移之第一速率，是不同於第二時間周期的轉移之第二速率。

5 6 . 如申請專利範圍第 5 5 項之編碼系統，其中對狀態表之更新包含：藉由增量或減量加速項而修改 PEM 狀態。

5 7 . 如申請專利範圍第 5 6 項之編碼系統，其中當加速項包含一預定數目時，沒有發生配合的加速。

5 8 . 如申請專利範圍第 5 6 項之編碼系統，其中根據連續碼字之數目而更新加速項。

5 9 . 如申請專利範圍第 5 8 項之編碼系統，其中連

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

續碼字包含上下文中之連續碼字。

6 0 . 如申請專利範圍第 5 8 項之編碼系統，其中連續碼字包含概率等級中之連續碼字。

6 1 . 如申請專利範圍第 5 6 項之編碼系統，其中根據改變碼字之數目而使加速項更新。

6 2 . 一種用於解碼許多碼字的資料流之一致性解碼器，包含：

許多位元流產生器，用於接收資料流；及

一狀態表，被耦合至許多位元流產生器，以提供概率估計至許多位元流產生器，其中該許多位元流產生器響應概率估計而產生資料流中之各碼字的解碼結果，此概率估計是使用多值的 n 之 $R_n(k)$ 碼，且狀態表包括許多第一狀態及許多第二狀態，其中當在許多第一狀態中轉移時，第一許多狀態中的不同碼之間的轉移，比當在許多第二狀態中轉移時碼之間的轉移更快。

6 3 . 如申請專利範圍第 6 2 項之一致性解碼器，其中許多第一狀態各包含一 $R_2(k)$ 碼。

6 4 . 如申請專利範圍第 6 2 項之一致性解碼器，其中許多第一狀態只使用於啓始期間。

6 5 . 一種用於解碼許多碼字的資料流之一致性解碼器，包含：

許多位元流產生器，用於接收資料流；及

一狀態表，被耦合而根據索引來提供概率估計，其中狀態表包括許多狀態，各狀態對應一碼，且狀態表中之每

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

一個碼被重複一預定數目的次數；

其中狀態表的狀態之間的轉移是根據可修改的加速項而發生，使得於第一時間周期的狀態之間的轉移之第一速率，是不同於第二時間周期的轉移之第二速率。

6 6 . 如申請專利範圍第 6 2 項之一致性解碼器，其中狀態表中的每一個碼被重複固定數目的次數。

6 7 . 如申請專利範圍第 6 6 項之一致性解碼器，其中狀態表之更新包含以一加速項來修改 PEM 狀態。

6 8 . 如申請專利範圍第 6 7 項之一致性解碼器，其中當加速項包含一預定數目時，沒有發生配合的加速。

6 9 . 如申請專利範圍第 6 7 項之一致性解碼器，其中根據連續碼字之數目而更新加速項。

7 0 . 如申請專利範圍第 6 7 項之一致性解碼器，其中根據改變碼字的數目而更新加速項。

7 1 . 一種用於解碼許多插入字之解碼器，該解碼器包含：

一移相器，用於接收資料流並輸出適當地對齊的編碼資料；

一運轉長度解碼器，被耦合至移相器以接收適當地對齊的編碼資料作為碼字以決定碼字型式；

一概率估計機器被耦合至運轉長度解碼器，以決定運轉長度解碼器之碼，使得運轉長度解碼器產生一運轉長度，及指示是否 LPS 響應各碼字而發生；

其中該移相器包含一可變長度移位機構，以移位來自

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

資料流之碼字；且

許多暫存器被耦合以響應移位機構而接收來自流的碼字，使得對齊的編碼資料被輸出而形成一系列的碼字。

7 2 . 如申請專利範圍第 7 1 項之解碼器，其中一部份的暫存器被耦合以接收來自資料流或來自其它的許多暫存器之資料。

7 3 . 如申請專利範圍第 7 1 項之解碼器，其中可變長度移位機構包括一桶形移相器，以將來自流之資料移位進入許多暫存器。

7 4 . 如申請專利範圍第 7 1 項之解碼器，其中移相器包含一具有許多暫存器之 FIFO，其中各暫存器從插入的編碼資料接收資料作為輸入，且至少許多暫存器的其中之一被耦合以接收來自該其它許多暫存器的碼字。

7 5 . 如申請專利範圍第 7 1 項之解碼器，其中移相器包含：

許多第一暫存器被耦合以接收碼字資料，其中各許多第一暫存器被耦合至不同的許多流；

一多工器被耦合以接收來自各許多第一暫存器之碼字資料作為輸入，並以一次而輸出來自許多流的其中之一一的碼字；

一桶形移相器，被耦合至多工器之輸出，以將來自多工器之碼字資料移位而輸出一對齊的編碼資料；

邏輯，被耦合至桶形移相器，以指出許多位元而移位碼字；及

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

一 FIFO，具有許多暫存器其被耦合以接收來自該多工器 1605 之碼字，其中 FIFO 包括許多暫存器，其中各許多暫存器接收資料作為來自插入的編碼資料之輸入，且至少許多暫存器的其中之一被耦合以接收來自其它該許多暫存器的碼字。

7 6 . 一種用於解碼資料之解碼系統，該解碼系統包含：

一 FIFO 結構，被耦合以接收資料；

一上下文模型，用於提供上下文；

一記憶器，被耦合至上下文模型以儲存狀態資訊，其中記憶器響應由上下文模型所提供的各上下文而提供狀態資訊；

許多解碼器，被耦合以接收來自 FIFO 結構的編碼資料，及記憶器用於使用來自記憶器之狀態資訊而解碼由 FIFO 結構所供給的碼字，其中許多解碼器提供由許多碼所產生的運轉計數。

7 7 . 如申請專利範圍第 7 6 項之解碼系統，其中 FIFO 結構供給編碼資料至與上下文及概率等級獨立的許多解碼器。

7 8 . 如申請專利範圍第 7 6 項之解碼系統，其中解碼器包含一運轉計數記憶器用於儲存運轉計數，且根據概率等級而可存取運轉計數記憶器。

7 9 . 如申請專利範圍第 7 6 項之解碼系統，其中 FIFO 結構提供兩個解碼器之資料。

六、申請專利範圍

8 0 . 如申請專利範圍第 7 6 項之解碼系統，其中 FIFO 結構包含許多輸出，每一個解碼器各一個輸出。

8 1 . 如申請專利範圍第 8 0 項之解碼系統，其中 FIFO 結構包含一對的多工器，及控制邏輯用於選擇此對的多工器以確保有一個碼字被提供給各解碼器。

8 2 . 如申請專利範圍第 8 1 項之解碼系統，其中根據從許多解碼器的其中之一所接收的要求，而由控制邏輯來選擇此對的多工器。

8 3 . 一種用於編碼輸入資料之編碼系統，包含：

一編碼單元，被耦合以接收輸入資料而產生許多流的形式之編碼資料，其中編碼資料根據一組臨界而被指定至許多流的其中之一；

一固定尺寸記憶器，被耦合至編碼單元以儲存許多流的編碼資料，其中如果固定尺寸記憶器溢流的話，則較不重要的編碼資料被拋棄。

8 4 . 如申請專利範圍第 8 3 項之編碼系統，其中記憶器包含許多儲存區域，且各許多儲存區域中所儲存的編碼資料包含不同重要性位準的編碼資料。

8 5 . 如申請專利範圍第 8 4 項之編碼系統，其中一重要性位準的編碼資料被儲存在一記憶器的儲存區域中，此記憶器是以另一重要性位準來儲存編碼資料。

8 6 . 如申請專利範圍第 8 5 項之編碼系統，其中該重要性位準的編碼資料重寫在記憶器的該儲存區域中該另一重要性位準的編碼資料。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

87. 一種用於啓始系統中的許多上下文而同時編碼資料之方法，該方法包含：

啓始許多上下文，其中各上下文是根據一計數器值而被存取；

得到目前上下文的PEM狀態，其中此得到步驟包含以下步驟：

存取指出PEM狀態的各上下文之儲存指示；

藉著比較目前的上下文數目與計數器值，而決定所存取的記憶器位置是否對於目前操作為有效，使得如果計數器值指出此位置已經被啓始化，則資料被決定為有效；

當所存取的記憶器位置之資料不是有效時，則使用上下文之啓始PEM狀態而忽略上下文之目前的PEM狀態；及

如果資料是有效的，則使用目前指定的PEM狀態。

88. 如申請專利範圍第87項之方法，如果PEM狀態改變時，則進一步包含重寫一新的PEM狀態之步驟。

89. 一種用於解碼輸入資料之解碼器，包含：

一上下文模型，以提供上下文筐；

一記憶器，被耦合至上下文模型以根據上下文筐而提供一概率狀態；

一邏輯，被耦合至記憶器以根據概率狀態而產生許多概率等級；

一解碼器，被耦合至邏輯以根據概率等級而產生一致能信號；

許多位元產生器，被耦合至解碼器且被耦合以接收編

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

碼資料，其中各許多位元產生器專屬至少一種不同的碼，其中解碼器根據概率等級而致能許多位元產生器的其中之一，使得該許多位元流產生器的其中之一解碼此編碼資料。

9 0 . 如申請專利範圍第 8 9 項之解碼器，其中至少許多位元產生器的其中之一解碼使用 R-碼之資料，且至少許多位元產生器的其中之一解碼使用非 R-碼之資料。

9 1 . 如申請專利範圍第 8 9 項之解碼器，其中用於短的運轉長度之位元產生器操作為 R-碼解碼器。

9 2 . 如申請專利範圍第 8 9 項之解碼器，其中用於長的運轉長度之位元產生器包含一短的運轉單元及一長的運轉單元，其中短的運轉單元處理第一預定長度的碼，且長的運轉單元處理其餘的位元，並決定那些其餘的位元需被輸出。

9 3 . 一種用於解碼輸入資料之方法，包含以下步驟：

提供一上下文筐；

使用上下文筐而存取一記憶器以得到一概率狀態；

致能許多位元產生器的其中之一，其中各許多位元產生器專屬於至少一種不同的碼，使得只有使用該不同的碼而用於解碼，使得許多位元流產生器的其中之一解碼此編碼資料。

9 4 . 一種用於編碼輸入資料之編碼器系統，包含：

一上下文模型，以提供一上下文筐；

六、申請專利範圍

一記憶器，被耦合至上下文模型以根據此上下文筐而提供一概率狀態；

一邏輯，被耦合至記憶器以根據概率狀態而產生一概率等級；

一編碼器，被耦合至邏輯以根據概率等級而產生一致能信號；

許多位元產生器，被耦合至解碼器且被耦合以接收輸入資料，其中各許多位元產生器專屬於至少一種不同的碼，其中編碼器根據概率等級而致能許多位元產生器的其中之一，使得許多位元流產生器的其中之一編碼此輸入資料。

9 5 . 如申請專利範圍第 9 4 項之編碼器系統，其中至少許多位元產生器的其中之一編碼使用 R-碼的資料，且至少許多位元產生器的其中之一編碼使用非 R-碼的資料。

9 6 . 如申請專利範圍第 9 4 項之編碼器系統，其中用於短的運轉長度之位元產生器操作為 R-碼編碼器。

9 7 . 如申請專利範圍第 9 4 項之編碼器系統，其中用於長的運轉長度之位元產生器包含一短的運轉單元及一長的運轉單元，其中短的運轉單元處理第一預定長度的碼，且長的運轉單元處理其餘的位元，並決定那些其餘的位元需被輸出。

9 8 . 一種用於編碼輸入資料之方法，包含以下步驟：

提供一上下文筐；

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

使用上下文筐來存取一記憶器而得到一概率狀態；

根據概率狀態而產生一概率等級；

致能許多位元產生器的其中之一，其中各許多位元產生器專屬於至少一種不同的碼，使得只使用該不同的碼來編碼，使得許多位元流產生器的其中之一編碼此輸入資料。

9 9 . 一種用於解碼包含許多碼字的編碼資料之方法，該方法包含以下步驟：

將計數值載入與各運轉計數器相關的計數器，其中當開始一新的運轉時，計數值對應於編碼期間所使用的碼字記憶器之尺寸，當取得用於各運轉計數器之新的碼字時，該計數值被載入；

每次取得碼字時則使計數值增量，及

當計數器減量至零時，則清除與該新的碼字相關的位元產生器狀態。

1 0 0 . 如申請專利範圍第 9 9 項之方法，其中各運轉計數器對應一 PEM 狀態。

1 0 1 . 如申請專利範圍第 9 9 項之方法，其中各運轉計數器對應一上下文筐。

1 0 2 . 一種用於解碼包含許多碼字的編碼資料之方法，該方法包含以下步驟：

每次要求一碼字時，使計數器值增量，其中計數器值包含一目前的時間指示；

當第一個碼字開始時，儲存計數器值作為一儲存時間

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

指示：

比較儲存時間指示加上編碼器記憶器之尺寸與目前的時間指示：

其中第一個碼字之位元產生器狀態被清除，且當目前的時間指示大於儲存時間指示加上編碼器記憶器之尺寸時，則要求第二個碼字。

1 0 3 . 如申請專利範圍第 1 0 2 項之方法，其中儲存時間指示包含一時間標記。

1 0 4 . 如申請專利範圍第 1 0 2 項之方法，進一步包含重新使用隨後的碼字之儲存時間指示。

1 0 5 . 一種用於解碼包含許多碼字的編碼資料之方法，該方法包含以下步驟：

儲存對應一碼字之索引，其中索引儲存於一佇列中，且當要求碼字時則儲存此索引；

標記佇列中的索引之進入成為有效；

儲存此進入中的碼字，且如果碼字完成則標記此進入為有效；

輸出來自欲被解碼之佇列的資料，其中如果此佇列進入是有效的則碼字被輸出；及

輸出來自佇列進入之資料，且當從佇列輸出資料時如果佇列進入被標記為無效，則指出資料對於解碼器是無效的，其中響應從標記為無效的佇列接收資料，而以解碼器來清除位元產生器狀態資訊。

1 0 6 . 一種用於解碼一編碼資料之解碼器，該解碼

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

器包含：

一 上下文模組化機構用於提供上下文，其中此上下文模組化機構包含許多積體電路；

一 記憶器，被耦合至上下文模型以儲存狀態資訊，其中此記憶器響應由上下文模型所提供的各上下文而提供狀態資訊；及

許多解碼器被耦合至記憶器，用於使用來自記憶器之狀態資訊而解碼碼字，其中許多解碼器使用許多R-碼之碼字，其中許多R-碼包括至少最大可能符號的其中一個非最大長度運轉，其後跟隨著一最小可能符號。

1 0 7 . 如申請專利範圍第 1 0 6 項之解碼器，其中非最大長度運轉計數具有一獨特的可解碼字首。

1 0 8 . 一種用於解碼具有許多碼字之碼流的系統，該系統包含：

一 上下文模組化機構用於提供上下文，其中上下文模組化機構包含許多積體電路；

一 記憶器，被耦合至上下文模型以儲存狀態資訊，其中此記憶器響應由上下文模型所提供的各上下文而提供狀態資訊；及

許多解碼器被耦合至記憶器，用於使用來自記憶器之狀態資訊而解碼碼字。

1 0 9 . 如申請專利範圍第 1 0 8 項之系統，其中上下文模組化機構包含至少一上下文模型提供來自許多積體電路之上下文，及至少一上下文模型提供來自許多積體電

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

路的第二個之上下文。

1 1 0 . 如申請專利範圍第 1 0 9 項之系統，其中在該許多積體電路的其中之一上的該至少一上下文模型，包含零順序上下文模型。

1 1 1 . 如申請專利範圍第 1 0 8 項之系統，其中來自許多積體電路之上下文被直接提供至記憶器。

1 1 2 . 如申請專利範圍第 1 0 8 項之系統，其中第一部份的第一上下文是由一積體電路來提供，而第二部份的第一上下文是由第二積體電路來提供。

1 1 3 . 如申請專利範圍第 1 7 項之編碼系統，其中位元產生器之位元產生器狀態被更新，且在更新後的狀態被寫入記憶器之前，重新使用此位元產生器。

1 1 4 . 如申請專利範圍第 1 1 3 項之編碼系統，其中於讀取-修改-寫入循環的修改階段期間重新使用位元產生器。

1 1 5 . 如申請專利範圍第 1 7 項之編碼系統，其中當在將一更新後的狀態寫入記憶器之前而重新使用時，位元產生器產生一非最小長度運轉計數。

1 1 6 . 如申請專利範圍第 1 7 項之編碼系統，其中位元產生器編碼使用 R-碼定義的資料，使得各運轉長度之後跟隨著至少一未編碼位元，使得相同運轉長度的兩個碼字不會被解碼於一系列中。

1 1 7 . 一種用於解碼具有許多碼字之碼流的系統，該系統包含：

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

一 上下文模組化機構用於提供上下文；

一 記憶器，被耦合至上下文模型以儲存狀態資訊，其中此記憶器響應由上下文模型所提供的各上下文而提供狀態資訊；及

許多解碼器被耦合至記憶器，用於使用來自記憶器之狀態資訊而解碼碼字，其中至少許多解碼器的其中之一包含一延遲容許解碼器。

1 1 8 . 如申請專利範圍第 1 1 7 項之系統，其中至少許多解碼器的其中之一根據在延遲之後可用的解碼資料而執行可變長度移位。

1 1 9 . 如申請專利範圍第 1 1 7 項之系統，其中各許多解碼器接收可變長度資料作為輸入。

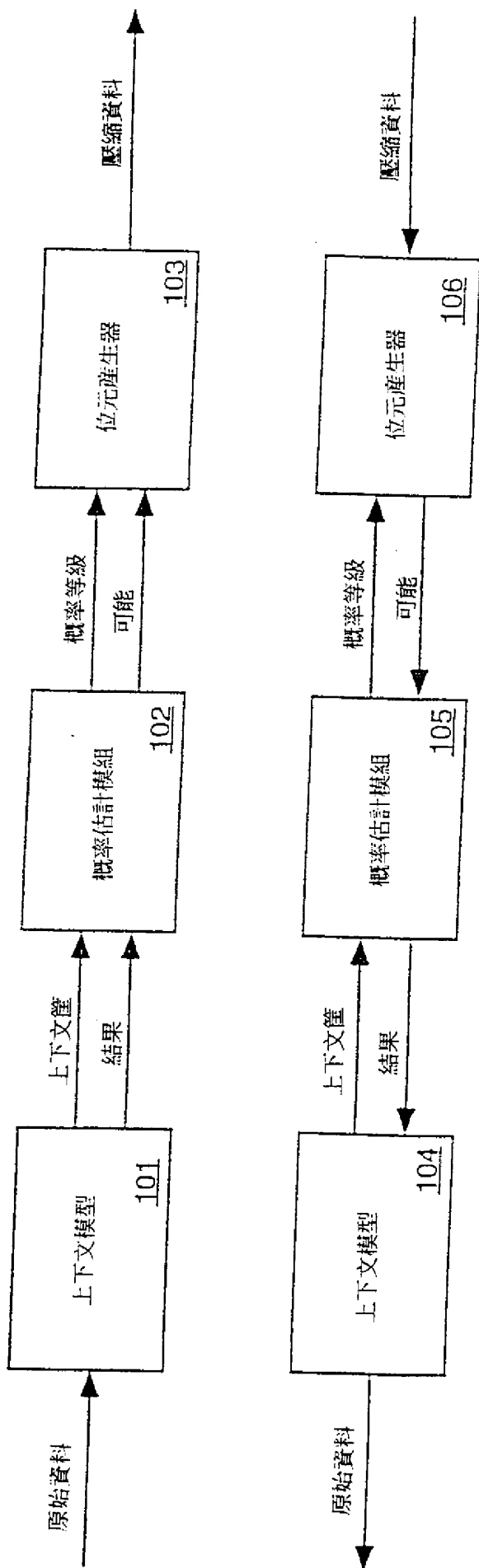
1 2 0 . 如申請專利範圍第 1 1 9 項之系統，其中許多解碼器平行地解碼可變長度輸入資料。

1 2 1 . 如申請專利範圍第 1 1 7 項之系統，其中許多解碼器的輸出被分割成固定長度的插入字。

(請先閱讀背面之注意事項再填寫本頁)

