

公告本

申請日期	89. 8. 15
案 號	89114935
類 別	H04L12/00

A4
C4

488134

(以上各欄由本局填註)

發明專利說明書

一、發明 名稱	中 文	數位用戶線路接取多工器之有效率封包緩衝管理
	英 文	EFFICIENT PACKET BUFFER MANAGEMENT IN A DIGITAL SUBSCRIBER LINE ACCESS MULTIPLEXER
二、發明 人	姓 名	盧小林
	國 籍	中國
	住、居所	美國德州普拉諾市西達山莊路4408號
三、申請人	姓 名 (名稱)	美商德州儀器公司
	國 籍	美國
	住、居所 (事務所)	美國德州達拉斯市梅爾史特遜邱吉爾路7839號
	代 表 人 姓 名	威廉 B. 坎普樂

裝

訂

線

(由本局填寫)

承辦人代碼：
大 類：
I P C 分類：

A6
B6

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 1999 年 07 月 26 日 09/360,747 ☒有 ☐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

經濟部智慧財產局員工消費合作社印製

五、發明說明（ 1 ）

相關申請案之對照

無。

與政府贊助的研究或開發有關之聲明

無。

發明背景

本發明是屬於資料通訊之領域，尤係有關在一多通訊協定環境中對多工化多個頻道通訊之控制。

在最近幾年中，經由傳統電話網路及接線進行通訊的資料傳送速率已大幅增加。此種資料傳送速率的增加大部分是由於新採用的用來代表所傳送的訊息或資料的信號之多工及調變技術，因而使通訊頻寬大幅增加。此外，在最近幾年中，用來進行此種通訊的載波信號頻率也已增加，因而進一步提高了位元傳送速率。

係以一種在本門技術中被稱為數位用戶線路(Digital Subscriber Line；簡稱DSL)的較新型之現有數據機通訊技術獲致此種較高的資料傳送速率。DSL一般意指一種經由傳統的電話公司線路而通常在有限的距離下傳送較高的頻寬之公眾網路技術。現在已根據特定的預期資料傳送速率、傳送資料所用的媒介之類型及長度、以及用來將所傳送的資料編碼及解碼之架構，而將DSL進一步分成數種不同類型的技術。根據該技術，各DSL數據機間之資料傳送速率可能遠大於現有語音數據機的速率。事實上，現有的DSL系統係在範圍自500 Kbps(每秒千位元)到18 Mbps(每秒百萬位元)或更高的資料傳送速率下工作(或預計在此種

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (2)

狀態下工作)。根據某些傳統的技術，例如根據被稱為非對稱數位用戶線路(Asymmetric Digital Subscriber Line；簡稱ADSL)且係對應於ANSI標準T1.413的通訊協定，資料通訊速率是不對稱的。所謂的下行通訊(亦即自電話網路總局到用戶數據機)通常有較高的速率，而自用戶數據機到總局的上行通訊則有比該下行通訊速率低許多的資料傳送速率。

一DSL通訊網路中的一不可或缺部分在本門技術中被稱為數位用戶線路接取多工器(Digital Subscriber Line Access Multiplexer；簡稱DSLAM)。DSLAMs通常係位於電話網路的一總局中，且包含可連接用戶數據機的多個DSL數據機埠。DSLAM的主要功能在於將用戶下行通訊自其多個DSL數據機埠多工化到諸如可包含一伺服器及一網際網路開道器的一區域網路(LAN)等的一網路；DSLAM亦將來自該網路的返回下行通訊解多工，以便經由該等DSL埠而傳送到用戶。

通常可以多個分別支援一個或多個DSL埠的線路介面卡實現傳統的DSLAMs。在該等DSL埠上接收的通訊通常是封包形式，每一封包具有起始碼或結尾碼(或以上兩者)，而起始碼或結尾碼包含用於封包訊息的適當路由資訊。一DSLAM控制器功能執行適當的訊框控制位元插入(framing)及資料格式化，利用此種方式且根據適當的通訊協定而將在DSL埠上接收的封包傳送到DSLAM中的一網路介面卡(Network Interface Card；簡稱NIC)，並因而傳送到該網

五、發明說明 (3)

路。

在有可經由該網路而進行的範圍寬廣之網路通訊時，需要典型的DSLAMs來處理多個通訊協定。這些通訊協定包括點對點通訊協定(Point-to-Point Protocol；簡稱PPP)、網際網路通訊協定(Internet Protocol；簡稱IP)、位址決定通訊協定(Address Resolution Protocol；簡稱ARP)、簡單網路管理通訊協定(Simple Network Management Protocol；簡稱SNMP)、及其他的通訊協定。在DSLAM對自DSL埠接收的及將要經由DSL埠傳送的DSL通訊進行多工化及處理時，可能需要該DSLAM將一特定的封包格式化或重新格式化到多種此類的通訊協定。因為每一這些程序可能都要對每一封包的封包起始碼或結尾碼進行修改、添加、或刪除，所以傳統的DSLAMs工作時要重複地讀取並重新寫入包含封包的記憶體區段。

考慮到DSLAM所產生的記憶體拷貝數時，以該傳統方式進行封包處理所需的記憶體及處理資源可能是相當多的。所需的記憶體資源意指可用來儲存各種拷貝的記憶體容量，而所需的處理資源意指處理器為了要寫入及讀取該等拷貝必須執行的作業(及週期)數目。在多個DSL埠的每一DSL埠有一封包的情形中，我們可以下式來估計用來控制多個DSL埠或頻道的一DSLAM處理器所要執行的記憶體拷貝數N：

$$N = (2n + A + 1) * C$$

其中C是DSLAM處理器所控制的DSL埠或頻道數，n是

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

經濟部智慧財產局員工消費合作社印製

五、發明說明 (4)

DSLAM處理每一封包所用的通訊協定數(由於有上行通訊及下行通訊，所以n是2的倍數)，且A是DSLAM執行的多個應用程式所用之記憶體拷貝數；額外的拷貝(亦即，在該方程式中為“1”)代表自DSLAM的接收緩衝區到傳送緩衝區的一封包拷貝。縱使對於一種採用兩個通訊協定且一DSLAM應用程式無須任何額外拷貝的簡單系統而言，在該DSLAM的每一DSL頻道中，每一封包也需要有五份拷貝。考慮到一DSLAM所要處理的DSL頻道數可能相當大，且亦考慮到DSL數據機通訊的封包資料傳送速率時，此種傳統DSLAM配置所需的記憶體及處理資源變得相當大。事實上，此種傳統作業中需要的記憶體空間通常超過傳統控制處理器所能使用的晶片內建記憶體，因而在DSLAM架構內需要有一記憶體子系統。

傳統的DSLAMs所執行的線路介面卡功能現在通常使用諸如由Texas Instruments Incorporated所供應的TMS320C5x及TMS320C6x系列數位信號處理器(Digital Signal Processors；簡稱DSPs)等的DSPs，以便以高速方式處理DSLAM所傳送的資料。最好是以一類似的DSPs實施DSLAM中之控制器，以便不需要在各線路介面卡DSPs與DSLAM控制器(及NIC)之間進行資料轉換；事實上，在諸如TMS320C6x等級的現階段最高技術DSPs之情形中，我們了解能會有足夠的處理容量，因而可使用處理線路介面卡處理所用的同一個DSP執行控制功能。然而，爲了保持最高的效能，最好是使用DSPs的晶片內建記憶體來處理封包

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明(5)

通訊協定處理；而如前文所述，由於在各通訊協定之間拷貝封包需要用到大量的記憶體資源，所以根據傳統的實施方式時，將無法將DSP晶片內建記憶體用於此種功能。

發明概述

因此，本發明之一目的在於提供一種數位用戶線路接取多工器(DSLAM)、及操作該DSLAM的方法，其中係將資料封包的通訊協定轉換所需之記憶體及處理資源減至最少。

本發明之另一目的在於提供此種DSLAM及方法，而根據該方法，可於通訊協定轉換期間將高速晶片內建記憶體用於儲存所傳送的封包。

本發明之又一目的在於提供此種DSLAM及方法，而根據該方法，可利用一數位信號處理器(DSP)實現DSLAM控制器的功能。

本發明之又一目的在於提供此種DSLAM及方法，而根據該方法，可以一種與DSP或其他控制器一致的方式執行晶片內建記憶體的分配。

本發明之又一目的在於提供此種DSLAM及方法，而根據該方法，可避免封包記憶體資源的分割。

熟悉本門技術者若參照下文的說明書並配合各圖示，將可以於了解本發明的其他目的及優點。

可利用一數位用戶線路接取多工器(DSLAM)內的一封包緩衝管理器功能實施本發明。該封包緩衝管理器分配可用來儲存資料封包及起始碼的一連續記憶體區段。不論是自網路介面或自一DSL埠接收資料封包，都可將每一封包儲

五、發明說明 (6)