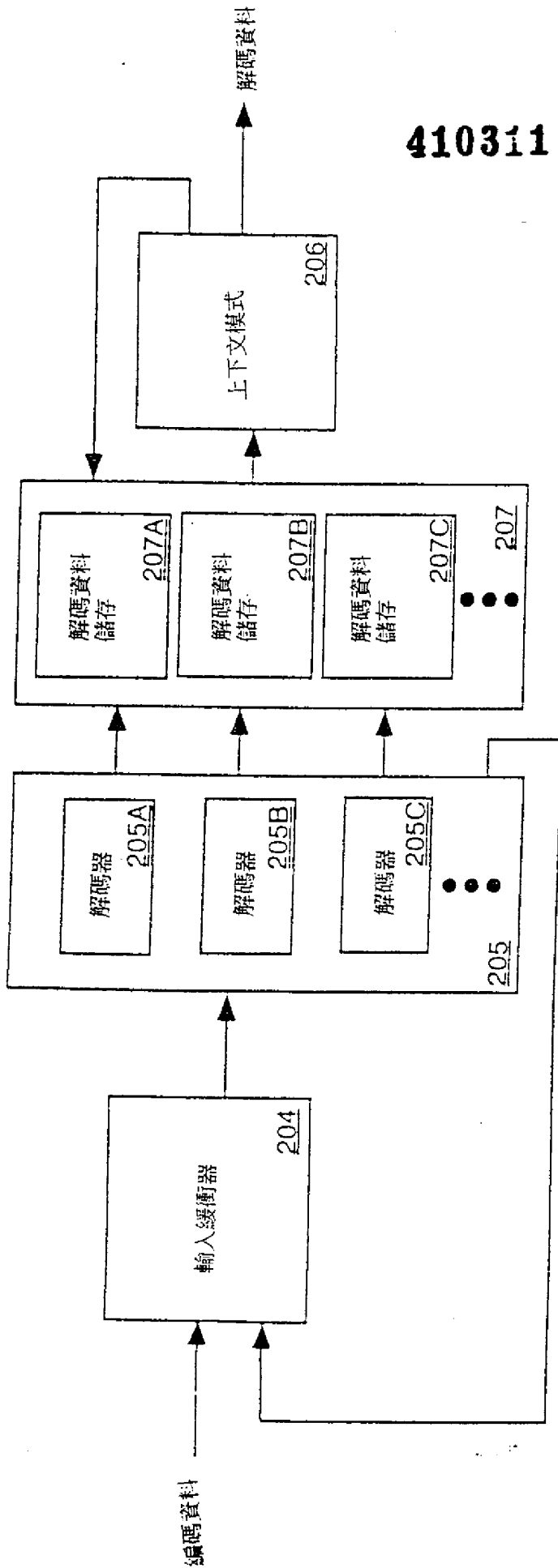
裝

訂

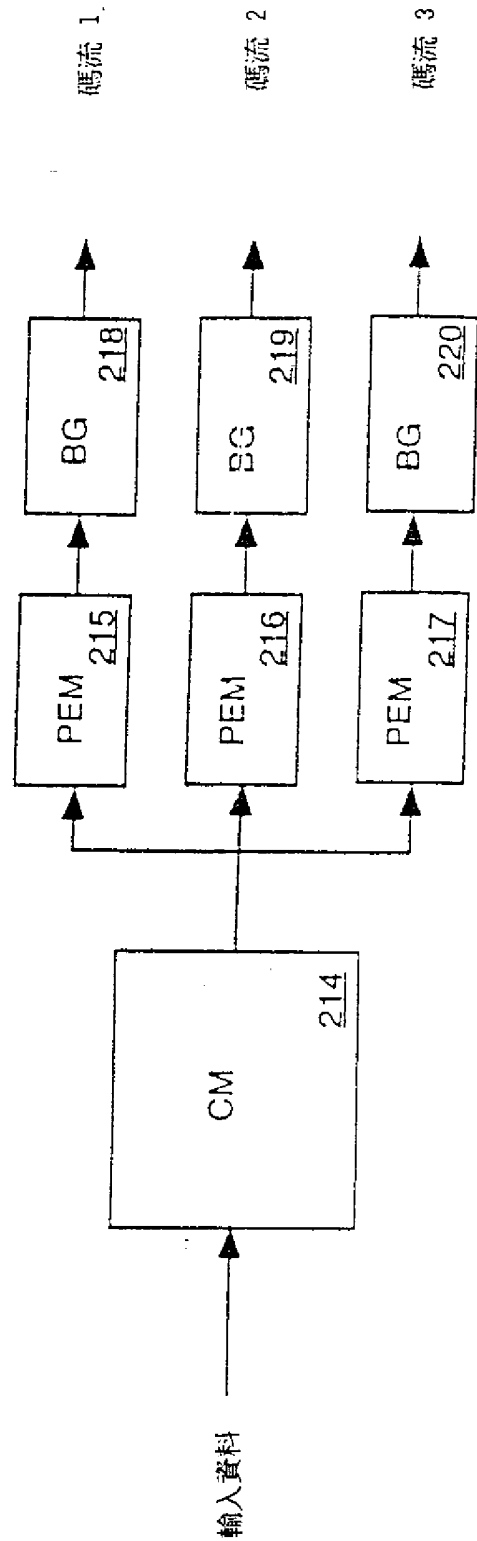


第 1 圖

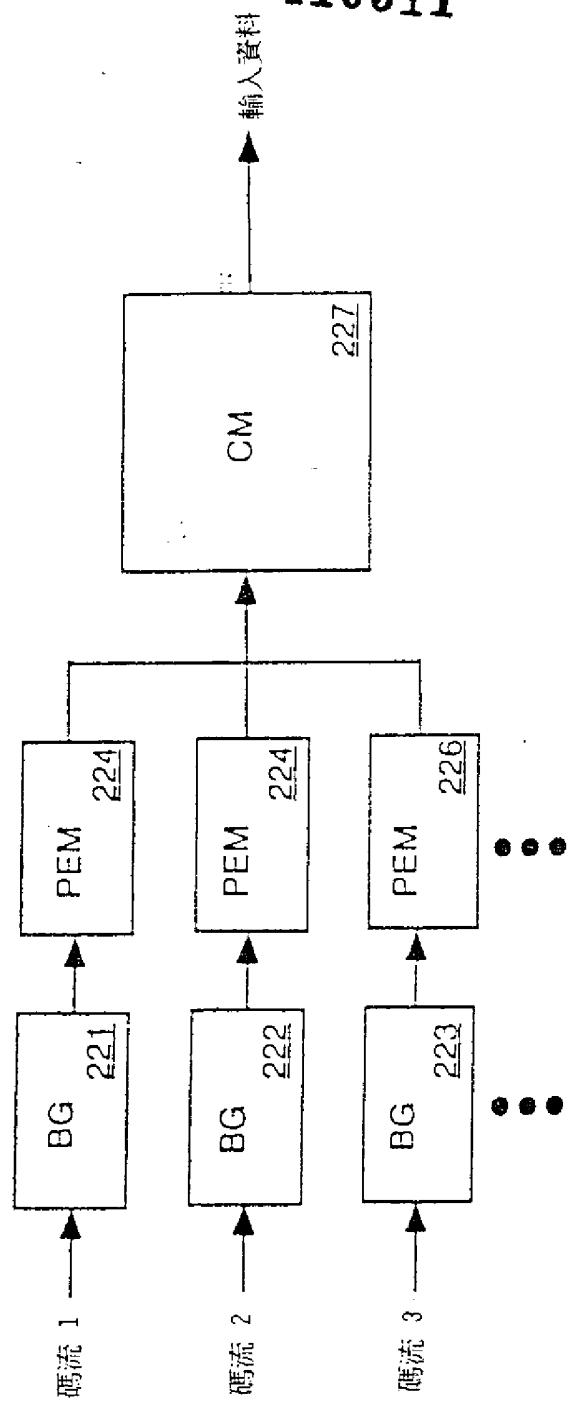
410311



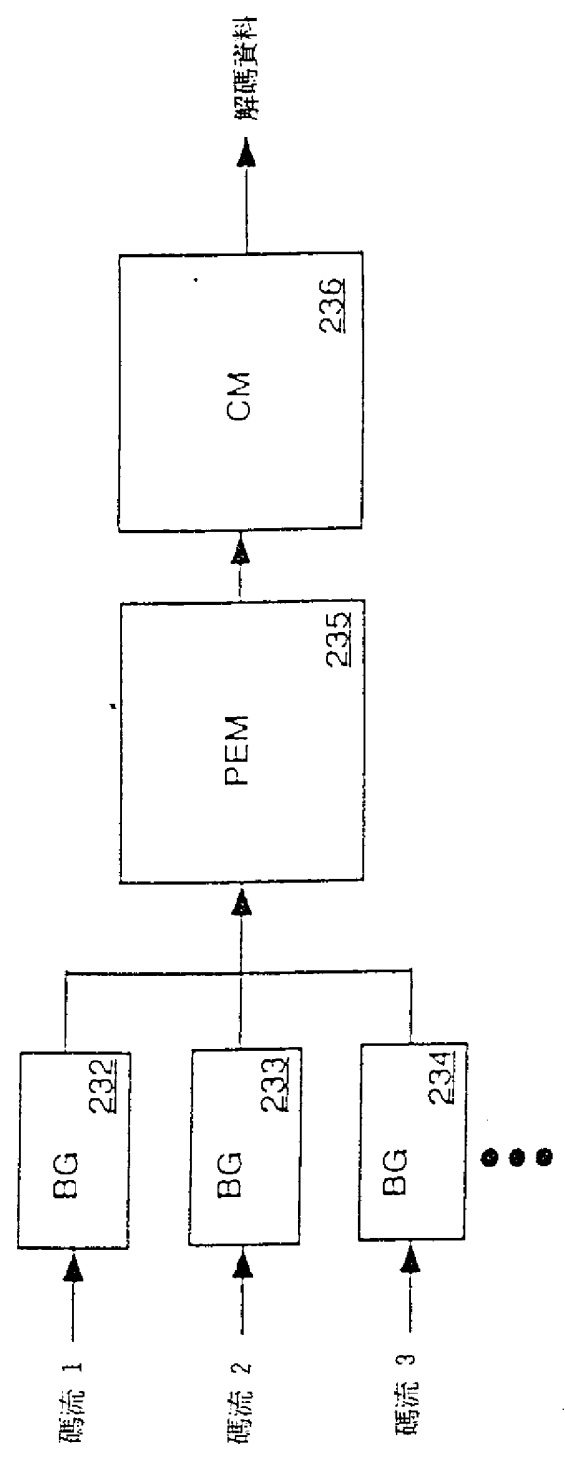
第2A圖



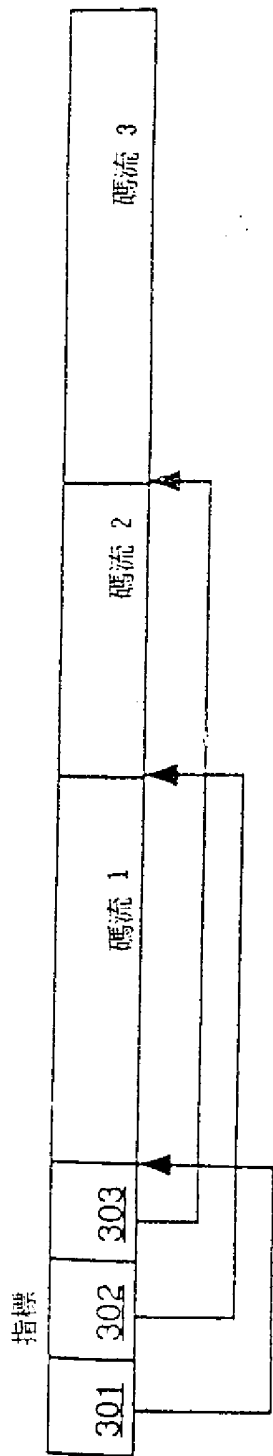
第2B圖



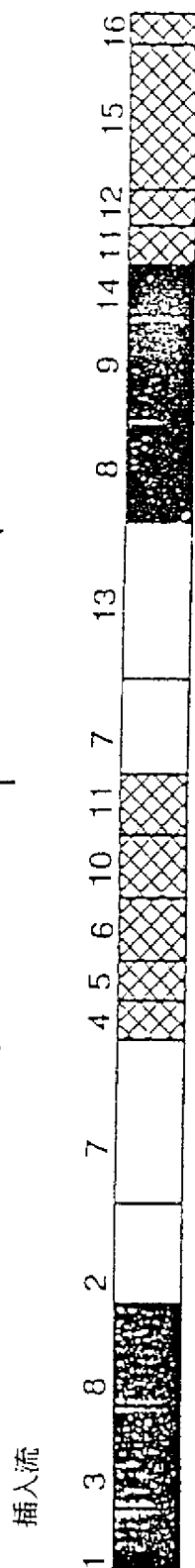
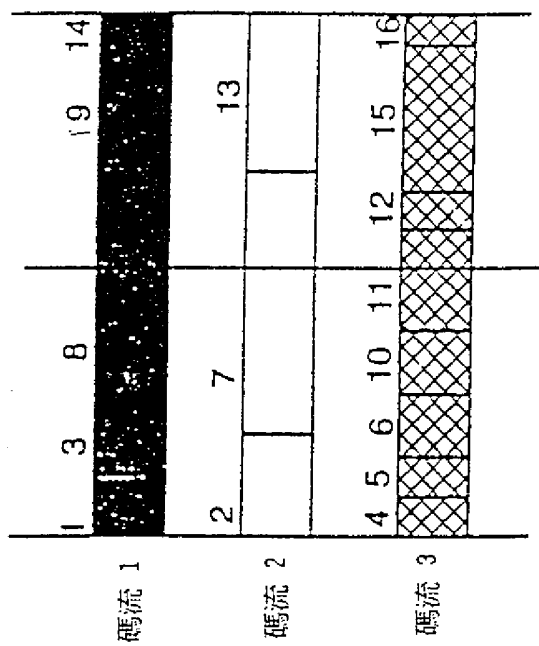
第2C圖



第2D圖



第 3 圖

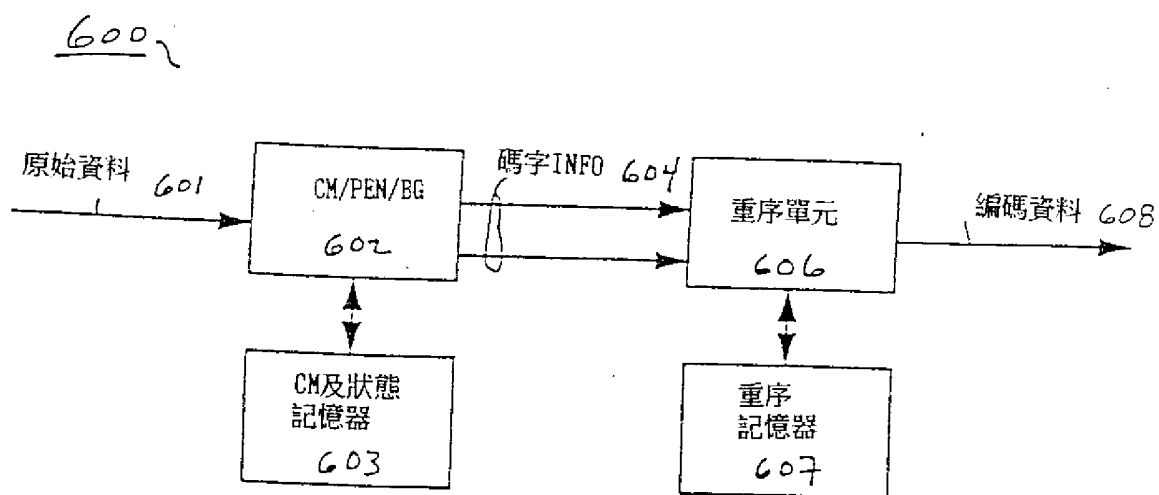


第 4 圖

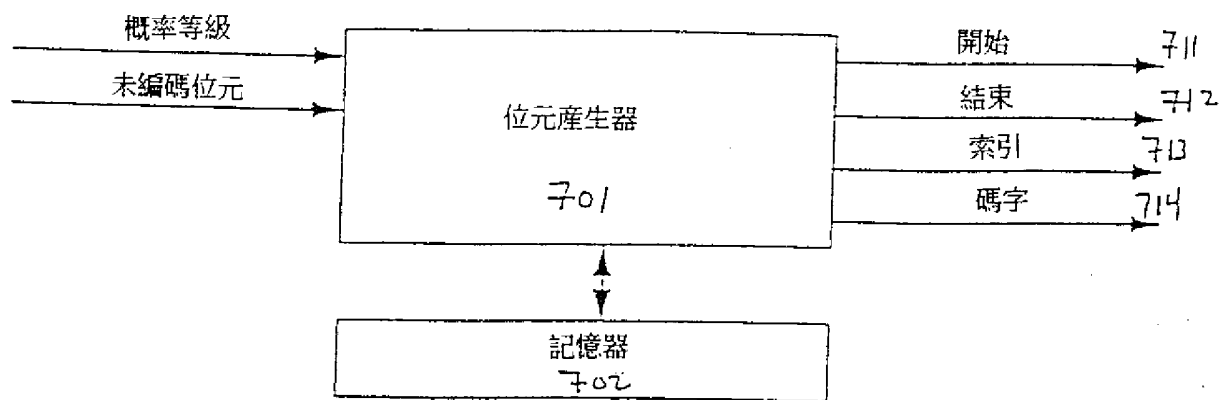
狀態	碼
-35	R2(12)
-34	R3(11)
-33	R2(11)
-32	R3(10)
-31	R2(10)
-30	R3(9)
-29	R2(9)
-28	R3(8)
-27	R2(8)
-26	R3(7)
-25	R2(7)
-24	R3(6)
-23	R2(6)
-22	R3(5)
-21	R2(5)
-20	R3(4)
-19	R2(4)
-18	R3(3)
-17	R2(3)
-16	R3(2)
-15	R2(2)
-14	R3(1)
-13	R3(1)
-12	R3(1)

狀態	碼
-11	R2(1)
-10	R2(1)
-9	R2(1)
-8	R2(1)
-7	R2(1)
-6	R2(1)
-5	R2(0)
-4	R2(0)
-3	R2(0)
-2	R2(0)
-1	R2(0)
-0	R2(0)
0	R2(0)
1	R2(0)
2	R2(0)
3	R2(0)
4	R2(0)
5	R2(0)
6	R2(1)
7	R2(1)
8	R2(1)
9	R2(1)
10	R2(1)
11	R2(1)

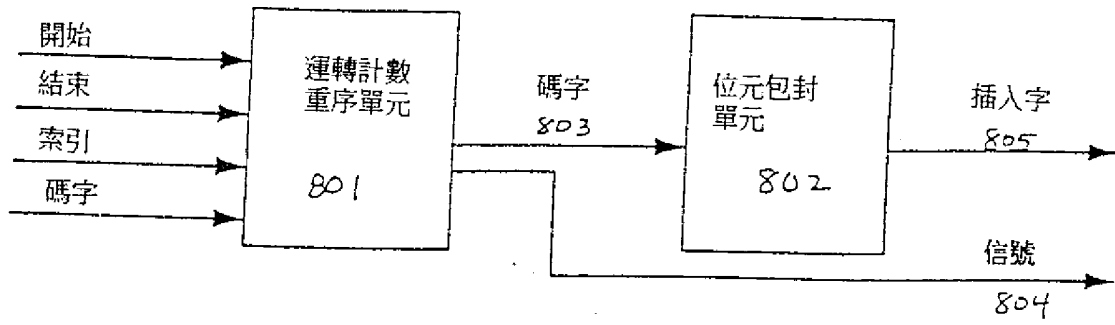
狀態	碼
12	R3(1)
13	R2(1)
14	R3(1)
15	R2(2)
16	R3(2)
17	R2(3)
18	R3(3)
19	R2(4)
20	R3(4)
21	R2(5)
22	R3(5)
23	R2(6)
24	R3(6)
25	R2(7)
26	R3(7)
27	R2(8)
28	R3(8)
29	R2(9)
30	R3(9)
31	R2(10)
32	R3(10)
33	R2(11)
34	R3(11)
35	R2(12)