存在一資料結構中的記憶體區段中，而該資料結構不只是爲了所傳送的資料而保留足夠的空間，也爲了預期將應用於該封包的每一通訊協定之未來起始碼及結尾碼而保留足夠的空間。該資料結構中的一資料偏移指標指向實際封包資料的開始處。參照其中包含資料指標的各種指標，即可對封包進行控管資料(起始碼及結尾碼)的添加、修改、及刪除，並不需要拷貝資料封包本身。於傳輸時，空出該區段所佔用的記憶體部分，而恢復連續區段，以避免分割。

附圖簡述

圖1是可用來實施本發明較佳實施例的一數位用戶線路(DSL)通訊網路之電氣方塊圖。

圖2是根據本發明較佳實施例而建構的一數位用戶線路接取多工器(DSLAM)之電氣方塊圖。

圖3是可用來實施本發明較佳實施例的DSLAM控制器的一數位信號處理器(DSP)架構之電氣方塊圖。

圖4是可根據本發明較佳實施例而操作圖2所示DSLAM的軟體架構之方塊圖。

圖5是在根據本發明較佳實施例的圖2所示DSLAM控制器的晶片內建記憶體中一記憶體共用區配置之記憶體對映圖。

圖6是根據本發明較佳實施例的DSLAM控制器作業之流程圖。

圖7是根據本發明較佳實施例的DSLAM控制器的一應用之方塊圖。

五、發明說明（ 7 ）

圖 8 a 到 8 c 是根據本發明較佳實施例的圖 5 所示記憶體共用區中封包緩衝區資料項實例的位置之記憶體對映圖。

圖 9 a 到 9 e 示出根據圖 6 所示方法而於通訊協定處理期間本發明較佳實施例在記憶體共用區中的一封包緩衝區上之作業。

圖 10 是根據本發明第二較佳實施例的封包緩衝區資料項的連結配置之記憶體對映圖。

本發明之詳細說明

首先請參閱圖 1，現在將說明可實施本發明較佳實施例的一數位用戶線路(DSL)通訊網路。將參照一DSL網路而說明本發明之較佳實施例，因為我們認為當將本發明應用於DSL通訊時將特別有利；當然，我們當了解，在熟悉本門技術者參閱本說明書之後，將可易於將本發明應用於其他類型的通訊。此外，只是以舉例方式說明圖 1 所示之特定DSL網路，我們認為熟悉本門技術者在參閱本說明書之後，將可易於將本發明之效益應用於根據其他配置的DSL網路，且如前文所述，亦可應用於根據其他技術的資料通訊。

如圖 1 所示，一DSL通訊網路包含多個用戶端位置H，該等位置可對應於家庭或辦公室用戶端位置。在該DSL配置中，例示的用戶端位置H₀包含一DSL數據機2₀，用以處理進出用戶端電腦4₀之通訊。電話手機6₀係以平行於DSL數據機2₀之方式連接到分配器8₀，而在本門技術中習見的該分配器提供適當的過濾作業，以便將語音通訊傳送

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明（ 8 ）

到手機6₀，並將資料通訊傳送到DSL數據機2₀。分配器8₀係連接到其中一個本地電話網路設施TWP，而可利用傳統的雙絞銅線實施該本地電話網路設施RWP。可以類似之方式構建其他的用戶端位置H，或者可以本門技術中習知的其他方式實施其他的用戶端位置H。

圖1所示實例對應於類比語音信號以較下行通訊低的頻率傳輸因而可讓兩類的通訊同時重疊在相同的電話網路設施上之這類DSL通訊。此外，亦可使用在本門技術中被稱為“無分配器式”DSL，其中DSL數據機接收語音通訊，並將語音通訊轉換成封包通訊，且以與資料通訊類似的方式將該語音封包當作數位封包而在電話網路上傳送。亦可根據此種無分配器式技術，在相同的效益下將本發明應用於DSL通訊。

在圖1所示之實例中，以直接方式，或以混合雙絞線及光纖且配備有為了可靠的通訊而設的中繼器及其他網路元件(圖中未示出)之方式，將各設施TWP連接到總局CO。當如同在本實例中涉及語音通訊時，總局CO通常係位於電話提供者所在的場所。此外，亦可將DSL通訊限於資料通訊，在此種情形中，總局CO可以是一伺服器位置，例如用戶端位置H可與其傳收資料的公眾或專屬(網際網路或企業內網路)伺服器位置，在此種情形中，將不具有類比語音信號的能力。

如圖1所示，數位用戶線路接取多工器(DSLAM)(10)接收每一雙絞線設施TWP，而將於下文中參照本發明較佳實

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明 (9)

施例而進一步詳述 DSLAM(10)之構造。在本實例中，DSLAM(10)可將語音信號自設施TWP上接收的資料通訊中分離，並將這些信號經由亦設於總局CO的語音數位交換機(12)而傳送到公眾電話交換網路(Public Switched Telephone Network；簡稱PSTN)；當然，亦可以傳統方式由數位交換機(12)處理自PSTN到一用戶端位置H(例如在用戶端位置H₀上的手機6₀)之語音通訊。

對於資料通訊而言，如圖1所示，DSLAM(10)係與諸如在總局CO上的區域網路(LAN)(14)等的一本地網路通訊；此外，作為網際網路服務供應商上的總局實施例之典型作業，DSLAM(10)亦可與一非同步傳輸模式(Asynchronous Transfer Mode；簡稱ATM)網路通訊，而利用該ATM網路以更直接的方式與網際網路進行雙向通訊。再回到圖1所示之例子，如本門技術中所習知的，可以諸如以太網路等的傳統LANs之方式配置LAN(14)，其中可以傳統方式經由以太網路進行通訊。如大家所習知的，此種LANs係與封包通訊相容。在圖1所示之實例中，各本地伺服器(15)係設於LAN(14)上，因而用戶端位置H可經由DSL連線而自該等本地伺服器(15)擷取儲存在本地的資料。此外，尤其在電話網路應用中，可將一網際網路閘道器設於LAN(14)上，因而用戶端位置H可經由總局CO而與網際網路雙向通訊。

DSLAM(10)亦係連接到元件管理系統(Element Management System；簡稱EMS)(17)。EMS(17)是諸如一個

五、發明說明 (10)

人電腦或工作站等的一本地主電腦，可利用EMS(17)經由串列或平行介面而使用並控制DSLAM(10)內的系統管理機能。

現在請參閱圖2，現在將說明根據本發明較佳實施例的DSLAM(10)之一例示架構。如圖2所示，DSLAM(10)包含多個類比前端(Analog Front End；簡稱AFE)機能(20)，每一AFE機能(20)係作為一DSL埠，因而係雙向連接到其中一個雙絞線設施TWP，以便自一連接的用戶端位置H接收通訊。係以電路板之方式建構AFE機能(20)，這些電路板包含混合信號(亦即包含數位及類比作業)積體電路，用以提供全雙工DSL通訊所需之所有迴路介面及線路驅動器功能。可用來實現AFE機能(20)的積體電路之例子包括TLV320AD12局端ADSL編碼解碼器、及THS6002線路驅動器，這兩種積體電路都是由Texas Instruments Incorporated所供應。

DSLAM(10)的各DSL頻道收發機(22)分別雙向連接到多個AFE機能(20)。在圖2所示之實例中，每一DSL頻道收發機(22)連接到四個AFE機能(20)；當然，可經由某一DSL頻道收發機(22)處理的AFE機能(20)數及DSL頻道數可隨著特定的DSLAM架構及DSL頻道收發機(22)之處理容量而有所變化。各DSL頻道收發機(22)最好是一可程式數位裝置，用以針對載送資料的傳輸及接收而執行必要的信號處理作業。這些作業包括諸如回波消除、將資料編碼及解碼成適當的副頻道以供傳輸、以及在頻域與時域之間執

五、發明說明 (11)

行快速傅立葉變換(Fast Fourier Transform；簡稱FFT)信號處理等的功能。尤其在DSL資料傳送速率上，DSL頻道收發機(22)的數位資料處理容量及能力最好是高階的，且最好是具有至少等於Texas Instruments Incorporated供應的TMS320CL548 DSPs等的數位信號處理器所提供之處理容量及能力。每一DSL頻道收發機(22)係連接到訊框控制位元插入機能(24)，而訊框控制位元插入機能(24)最好是實施為諸如前文所述的TMS320CL548等的DSP。訊框控制位元插入機能(24)執行以適當的方式將數位資料格式化成為適當的封包及訊框，以便在DSLAM(10)內處理。雖然圖2中明確地示出兩個DSL頻道收發機(22)，但是我們當然了解，可根據DSLAM(10)的特定架構及能力，而將任何數目的DSL頻道收發機(22)連接到訊框控制位元插入機能(24)。