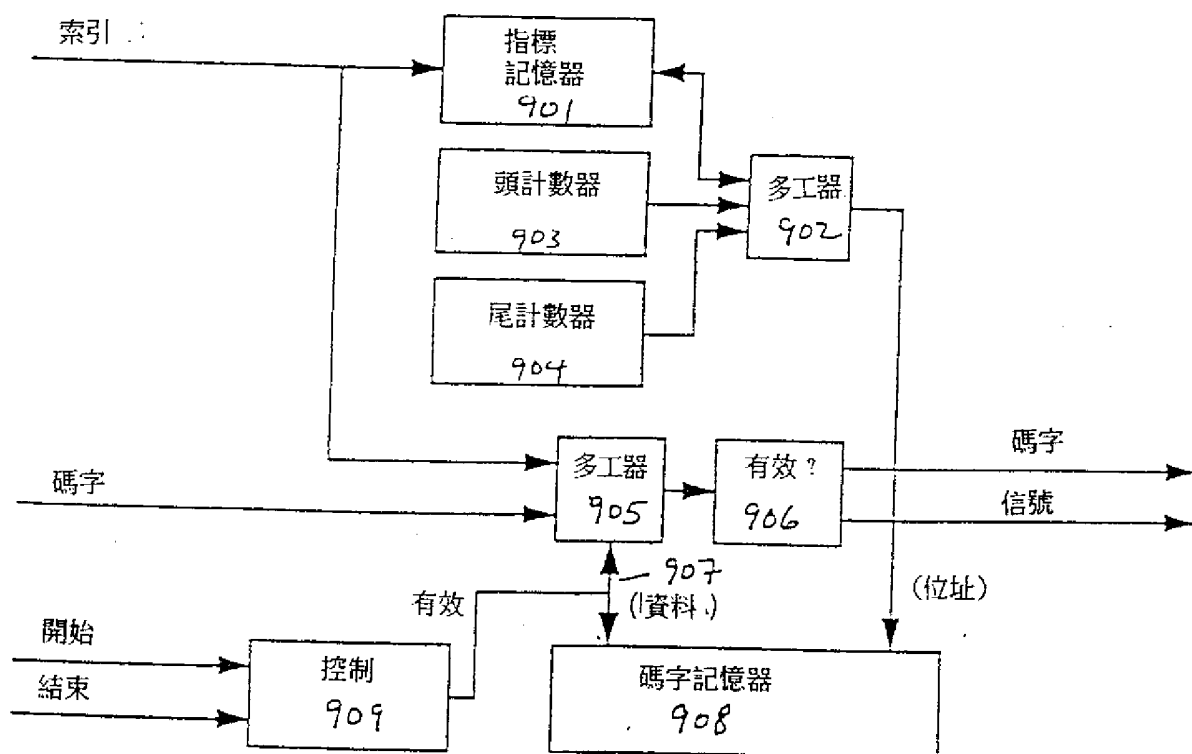
第 6 圖



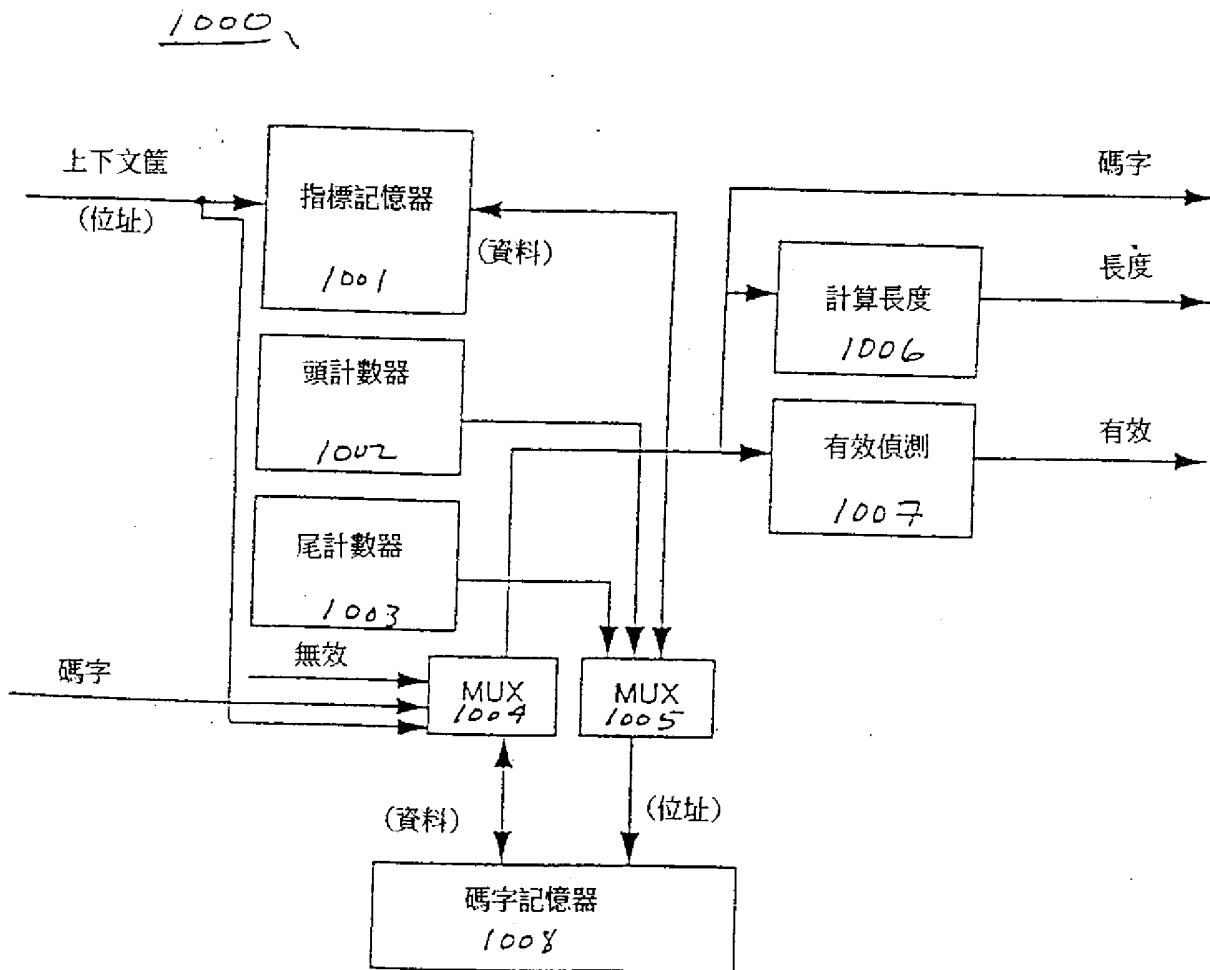
第 7 圖



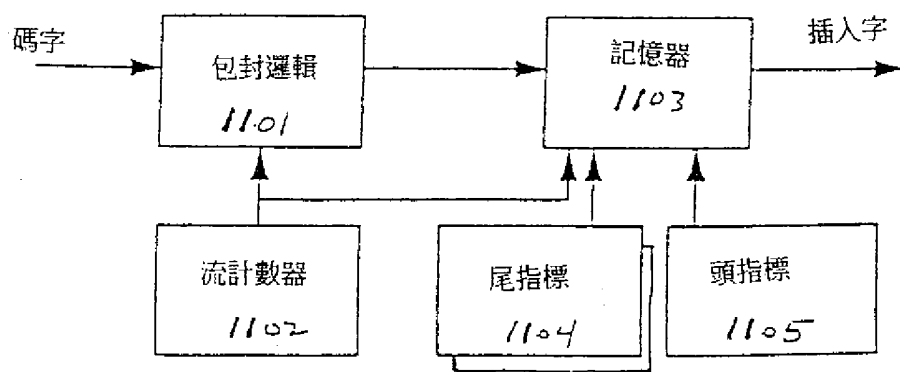
第 8 圖



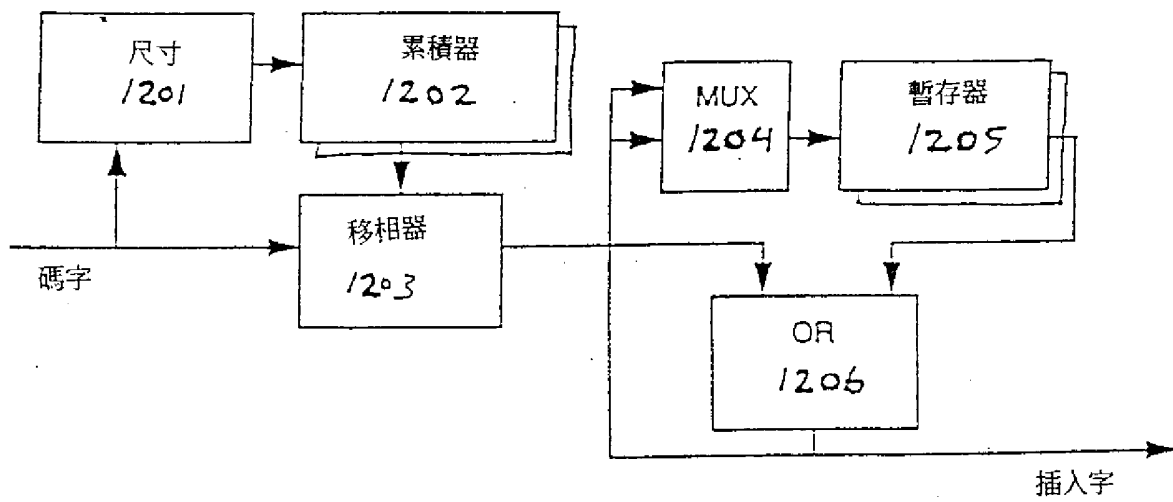
第 9 圖



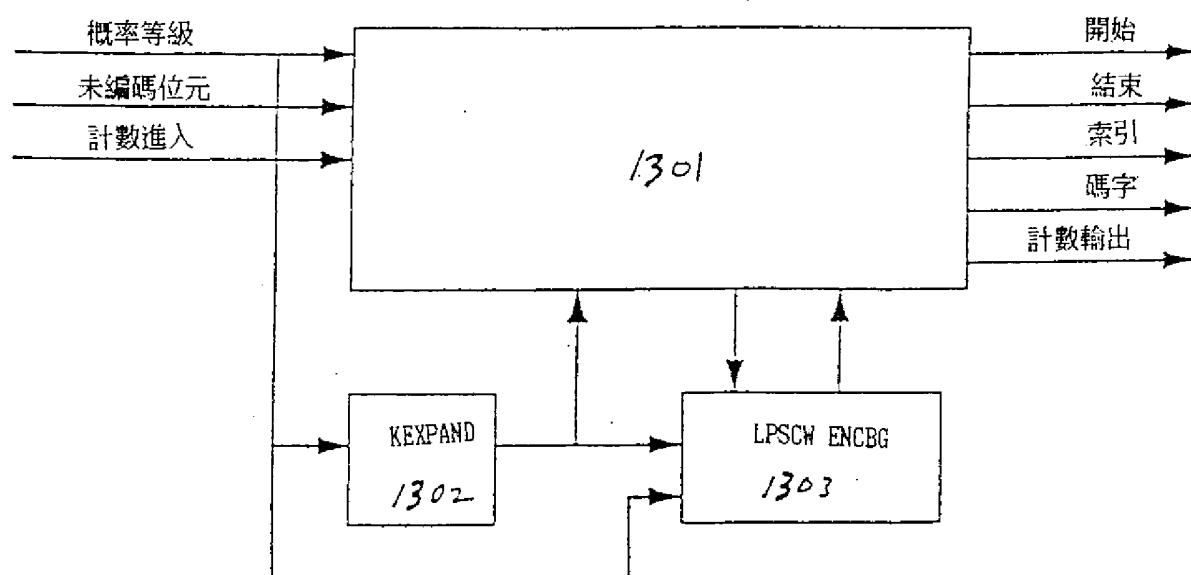
第10圖



第11圖

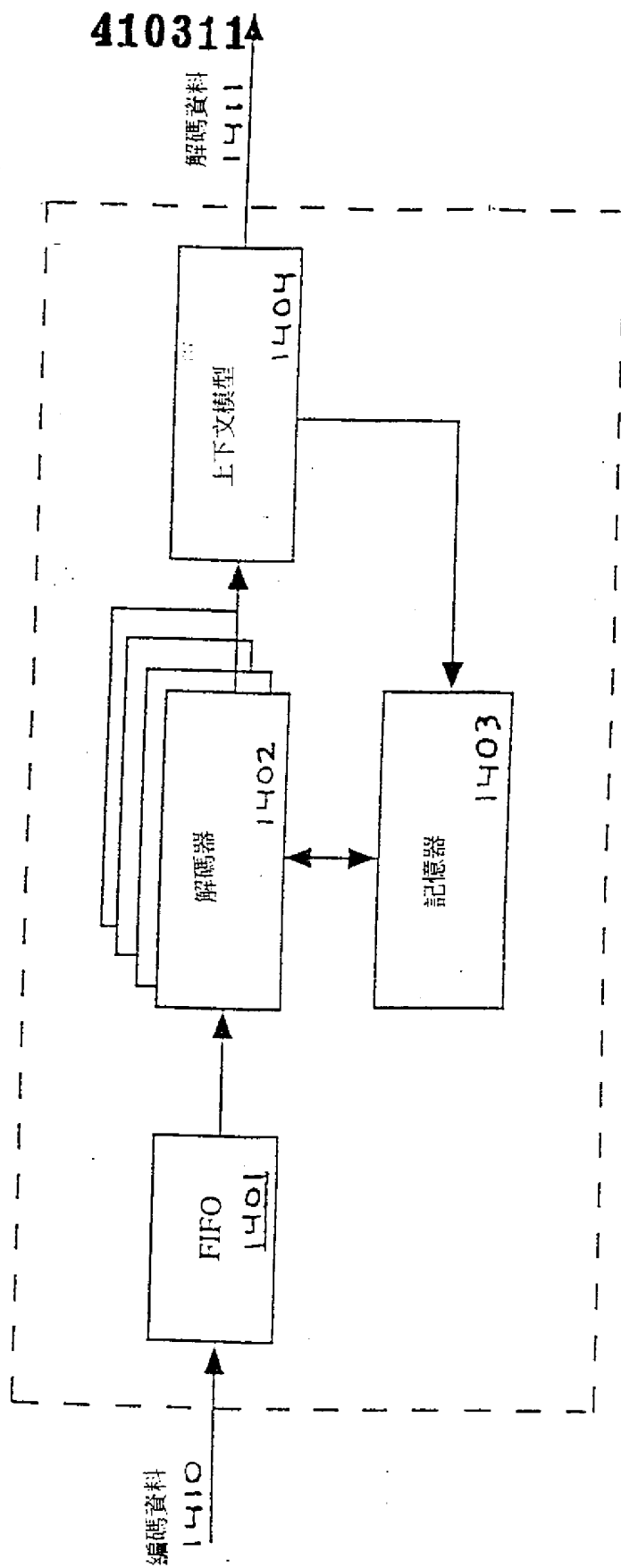


第12圖

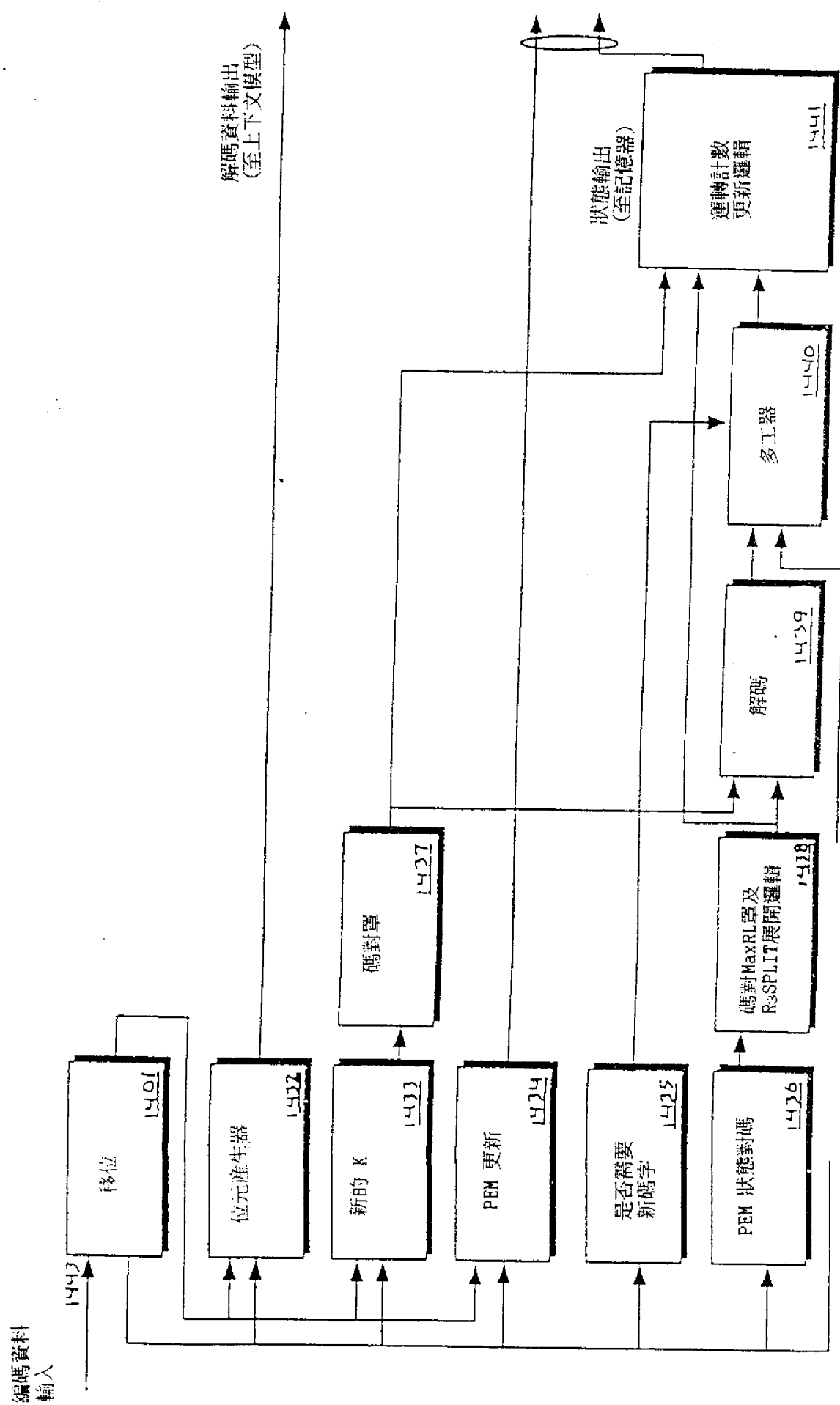


第13圖

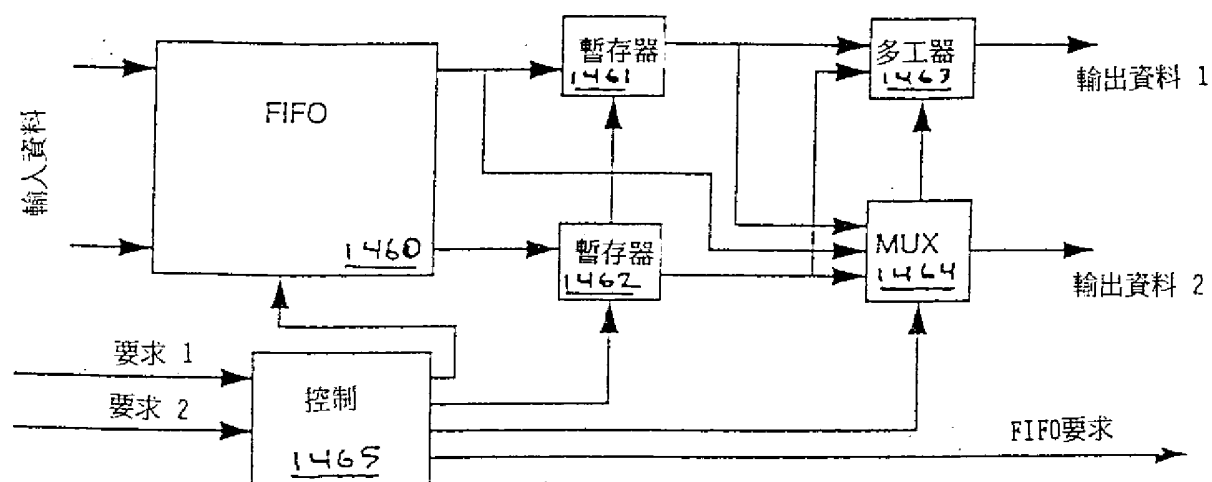
1400



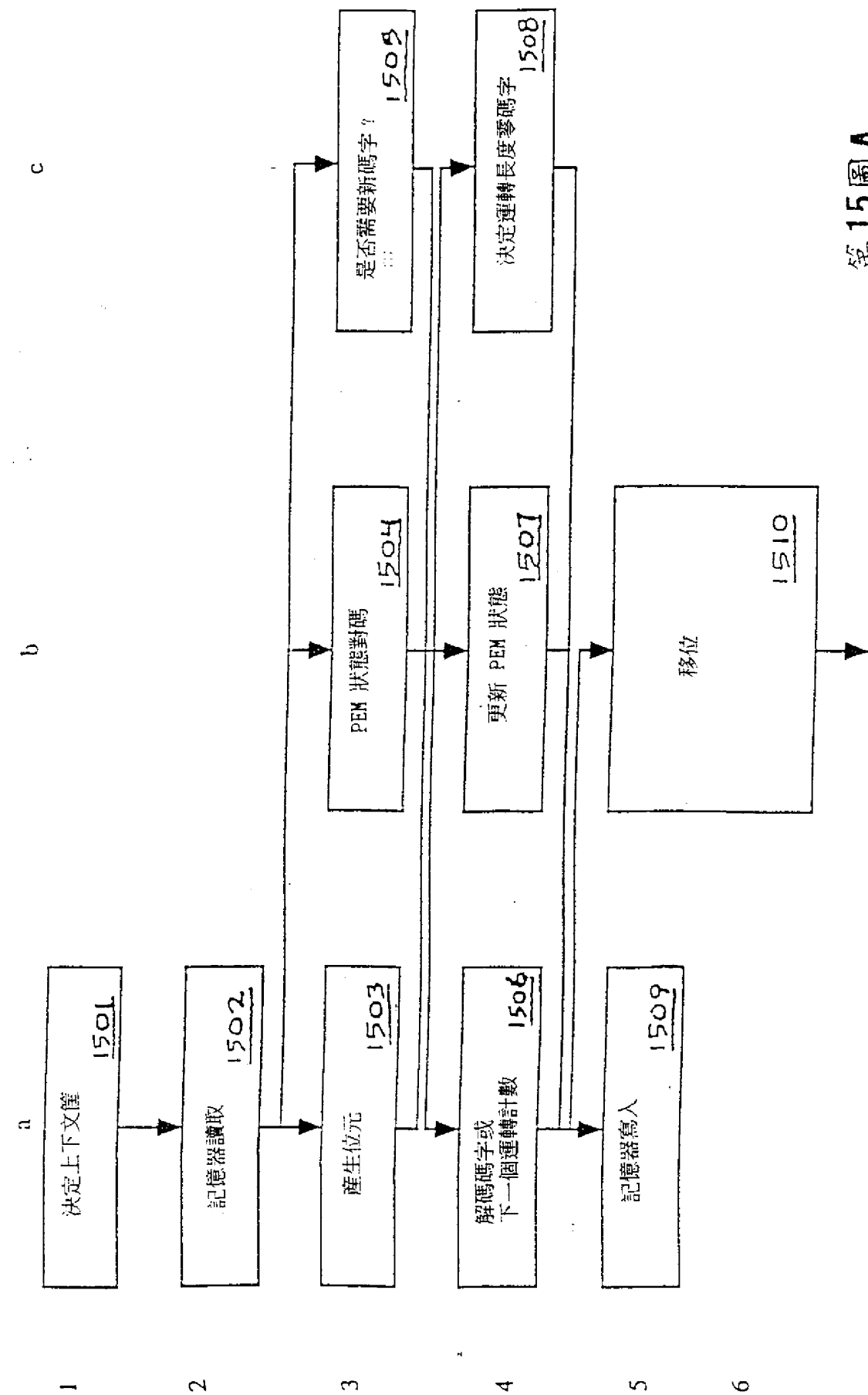
第14圖A



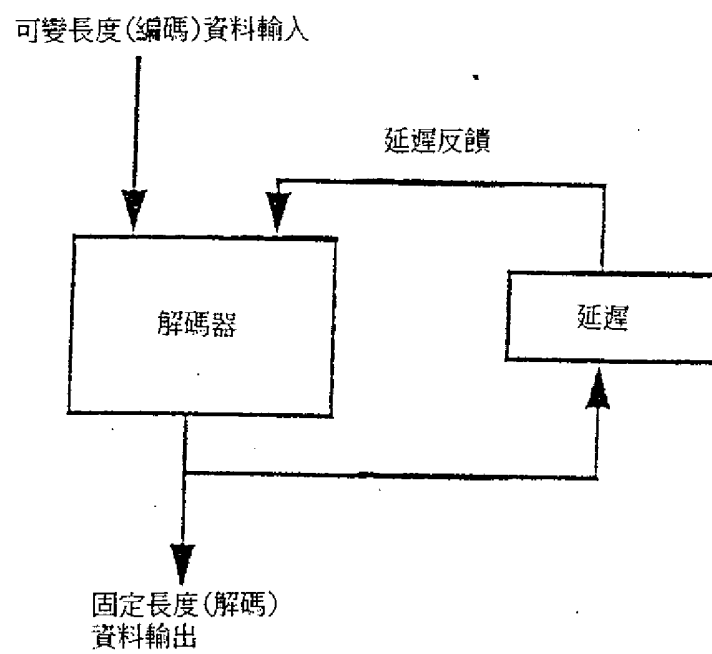
2442



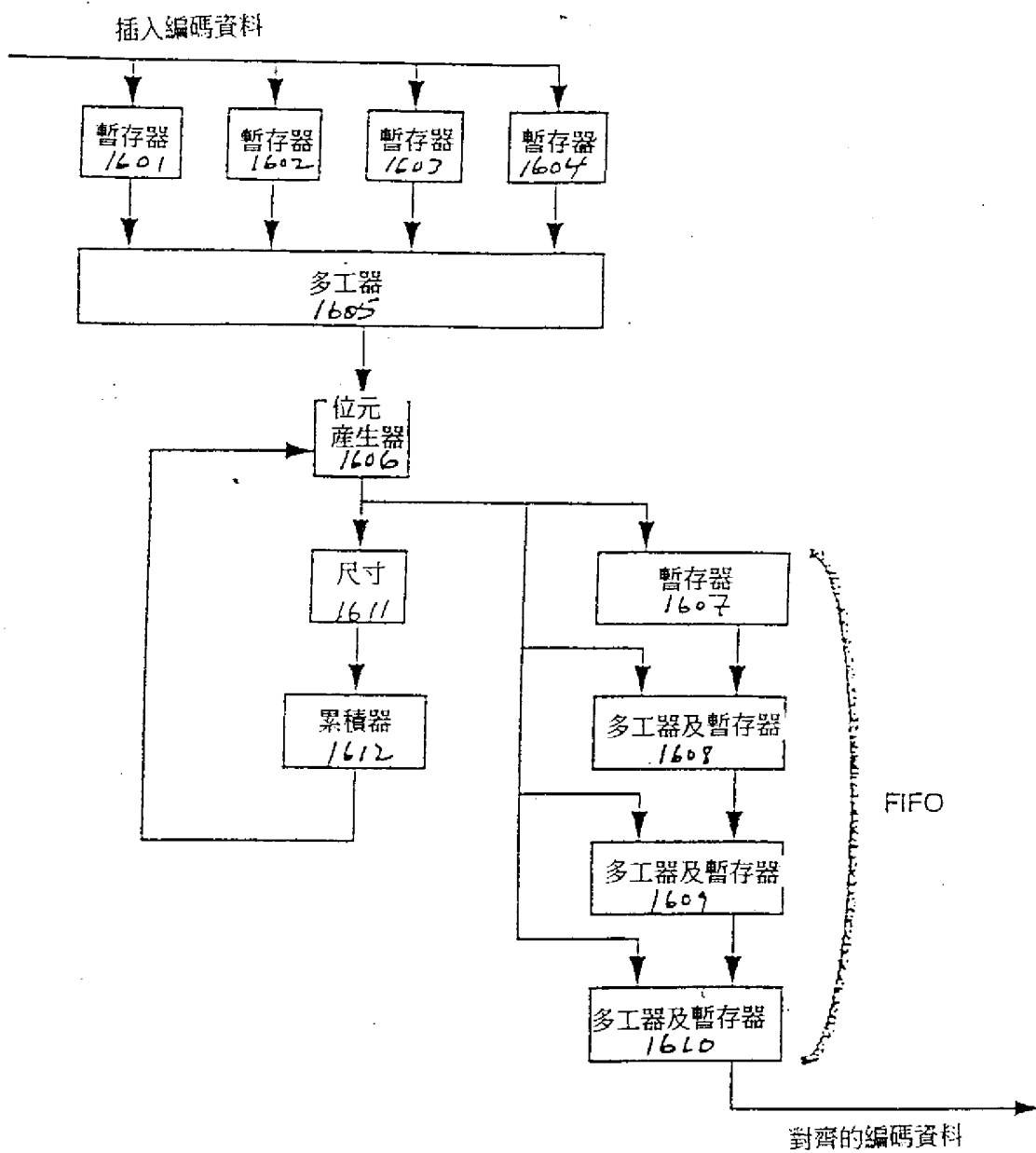
第14圖C



第15圖A



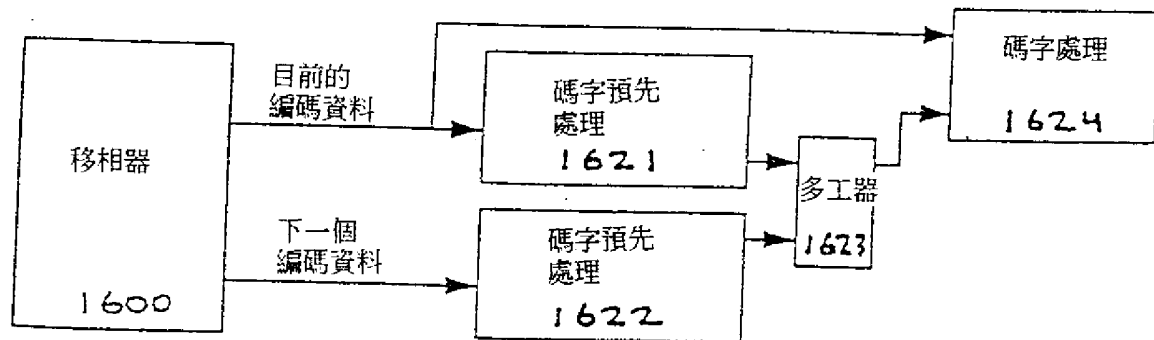
第15圖B



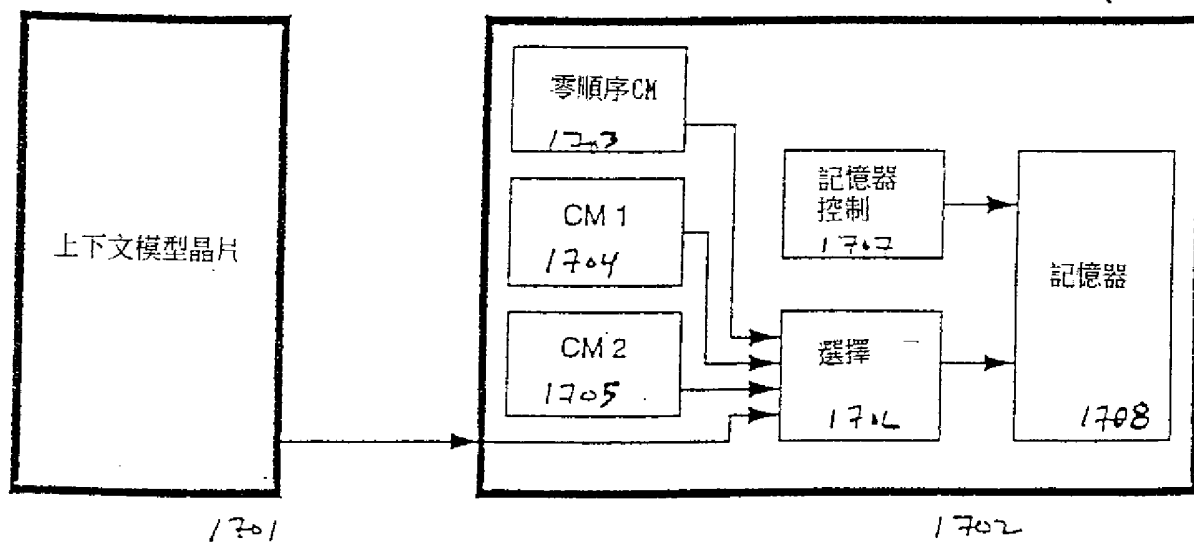
第16圖A

410311

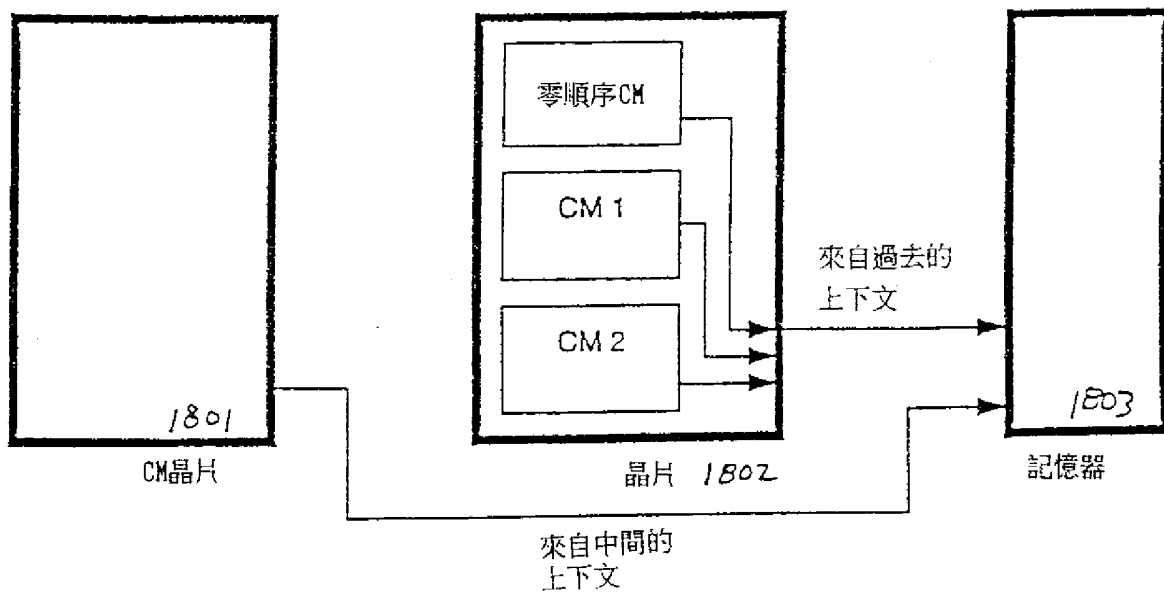
1620



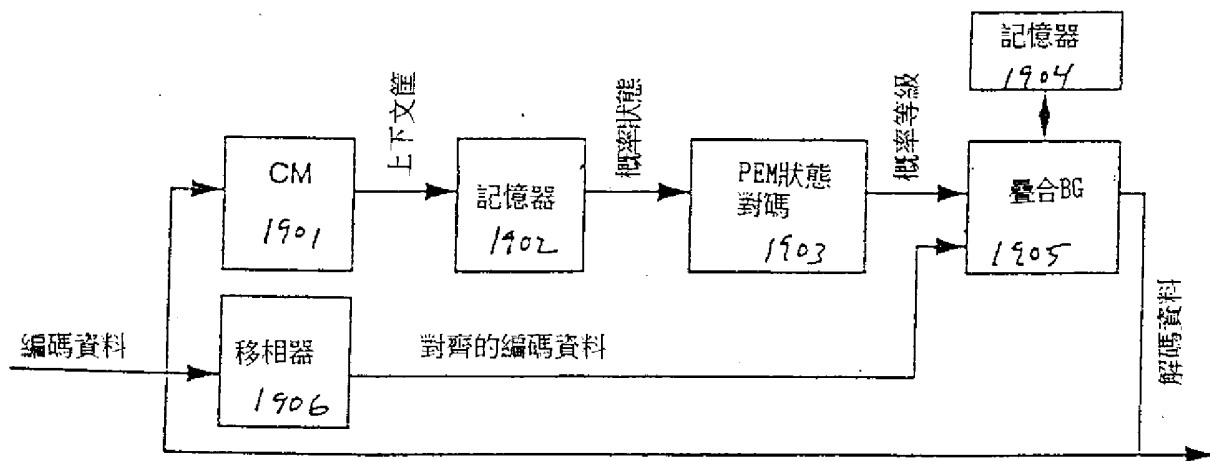
第16圖B



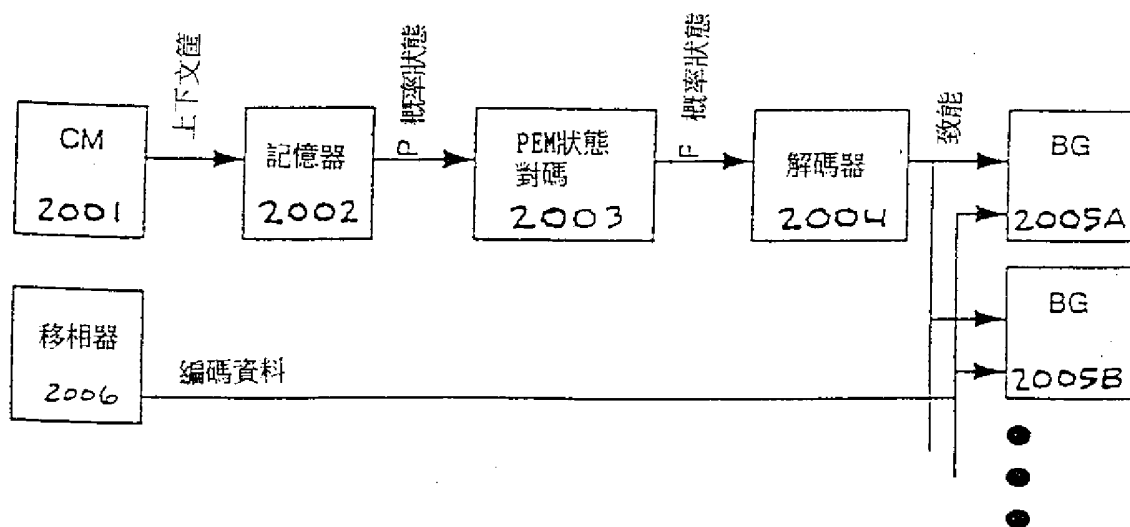
第17圖



第18圖

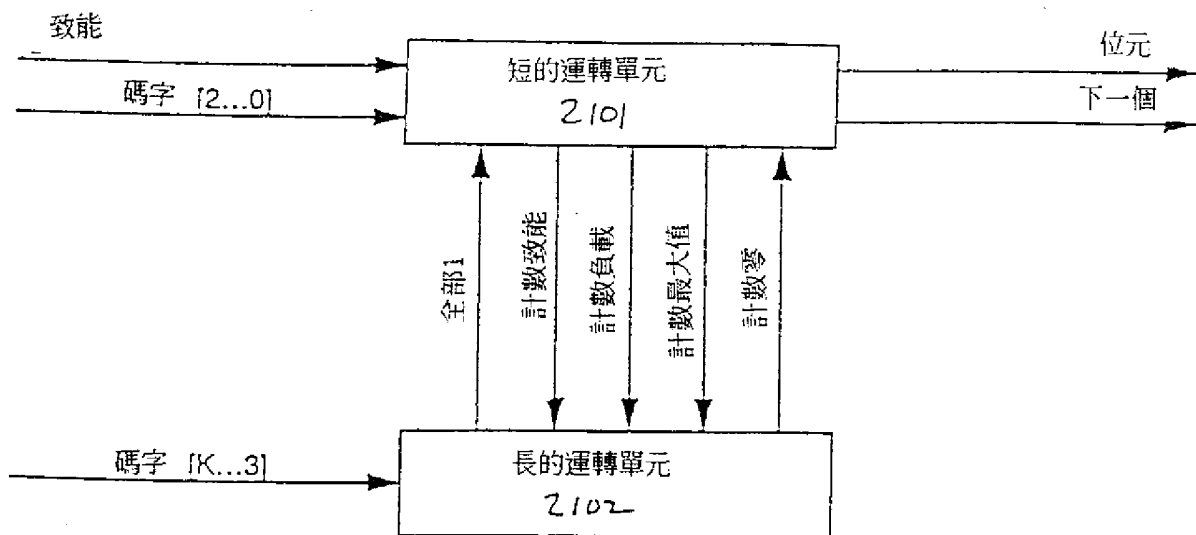


第19圖



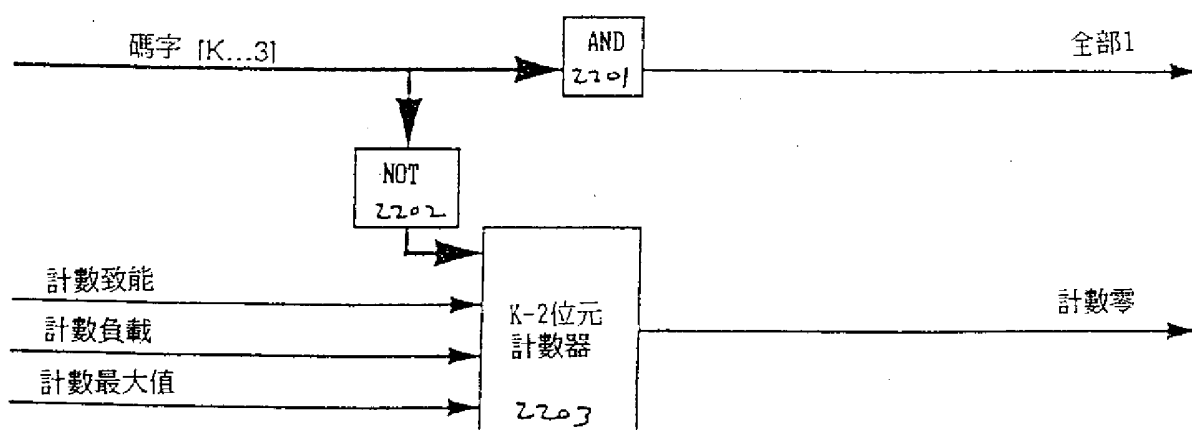
第20圖

410311

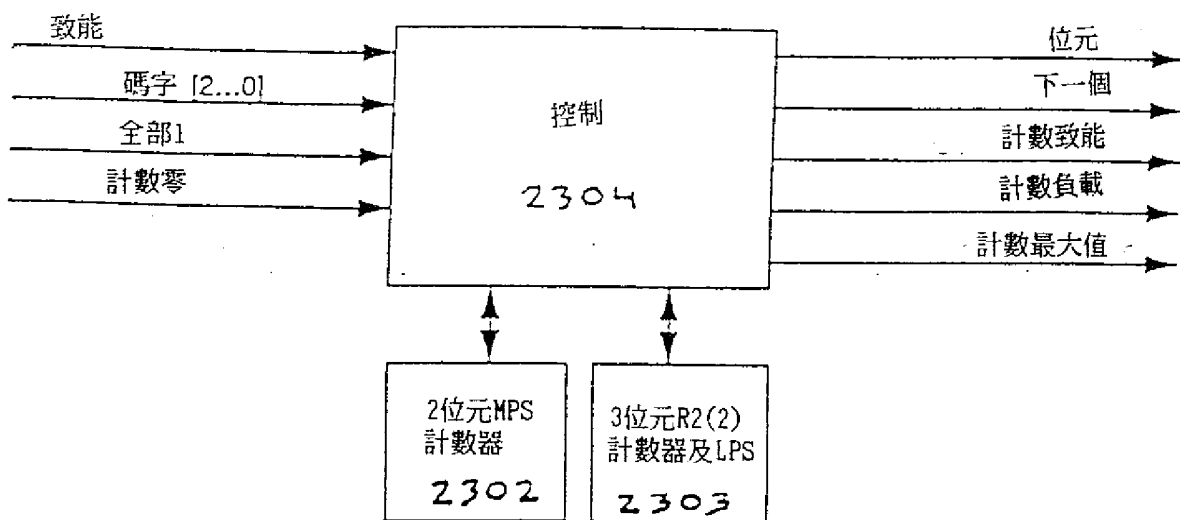


第 21 圖

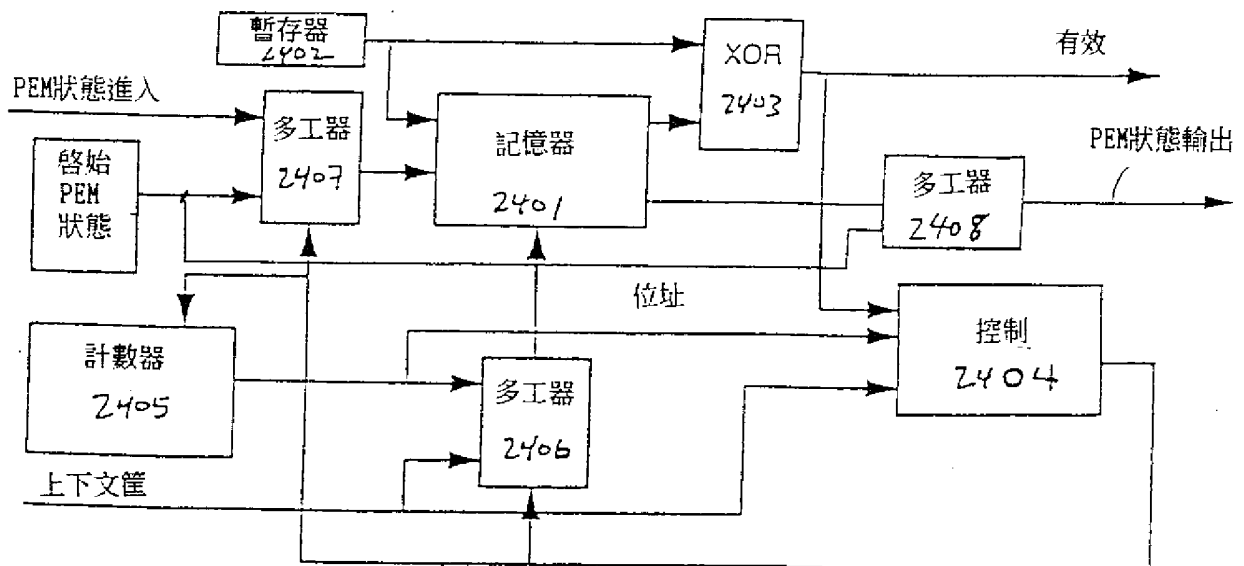
410311



第 22 圖

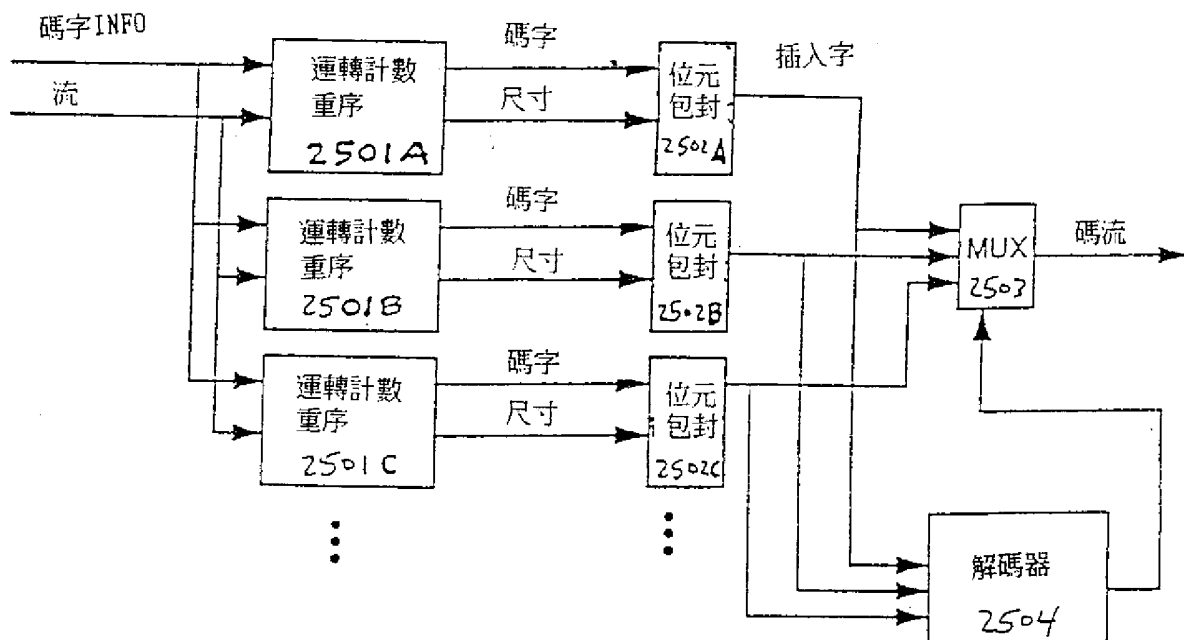


第 23 圖

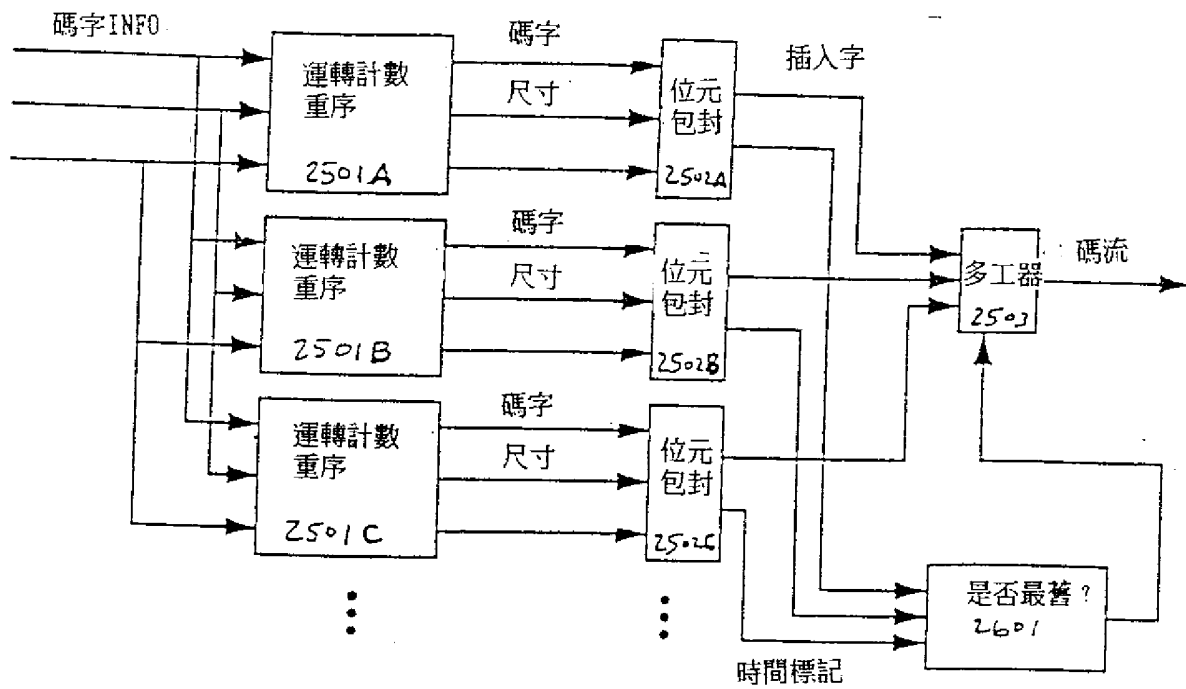


第24圖

410311

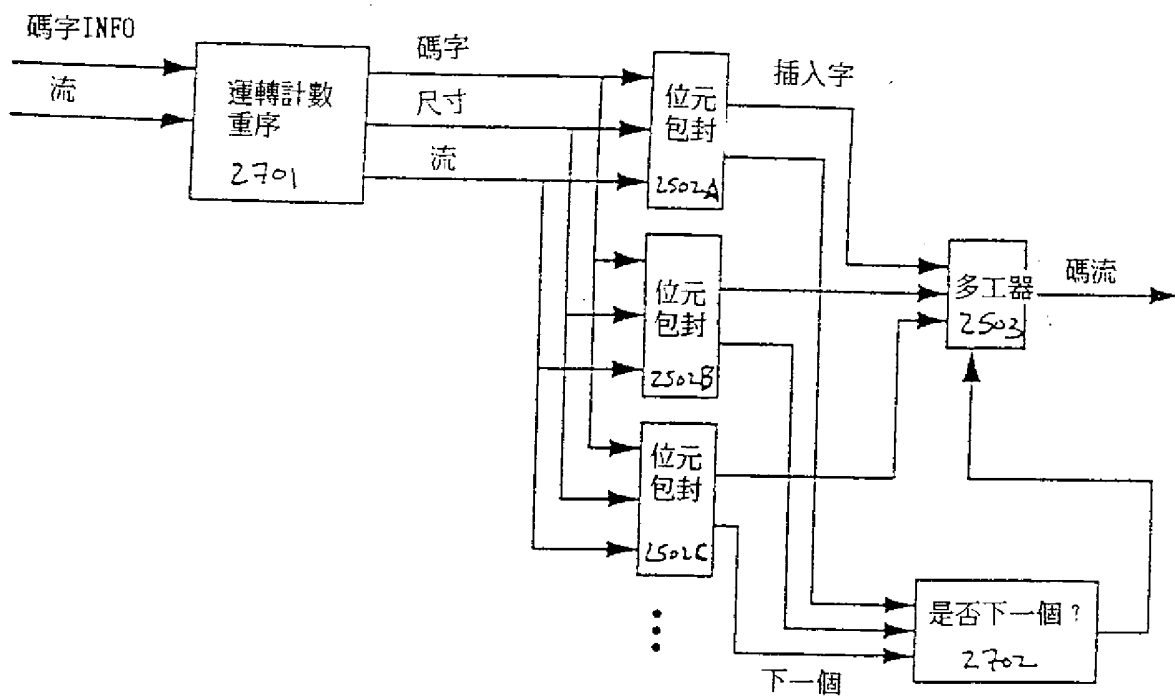


第25圖

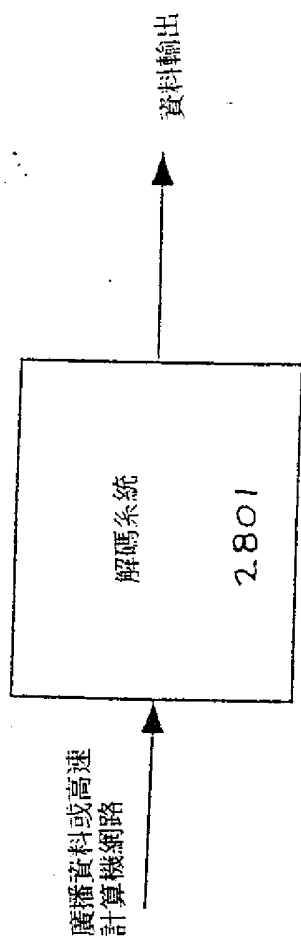


第26圖

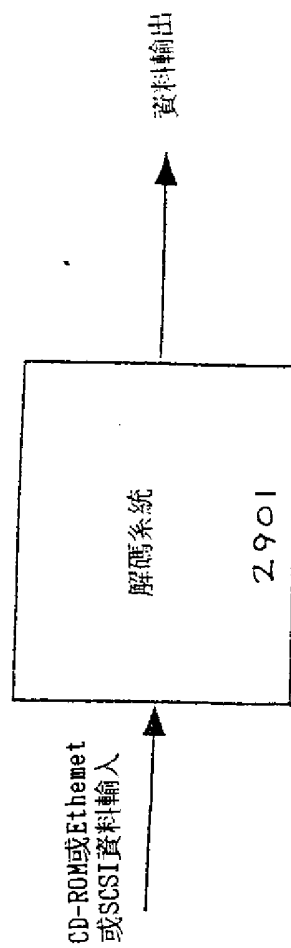
410311



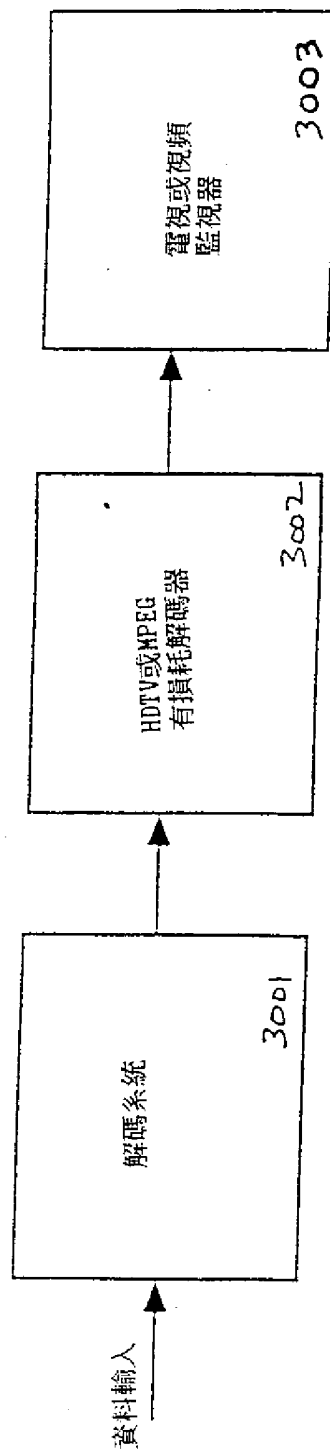
第 27 圖



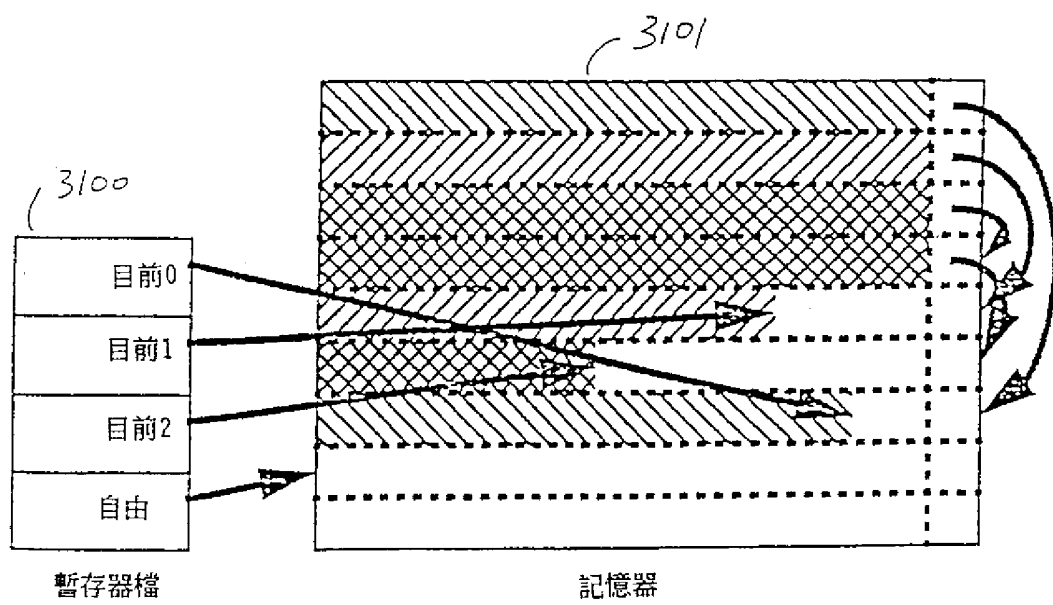
第28圖



第29圖

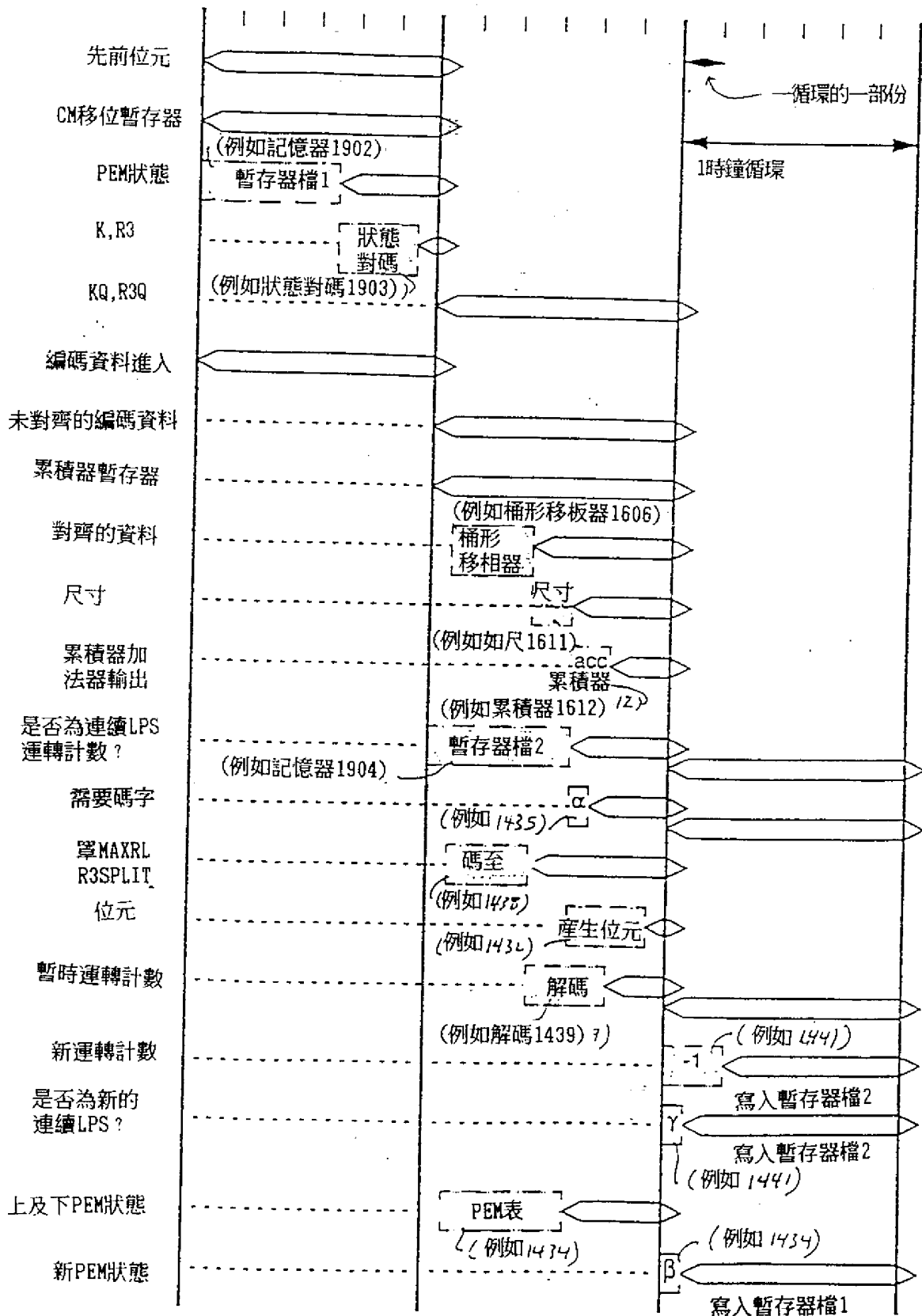


第30圖



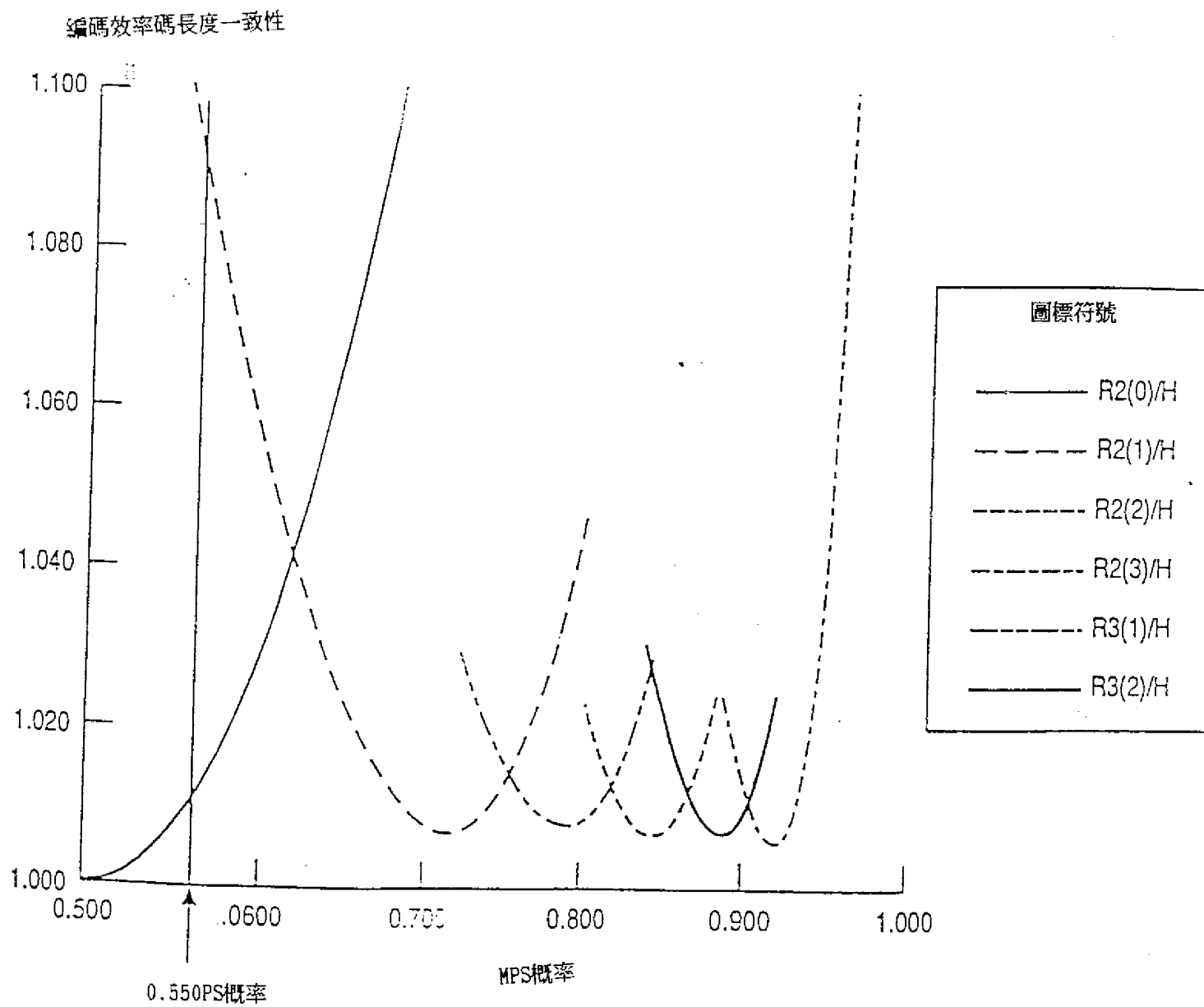
第 31 圖

時間圖



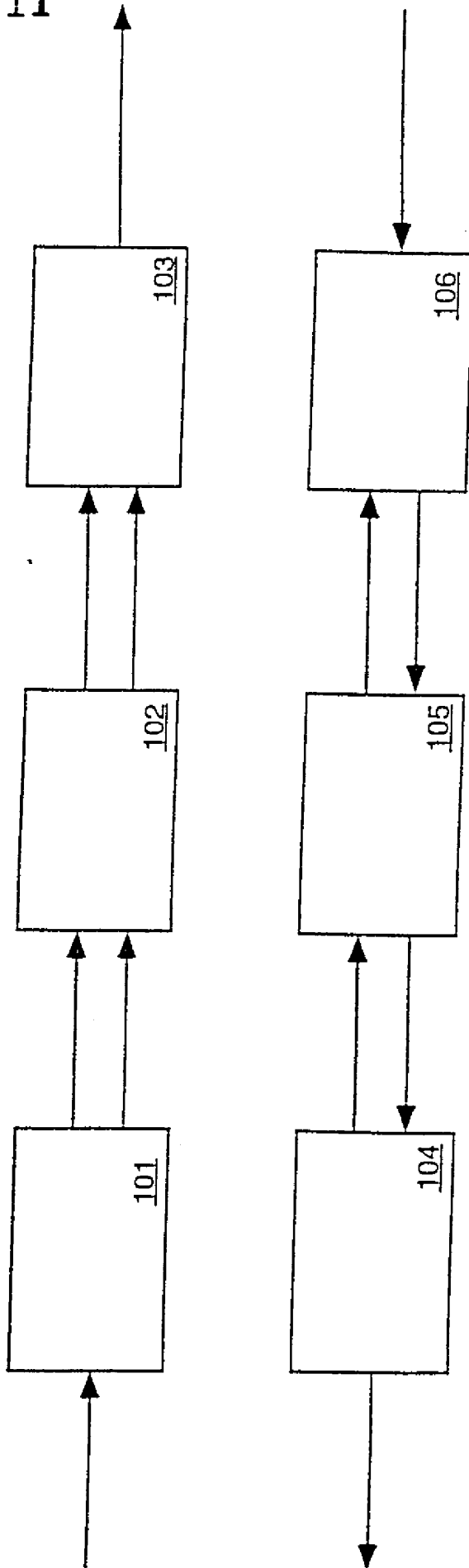
第32圖

410311

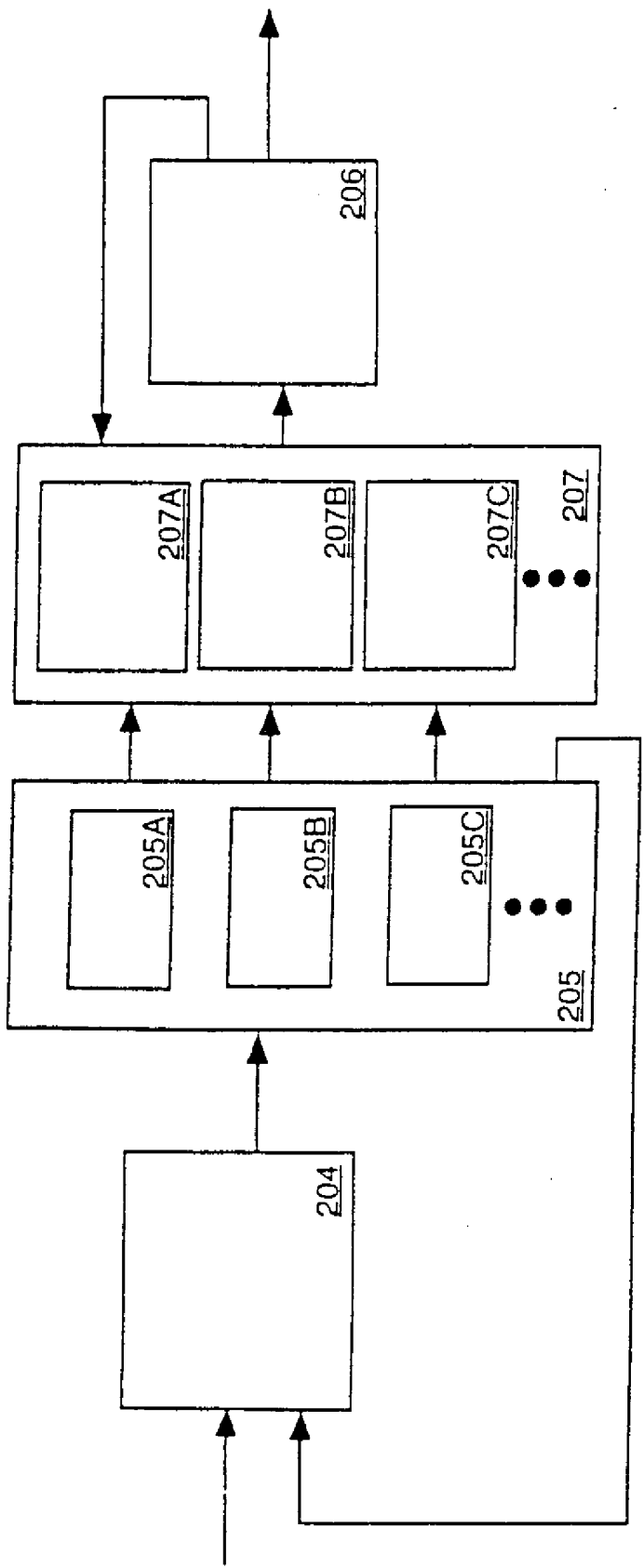


第 33 圖

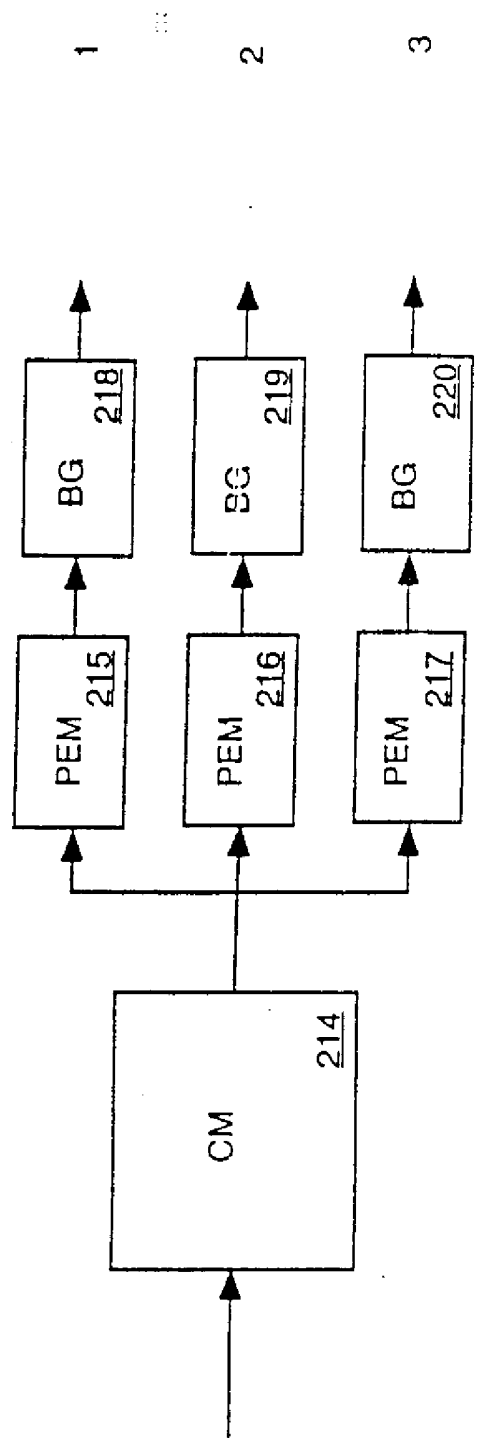
410311



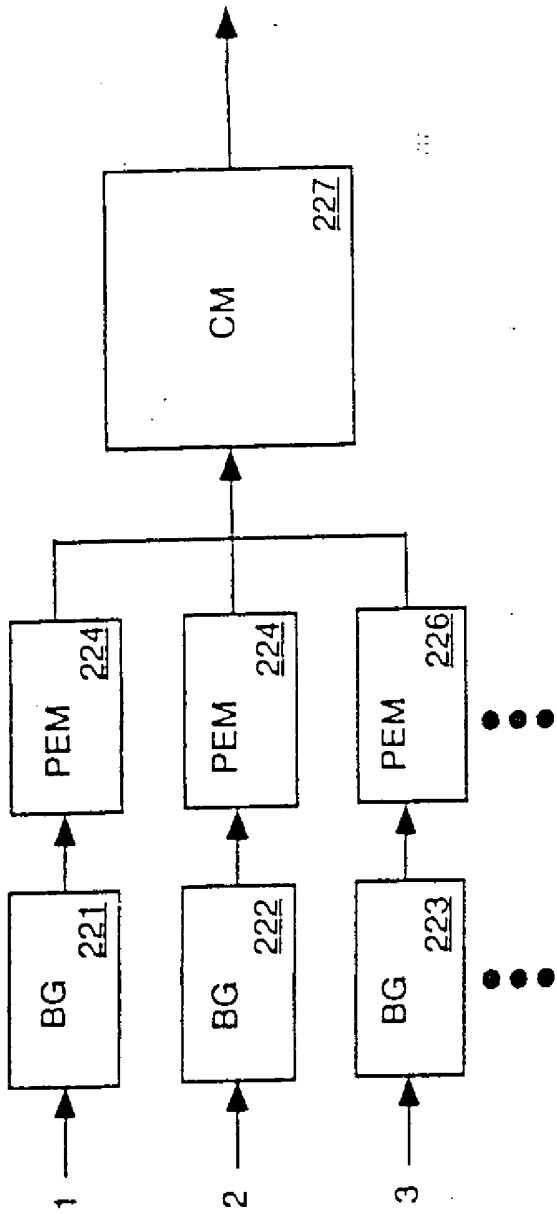
第 1 圖



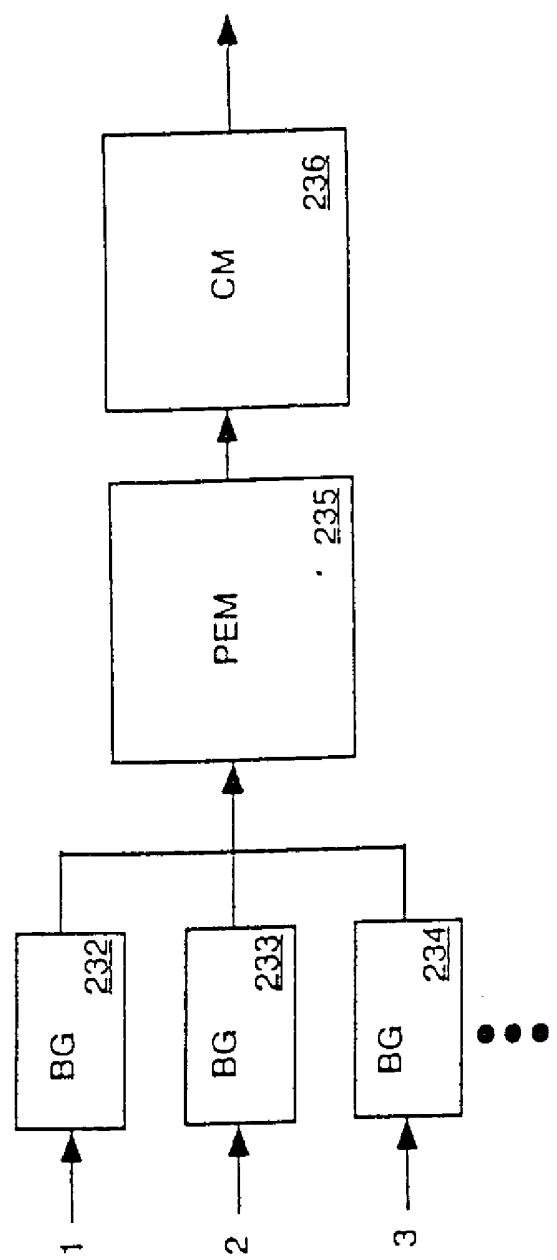
第2A圖



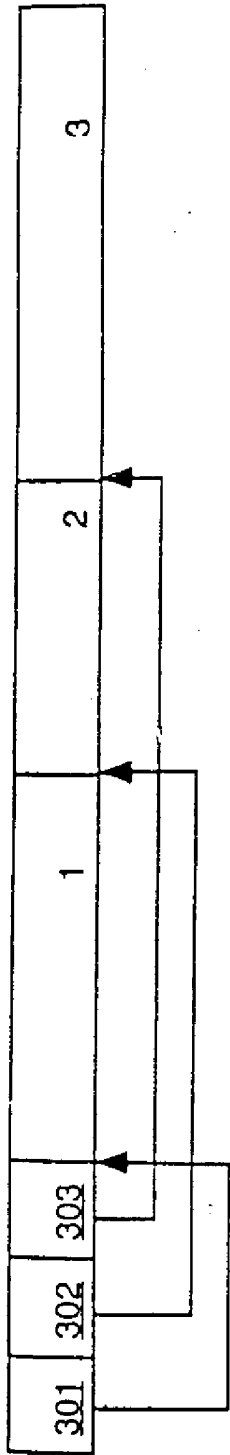
第2B圖



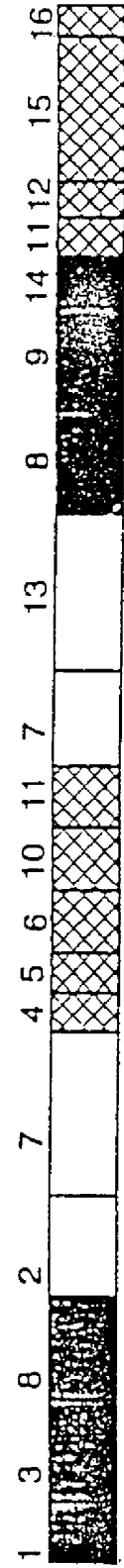
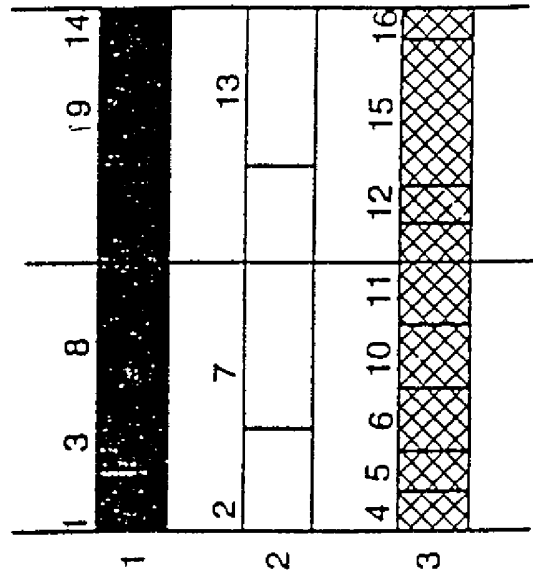
第2C圖



第2D圖



第 3 圖



第 4 圖

410311

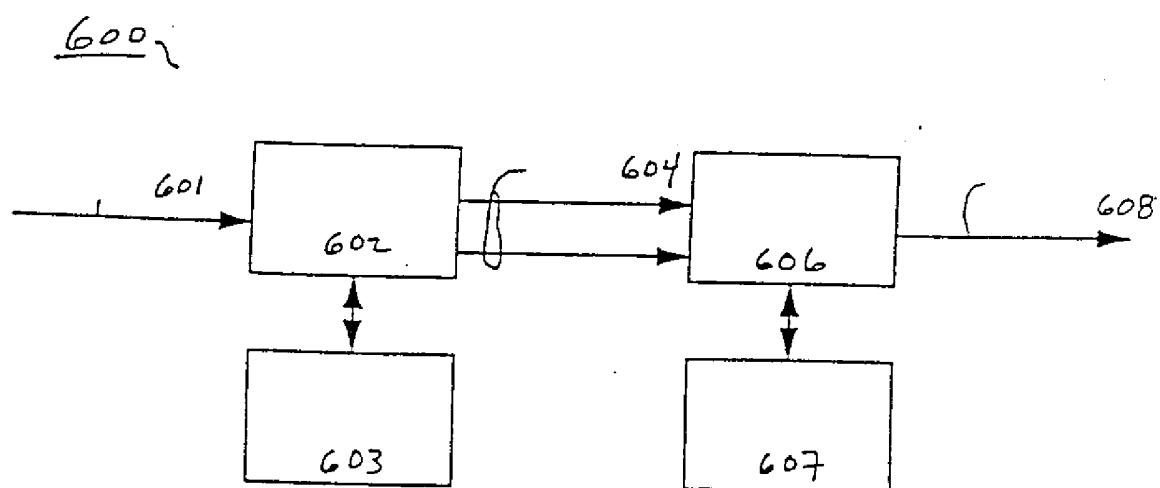
-35	R2(12)
-34	R3(11)
-33	R2(11)
-32	R3(10)
-31	R2(10)
-30	R3(9)
-29	R2(9)
-28	R3(8)
-27	R2(8)
-26	R3(7)
-25	R2(7)
-24	R3(6)
-23	R2(6)
-22	R3(5)
-21	R2(5)
-20	R3(4)
-19	R2(4)
-18	R3(3)
-17	R2(3)
-16	R3(2)
-15	R2(2)
-14	R3(1)
-13	R3(1)
-12	R3(1)

-11	R2(1)
-10	R2(1)
-9	R2(1)
-8	R2(1)
-7	R2(1)
-6	R2(1)
-5	R2(0)
-4	R2(0)
-3	R2(0)
-2	R2(0)
-1	R2(0)
-0	R2(0)
0	R2(0)
1	R2(0)
2	R2(0)
3	R2(0)
4	R2(0)
5	R2(0)
6	R2(1)
7	R2(1)
8	R2(1)
9	R2(1)
10	R2(1)
11	R2(1)

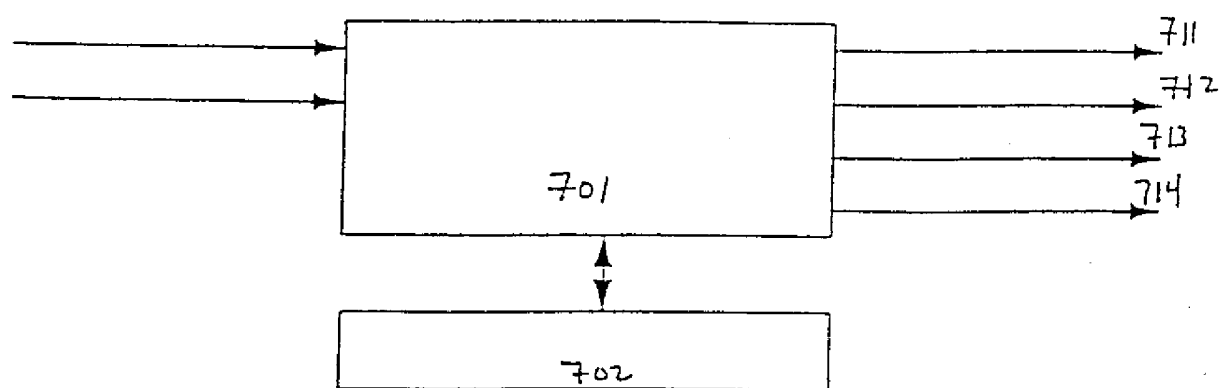
12	R3(1)
13	R2(1)
14	R3(1)
15	R2(2)
16	R3(2)
17	R2(3)
18	R3(3)
19	R2(4)
20	R3(4)
21	R2(5)
22	R3(5)
23	R2(6)
24	R3(6)
25	R2(7)
26	R3(7)
27	R2(8)
28	R3(8)
29	R2(9)
30	R3(9)
31	R2(10)
32	R3(10)
33	R2(11)
34	R3(11)
35	R2(12)

第 5 圖

410311

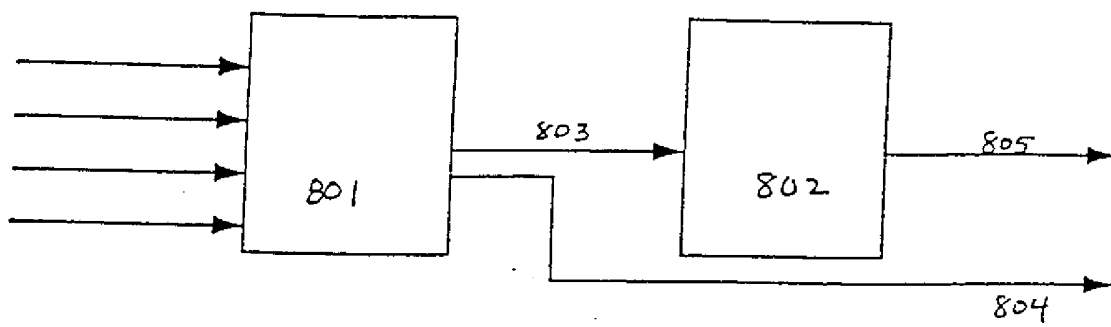


第 6 圖



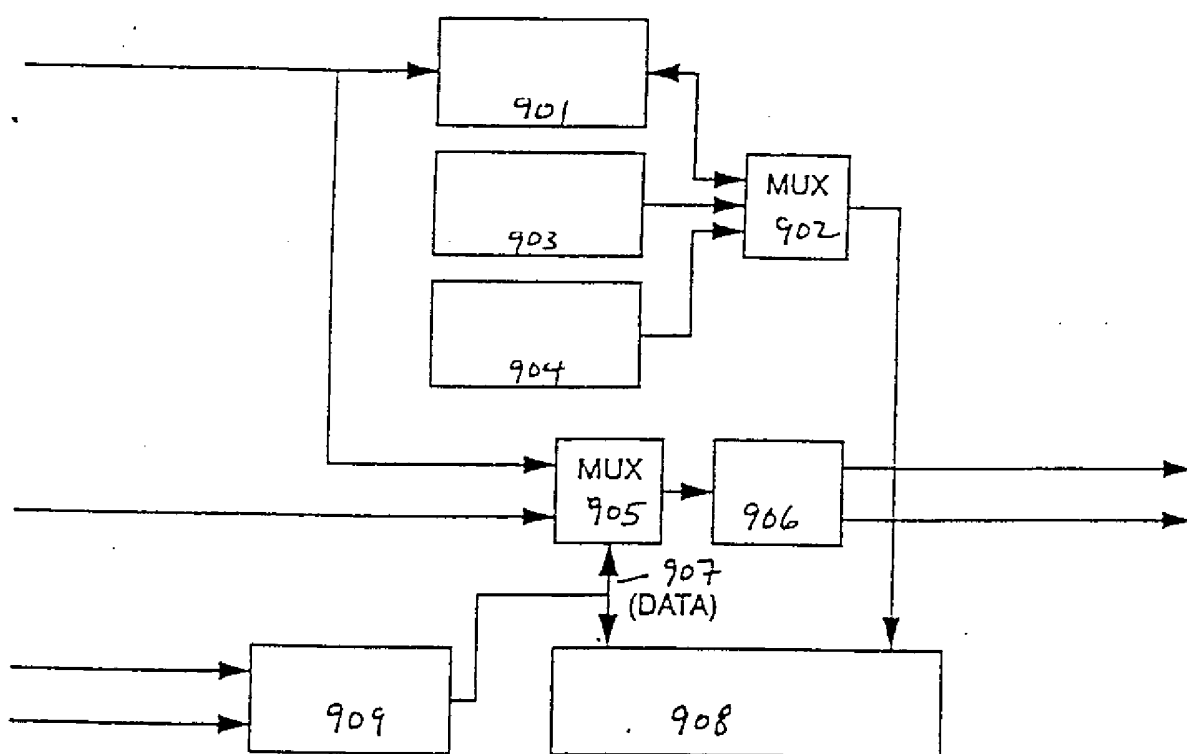
第 7 圖

410311

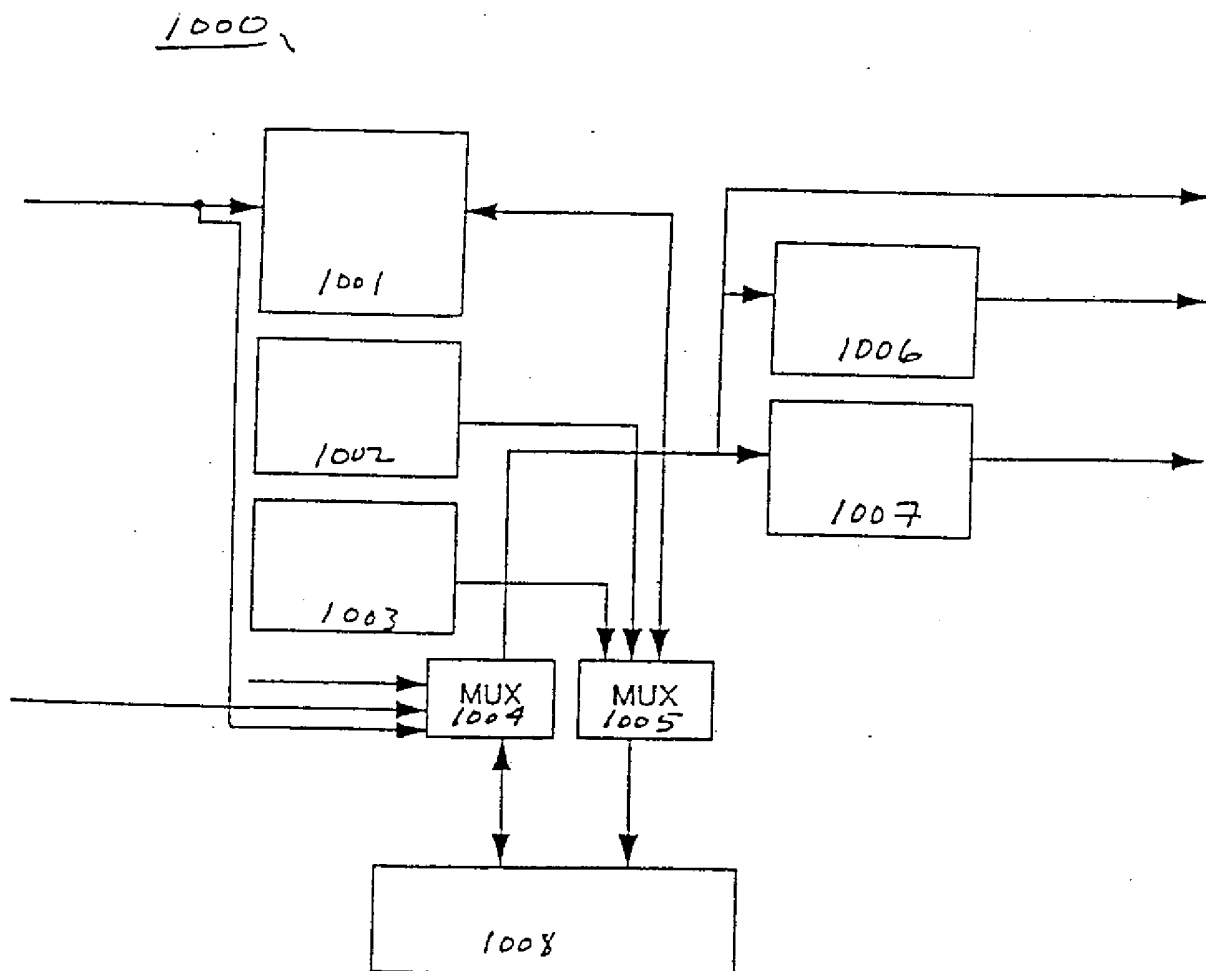


第 8 圖

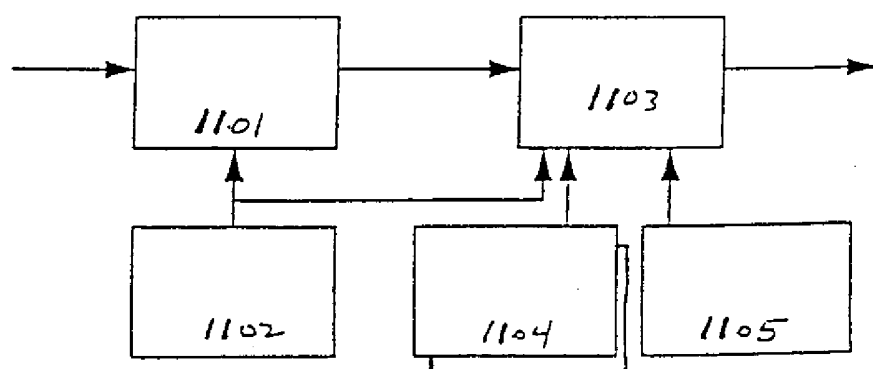
410311



第 9 圖

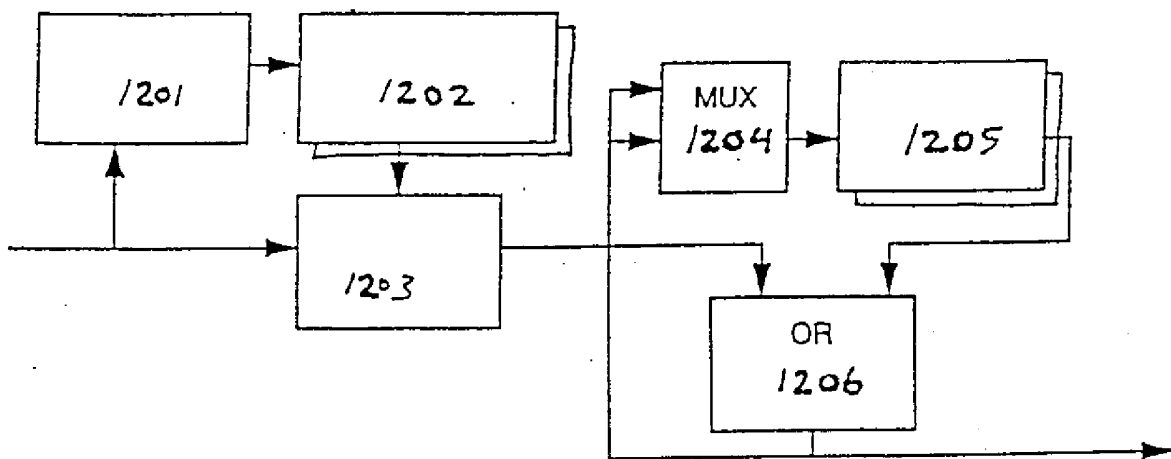


第10圖



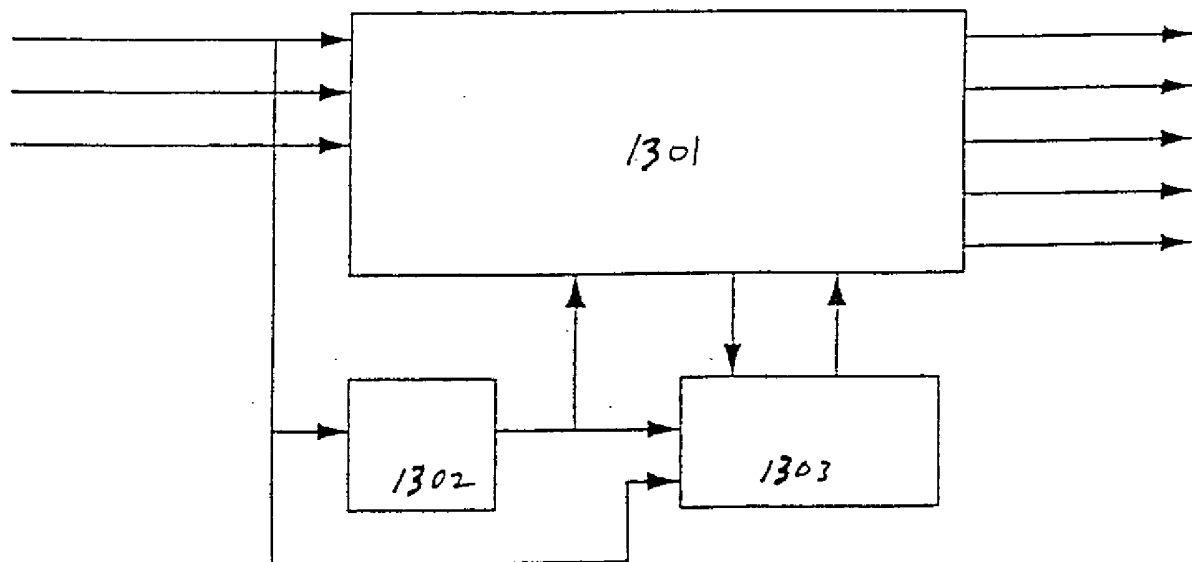
第11圖

410311



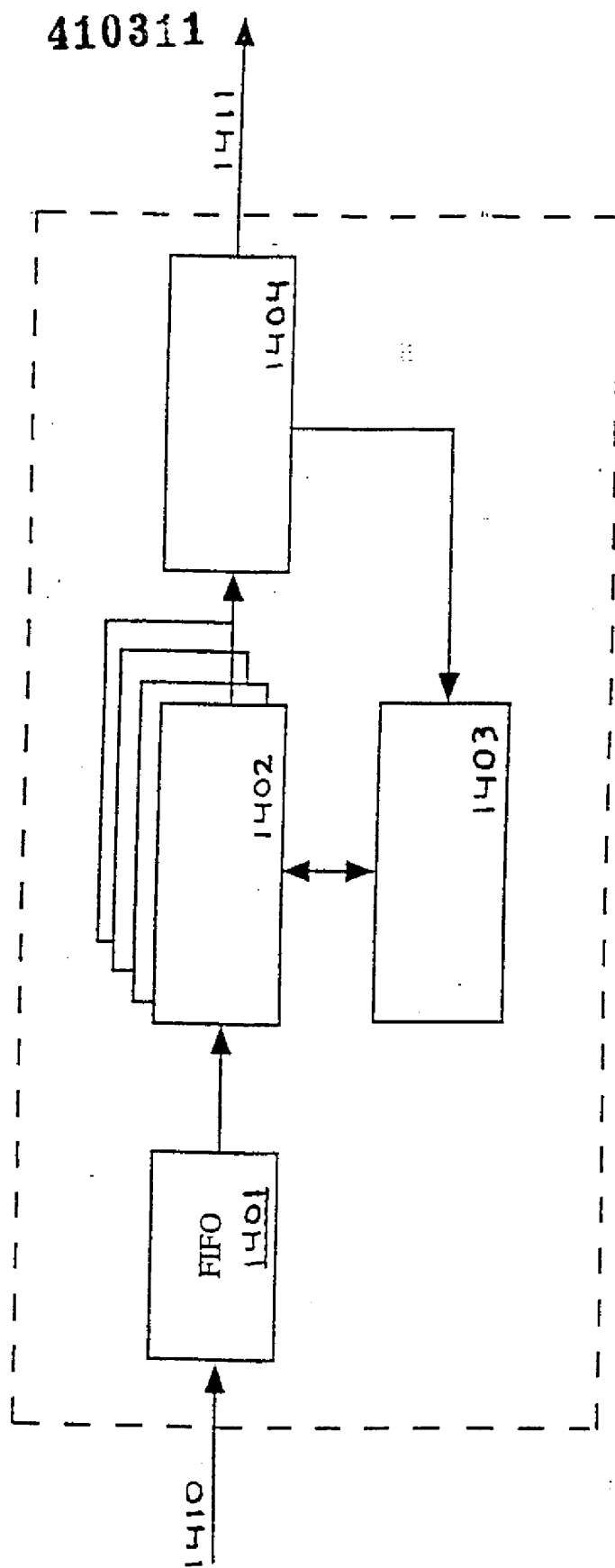
第12圖

410311

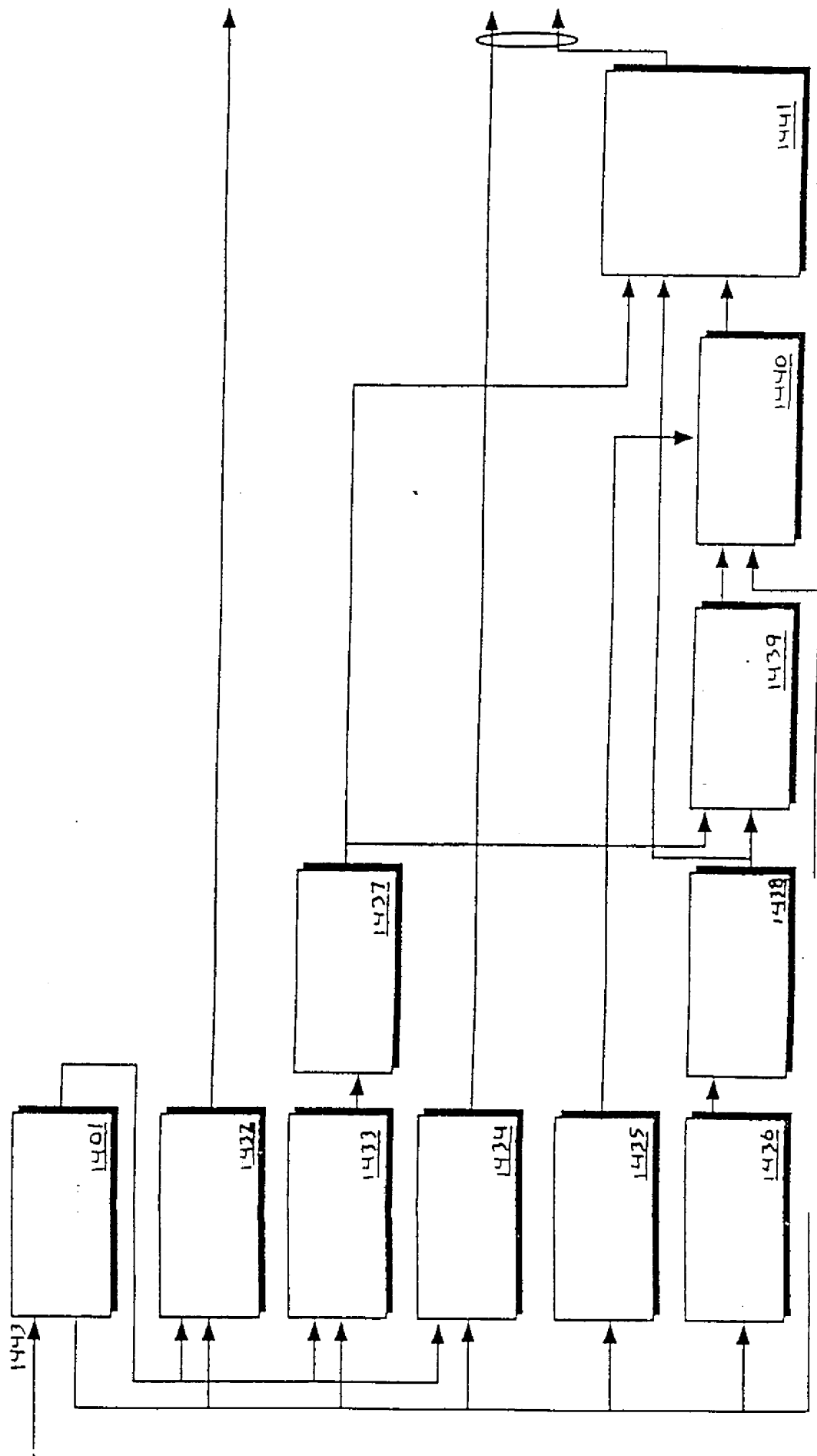


第13圖

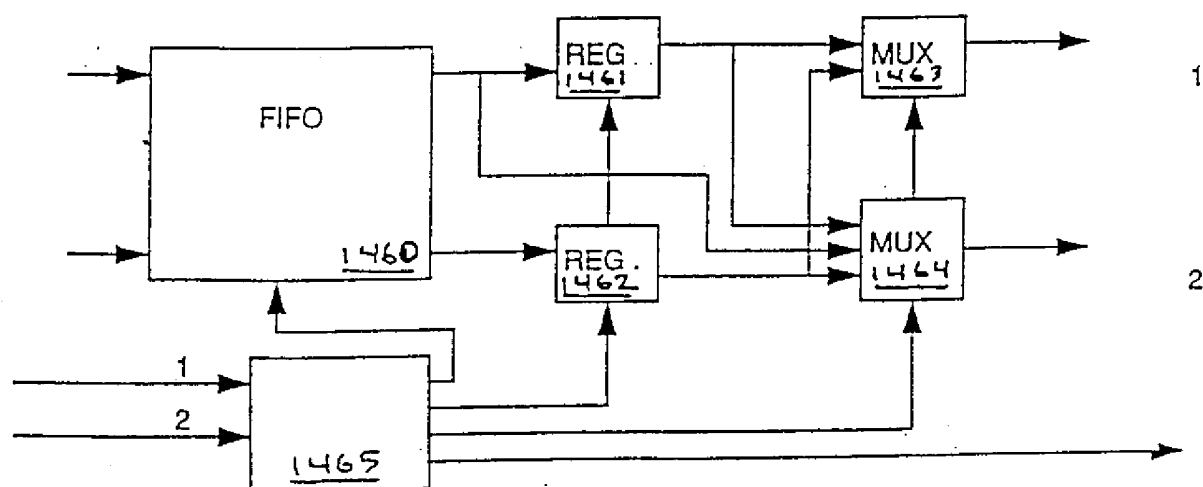
1400



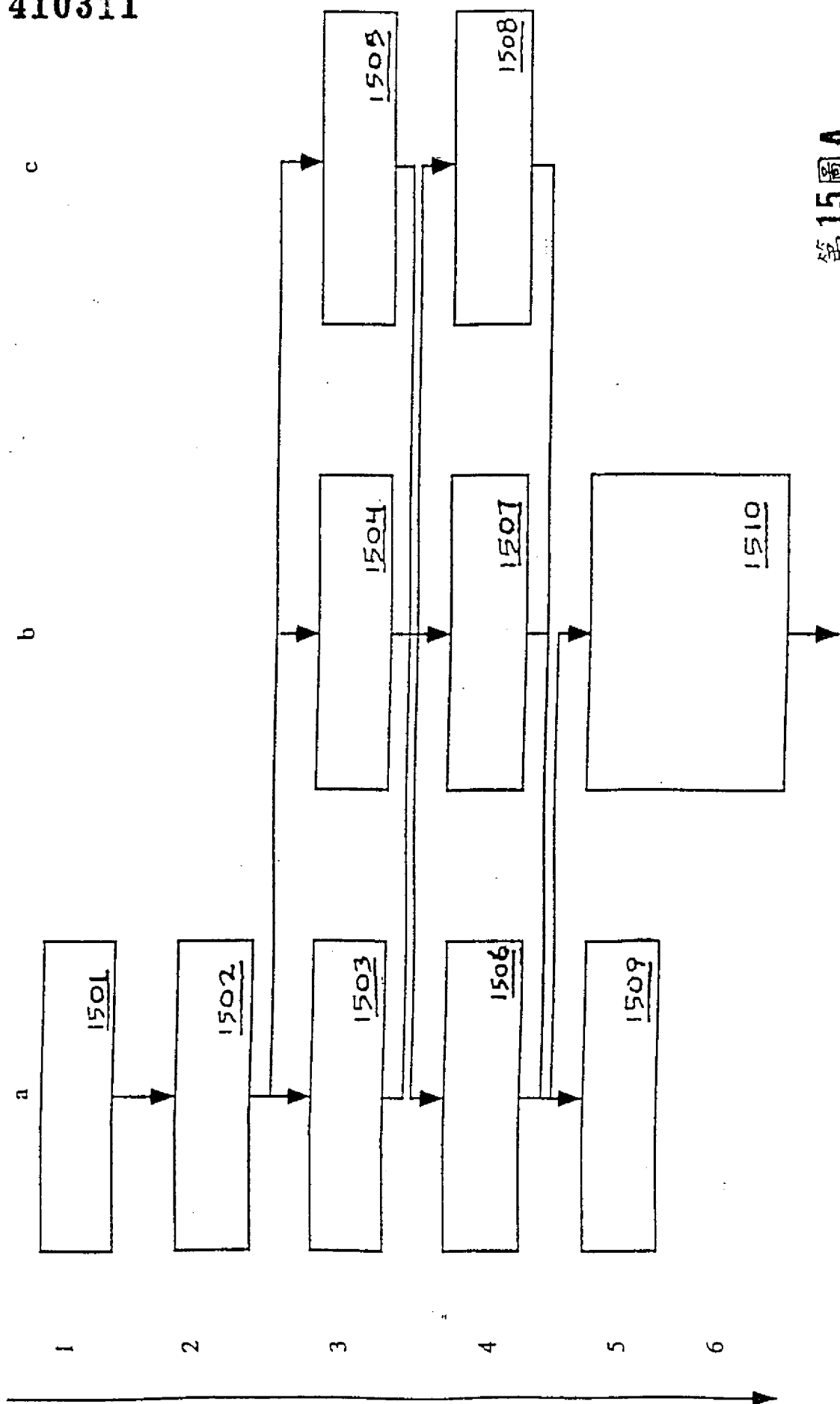
第14圖A



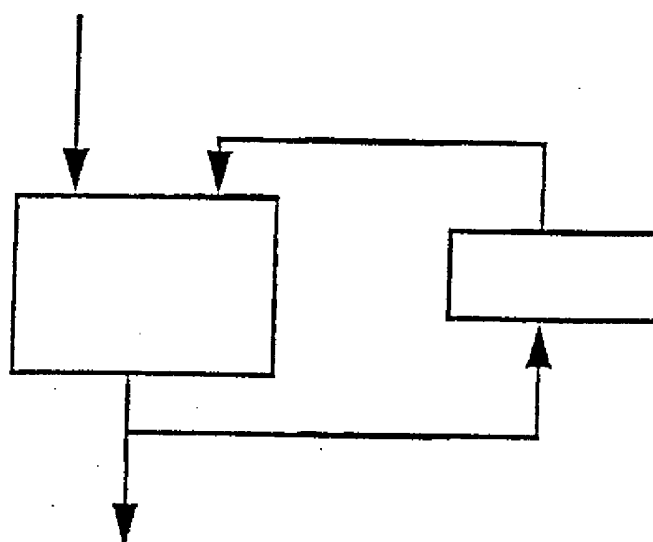
第14圖B



第14圖C

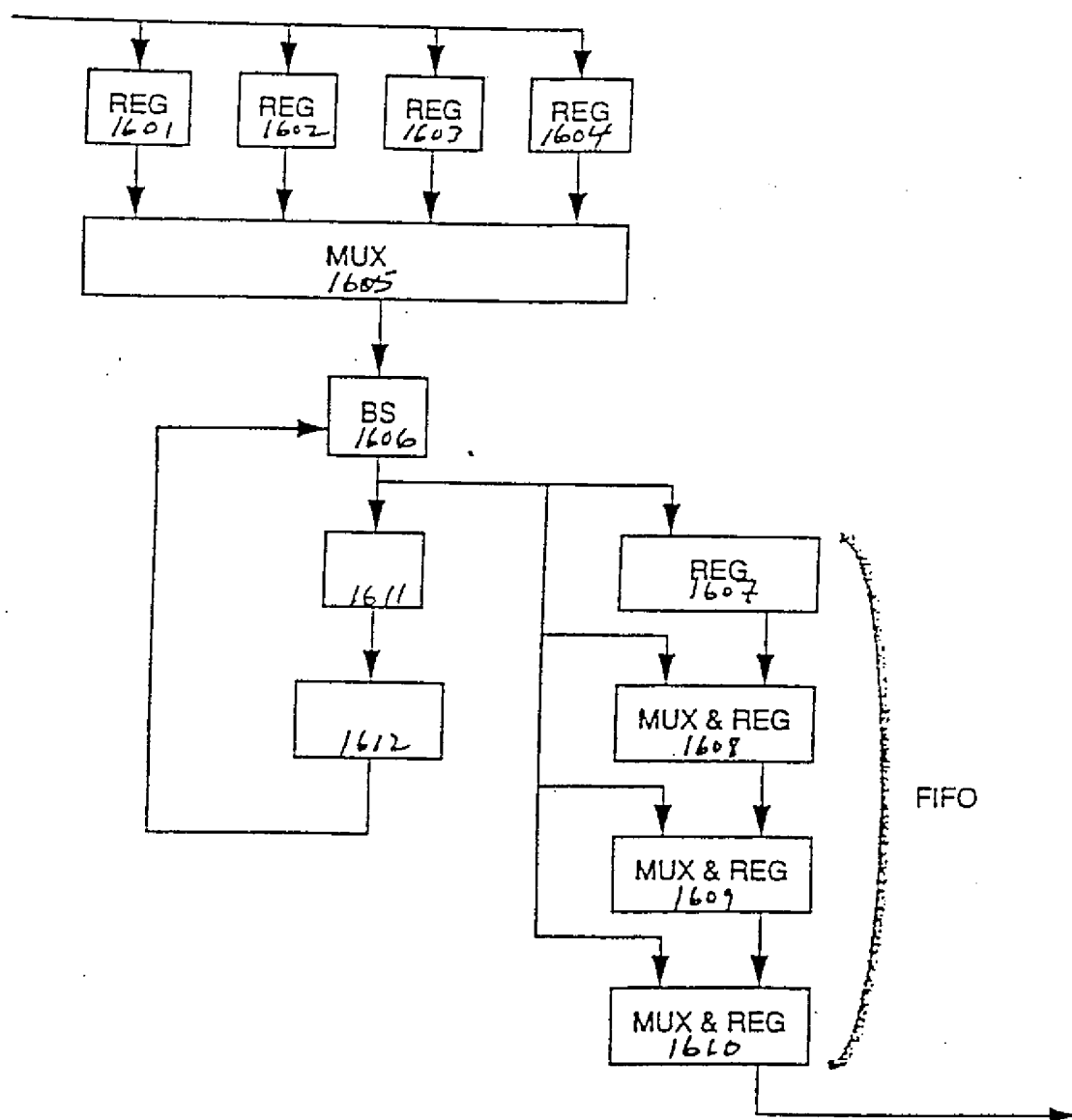


第15圖A

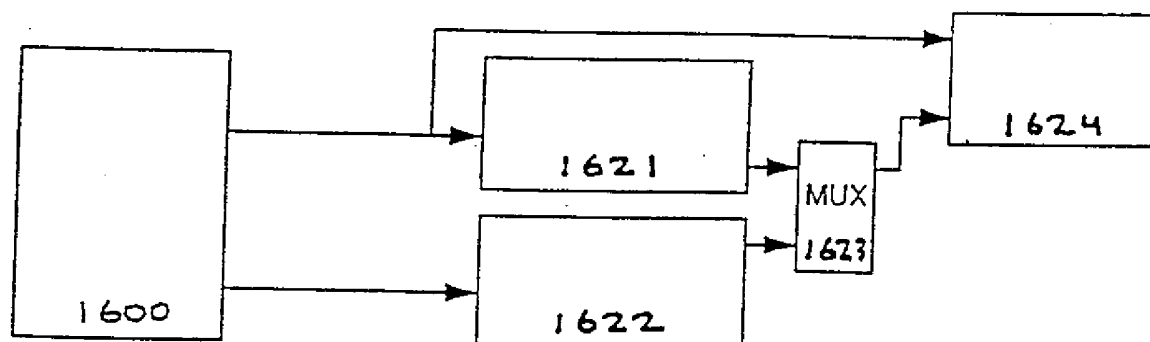


第15圖B

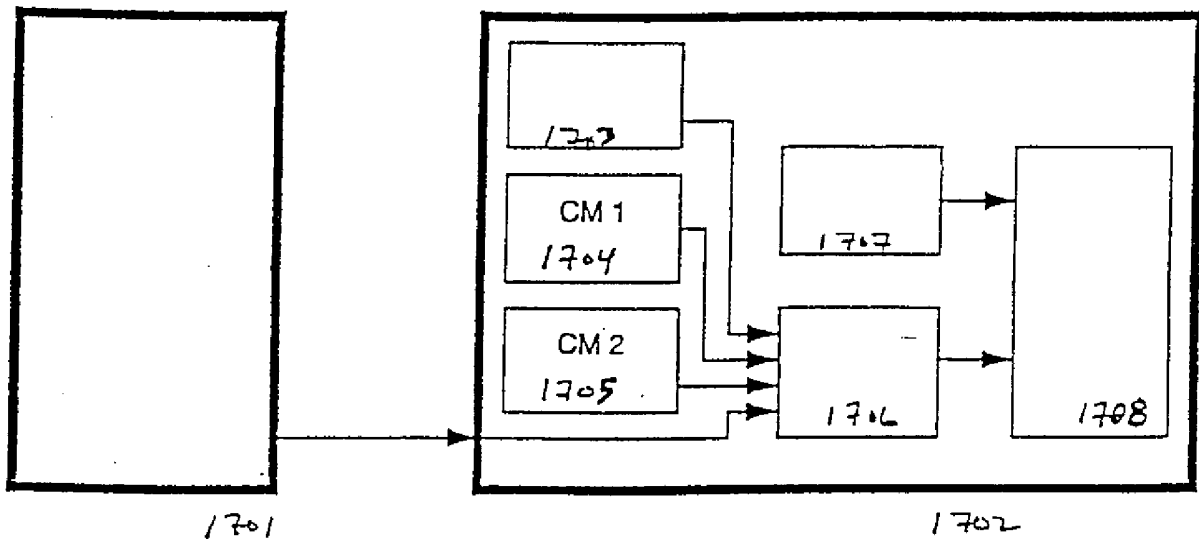
410311



第16圖A

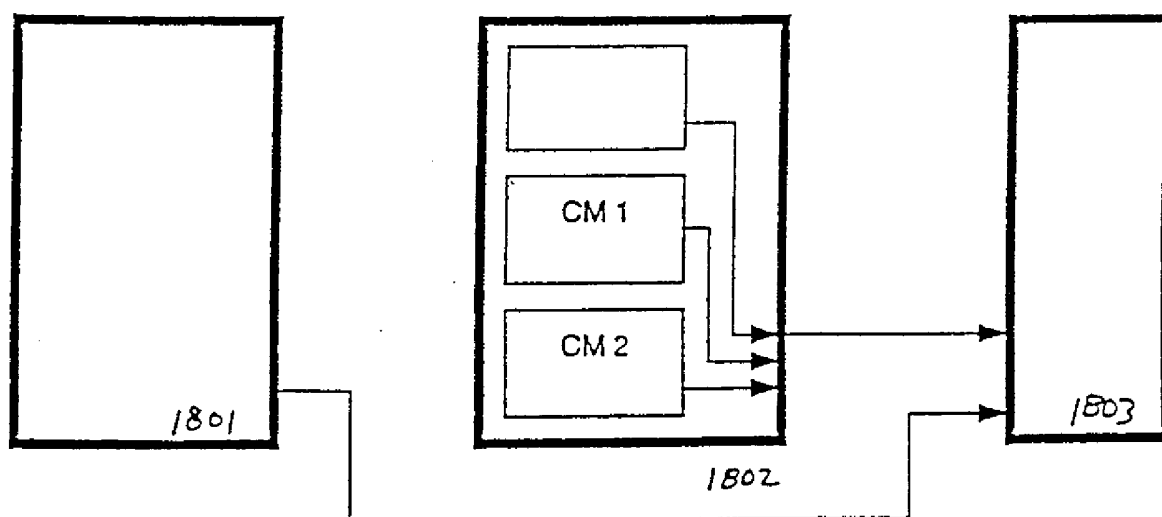
1620

第16圖B



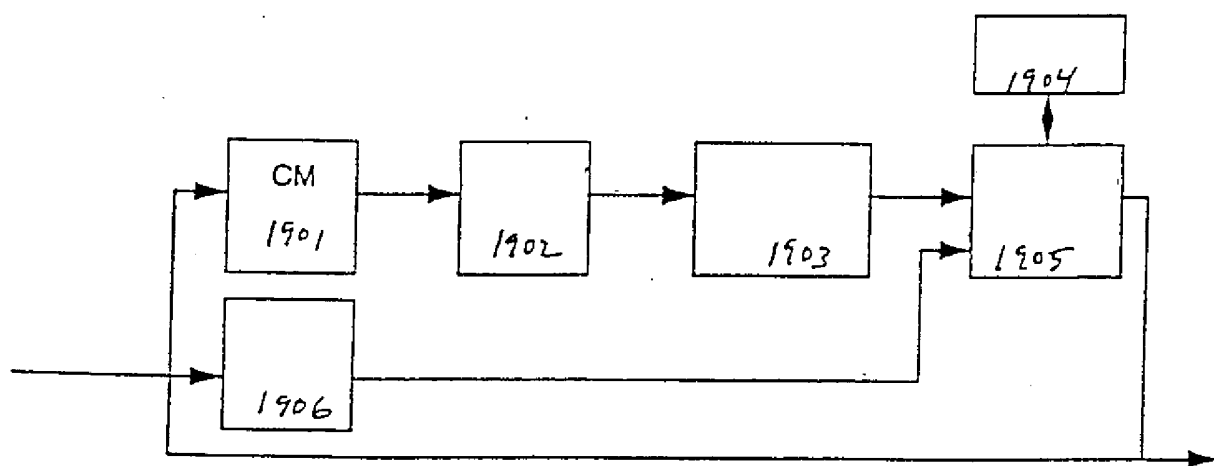
第17圖

410311



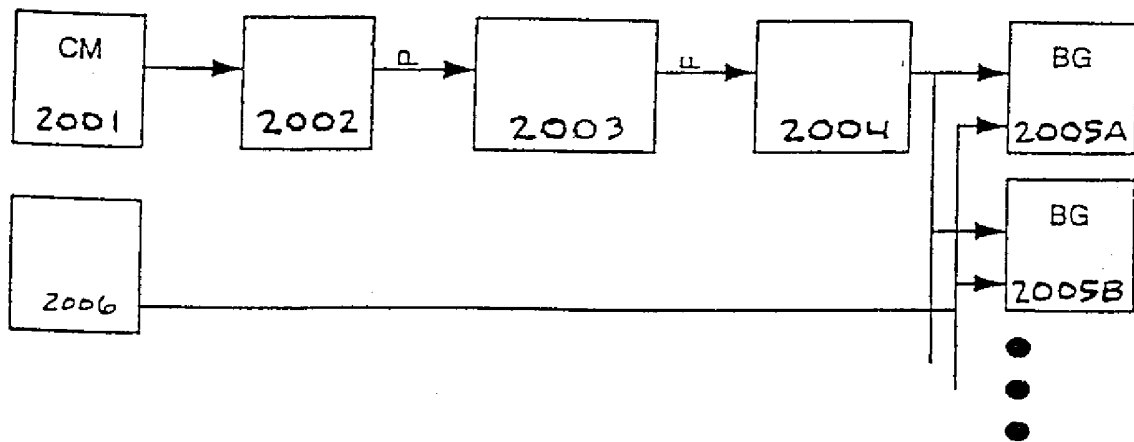
第18圖

410311



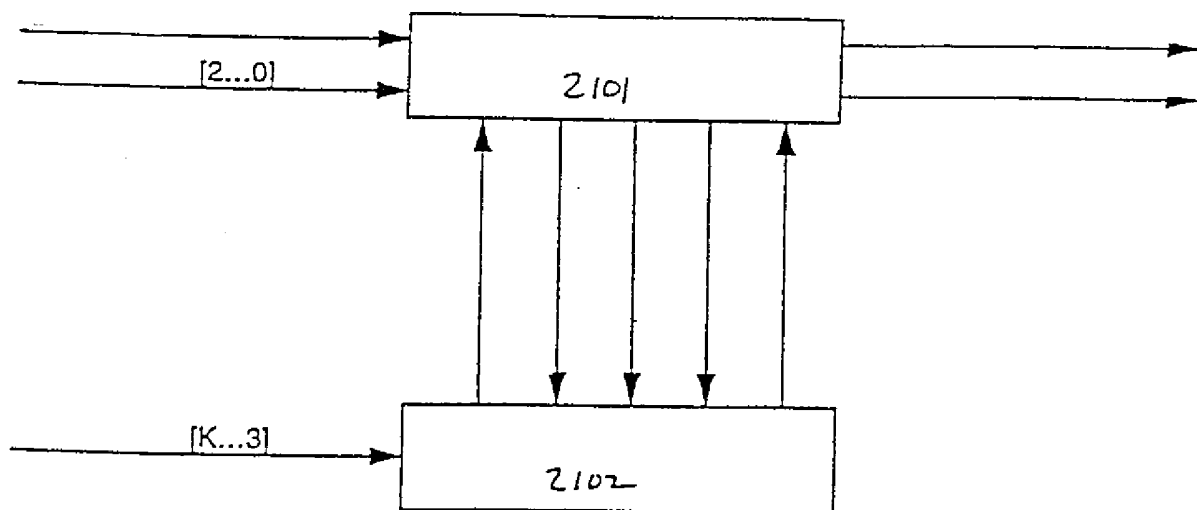
第19圖

410311



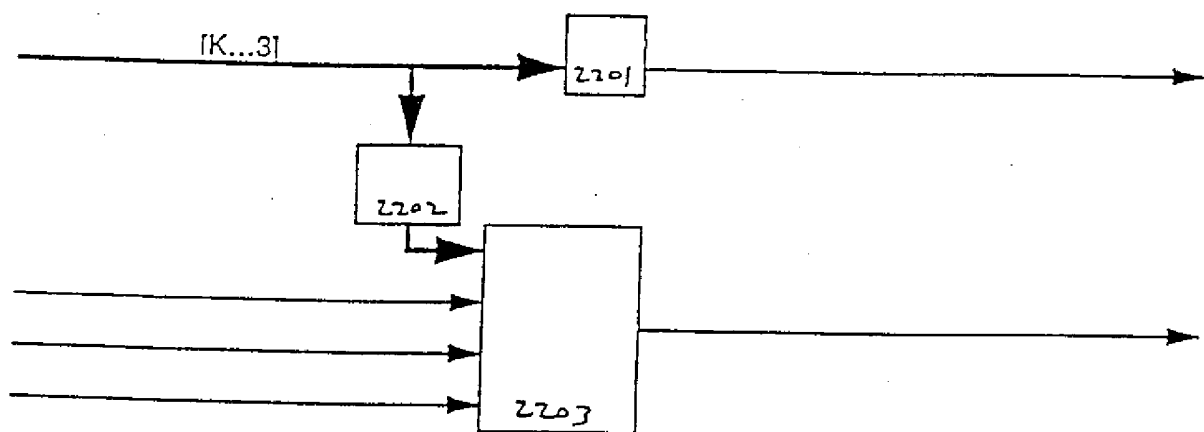
第20圖

410311

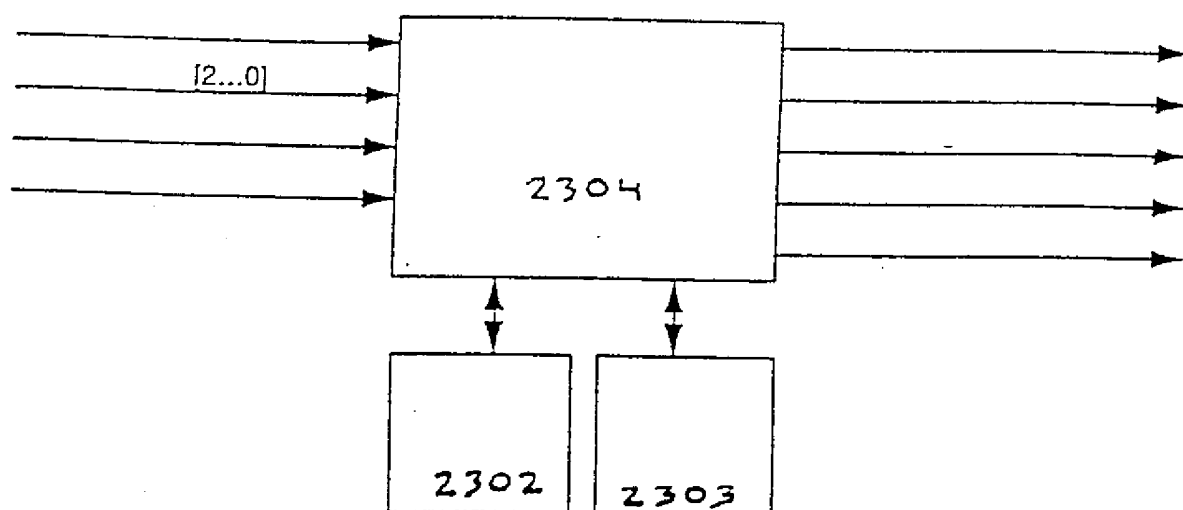


第21圖

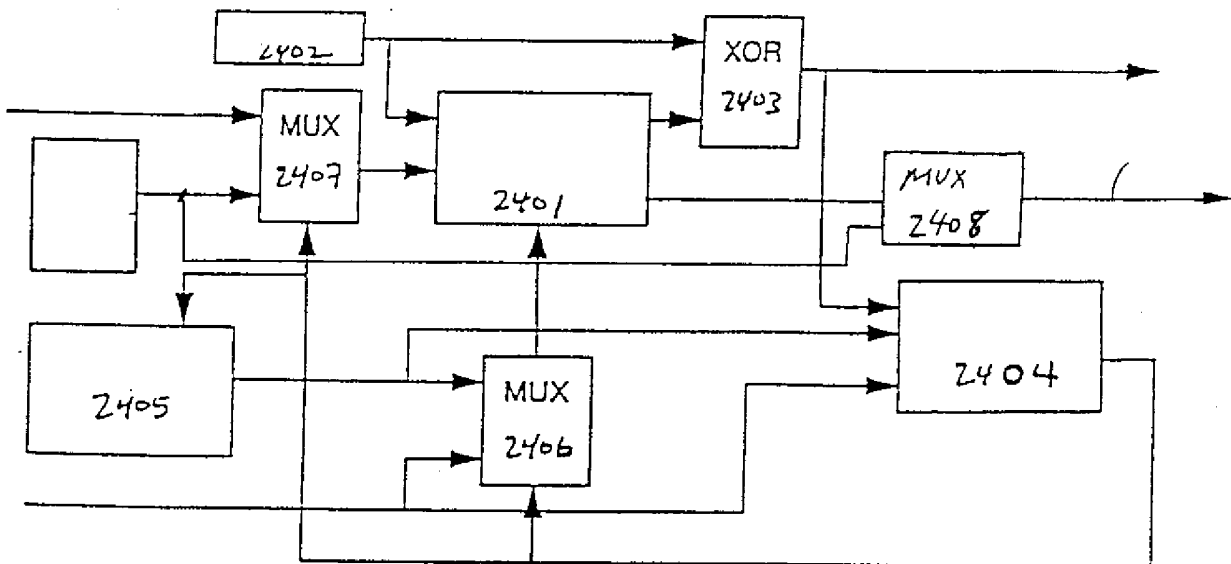
410311



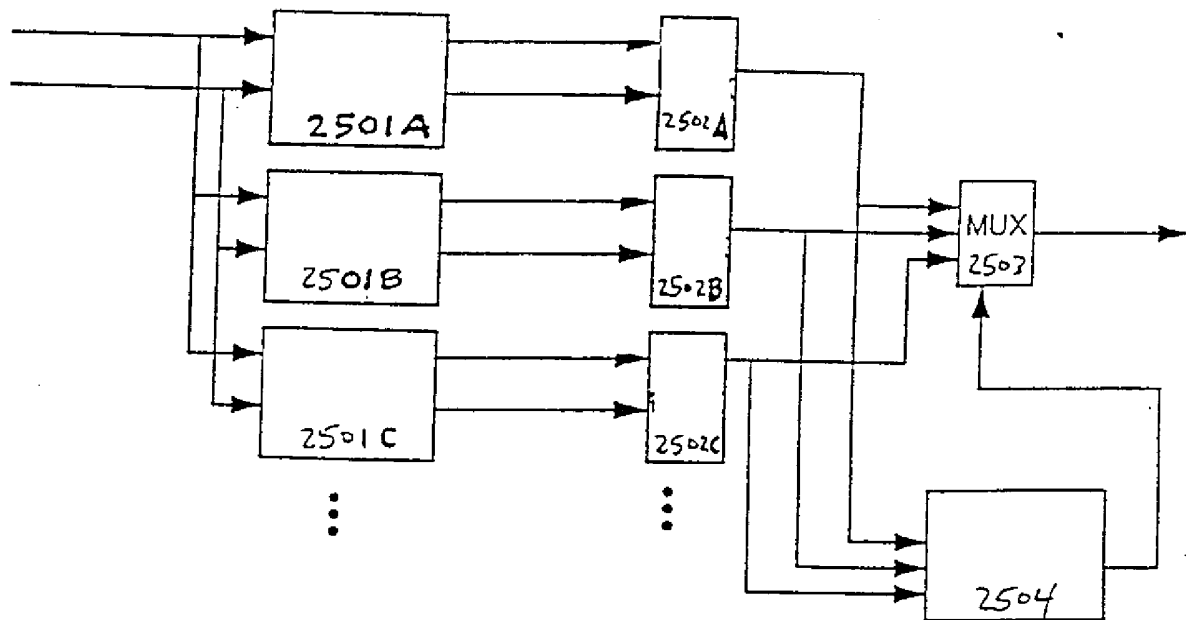
第 22 圖



第23圖

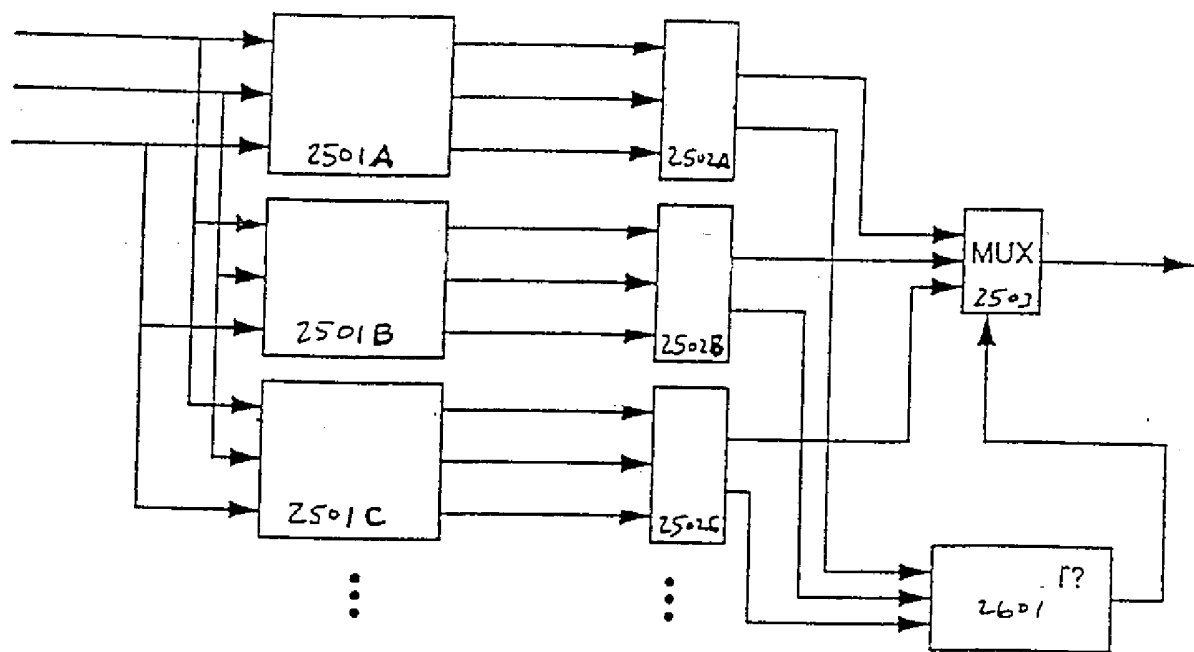


第24圖

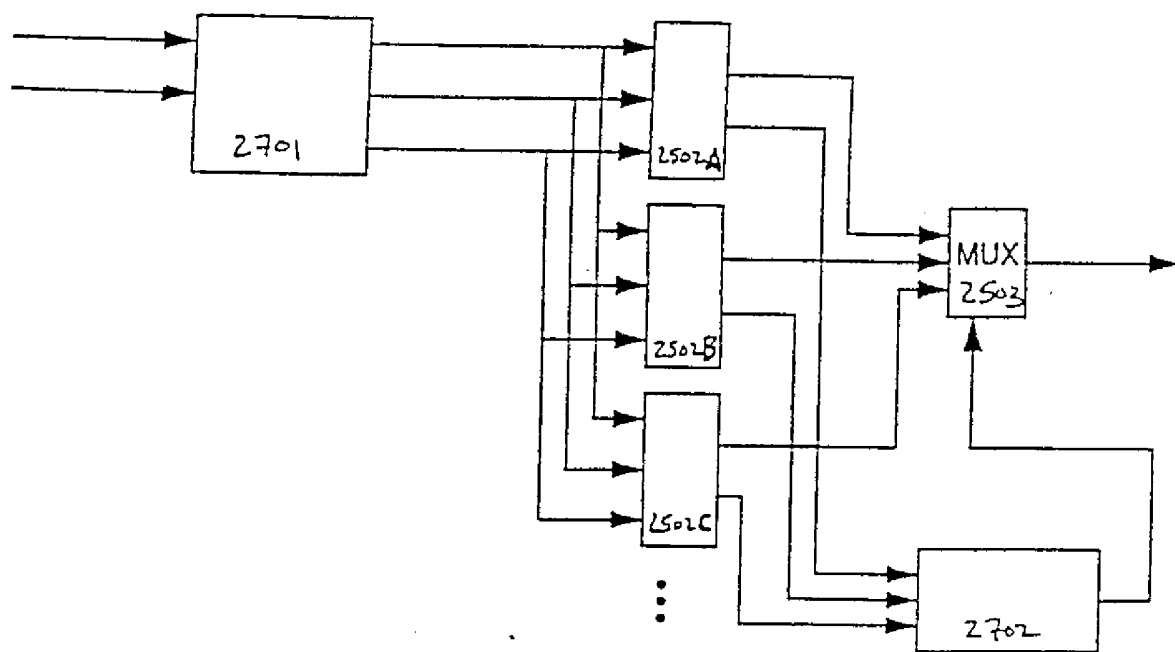


第25圖

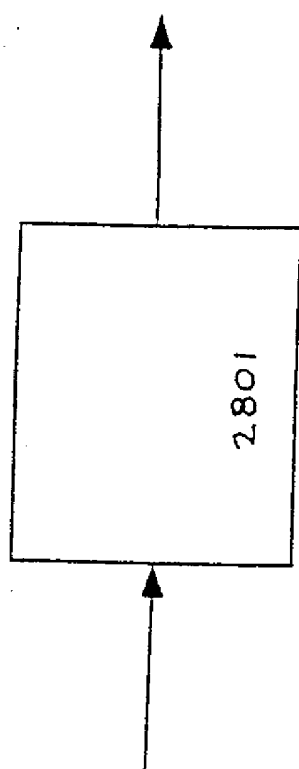
410311



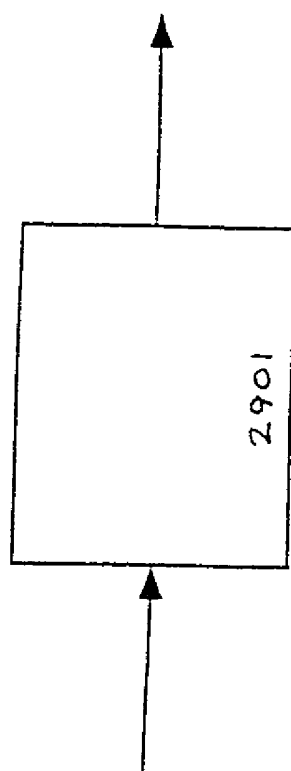
第 26 圖



第 27 圖

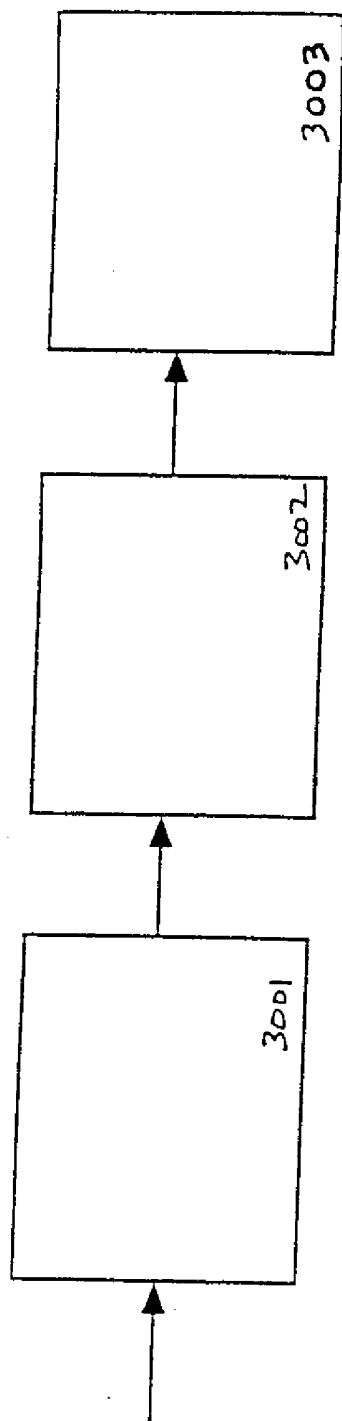


第 28 圖



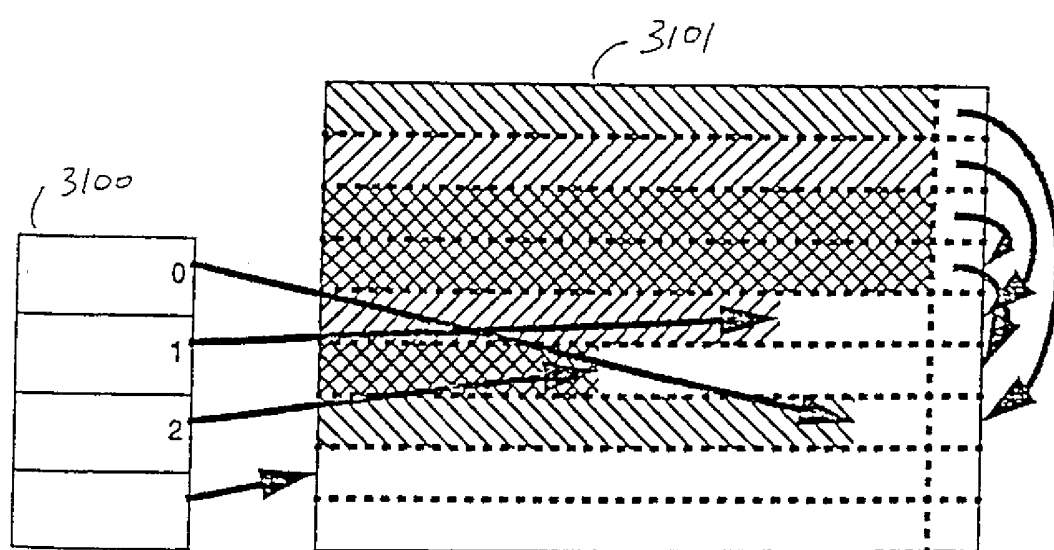
第 29 圖

410311

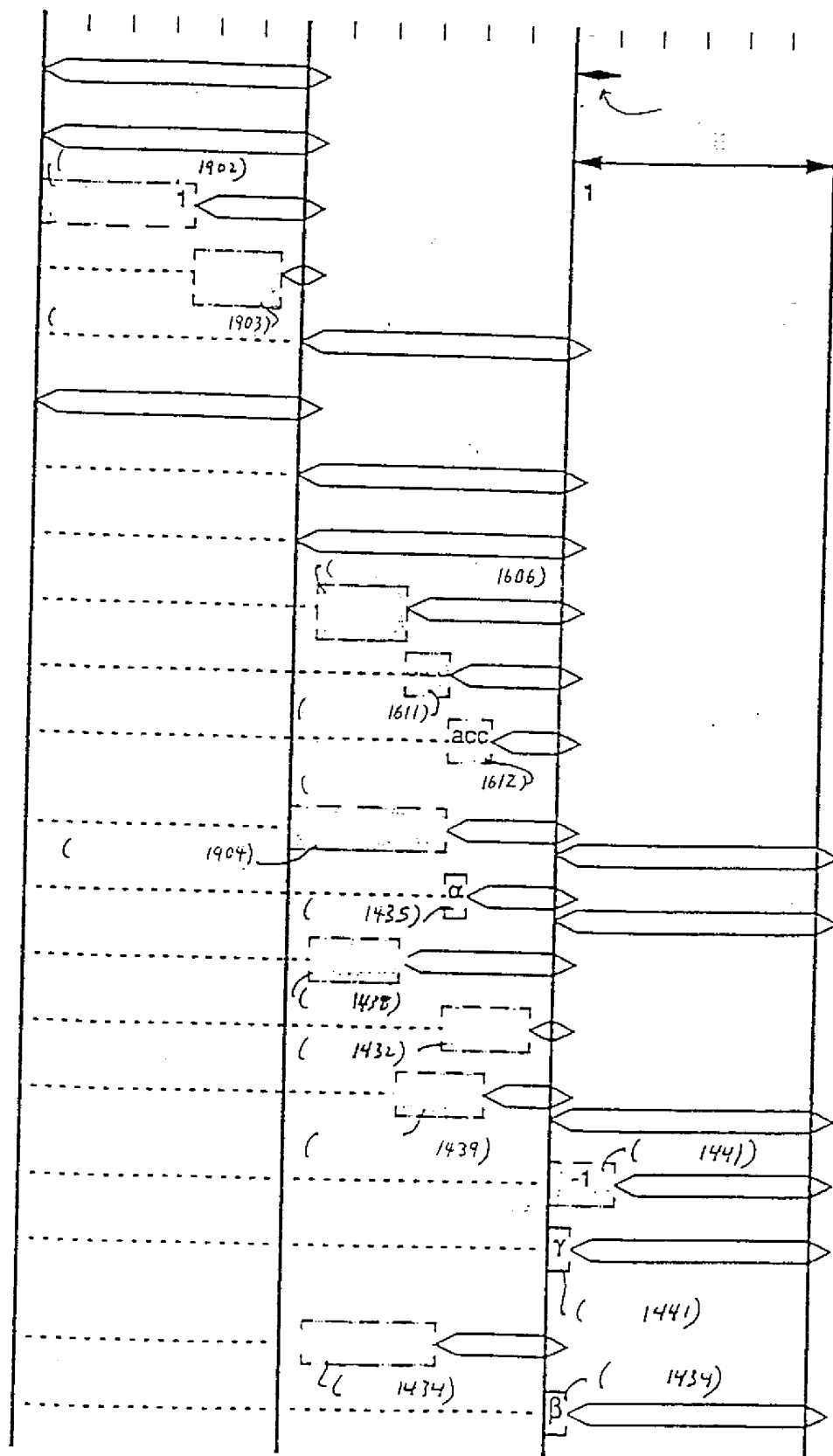


第30圖

410311

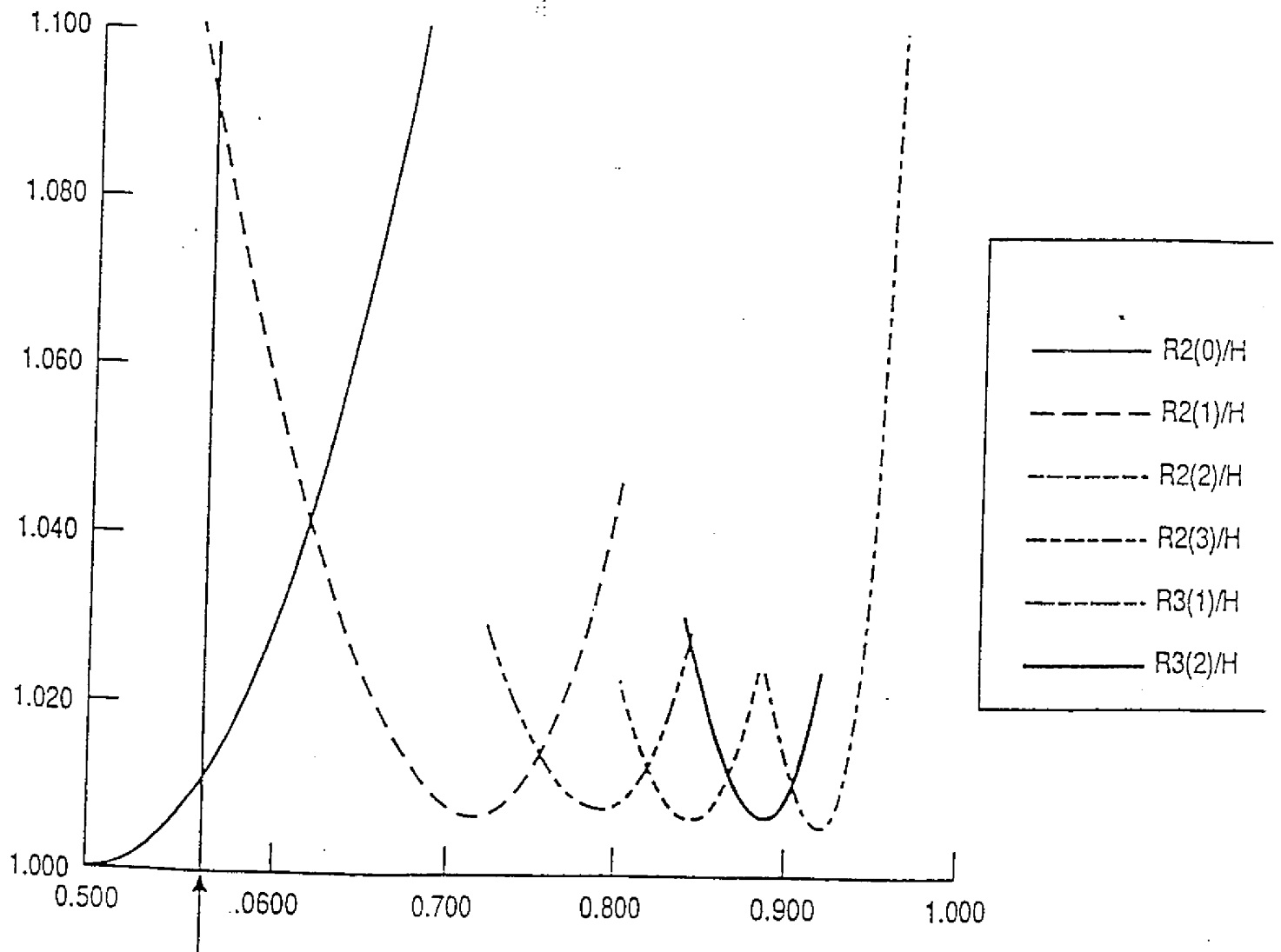


第 31 圖



第32圖

410311



第 33 圖