DSLAM控制器(25)在一端上係雙向連接到訊框控制位元插入器(24)，而在DSLAM控制器(25)的另一端上係雙向連接到以太網路介面卡NIC(26)、RS422高速串列介面(27)、及RS232串列介面(28)。以太網路NIC(26)將DSLAM(10)連接到圖1所示的LAN(14)；串列介面(27)、(28)將DSLAM(10)連接到亦如圖1所示的EMS(17)。就這一點而言，DSLAM控制器(25)根據圖2所示例示架構而執行的一種功能即是：回應經由串列介面(27)、(28)自EMS(17)接收的(或替代性地經由以太網路NIC(26)自LAN(14)接收的)控制指令，而在當地控制DSLAM(10)之設定及作業。在將資料傳

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

經濟部智慧財產局員工消費合作社印製

五、發明說明 (12)

送進出該等DSL埠時，DSLAM控制器(25)執行下列功能：處理連接到DSLAM(10)中的各AFE機能(20)的該等DSL頻道之資料流控制及頻道管理；以及處理該等DSL頻道與NIC(26)間之資料流控制。如將於下文中詳細說明的，DSLAM控制器(25)亦根據諸如PPP、IP路由、及ARP等的各種通訊協定，而執行第2層及第3層的網路通訊協定處理。

根據本發明之較佳實施例，可將DSLAM控制器(25)實施為前所述DSLAM控制器機能專用的一單一DSP裝置，該DSP裝置具有諸如由前文所述Texas Instruments Incorporated所供應的TMS320CL548 DSP所具有的能力。就這一點而言，用來作為DSLAM控制器(25)的此種DSP裝置最好是具有晶片內建的隨機存取記憶體(Random Access Memory；簡稱RAM)，例如前文所述TMS320CL548 DSP的情形。此外，在將諸如由Texas Instruments Incorporated所供應的TMS320C6x等級DSP等具有較高處理能力的DSP用於DSLAM(10)中的其他機能(亦即訊框控制位元插入機能(24)及DSL頻道收發機(22))之情形中，且如果該DSP中還有足夠的處理容量來處理DSLAM控制器(25)之機能時，則可在一單一DSP裝置中實施DSLAM(10)的數位機能。因此，根據特定電路的處理能力的不同狀況，圖2所示DSLAM(10)的機能界限可能或可能不代表用來實施這些機能的積體電路之界限。

此外，當然可利用專用於將於下文中說明的資料流控制

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

經濟部智慧財產局員工消費合作社印製

五、發明說明 (13)

作業之客戶訂製邏輯電路來實施DSLAM控制器(25)。

如前文所述，最好是將DSLAM控制器(25)實施為一數位信號處理器(DSP)，且該DSPs最好是與在DSLAM(10)內用來作為DSL頻道收發機(22)及訊框控制位元插入機能(24)的其他DSP相容。以DSP實施DSLAM控制器(25)時，可在無須資料轉換及其他應考慮的介面之情形下，使資料流經DSLAM(10)，而將DSLAM控制器(25)實施為一般用途微處理器時，可能需要該資料轉換及其他應考慮的介面。此外，將DSLAM控制器(25)實施為一DSP時，也可將DSLAM控制器(25)作業的相同開發平台用於DSLAM(10)內的其他DSPs。當然，熟悉本門技術者在參閱本說明書之後當可了解，縱使利用非DSP裝置的此種微處理器或其他硬體實施DSLAM控制器(25)，也可實現本發明的許多效益。

現在請參閱圖3，現在將詳細說明可用來實施DSLAM控制器(25)的DSP(130)之一例示架構。該DSP(130)實例對應於前文所述TMS320CL548之架構，我們當了解，亦可替代性地使用根據其他架構而建構之DSPs。

可利用一經過修改的Harvard架構實施本實例中之DSP(130)，DSP(130)因而採用分別連接到多個執行單元的三個獨立資料匯流排C、D、E，這些執行單元包括指數單元(132)、乘法/加法單元(134)、算術邏輯單元(Arithmetic Logic Unit；簡稱ALU)(136)、筒形移位器(138)。累積器(140)可讓乘法/加法單元(134)與ALU(136)平行運算，而可同時執行乘法-累積(MAC)及算術運算。在該實例中可由

五、發明說明 (14)

DSP(130)執行的指令集包括單一指令重複及區塊重複作業、區塊記憶體移動指令、兩個及三個運算元讀取、有條件的儲存作業、平行載入及儲存作業、以及專用數位信號處理指令。DSP(130)亦包含耦合到資料匯流排E之比較、選擇、及儲存單元(Compare, Select, and Store Unit；簡稱CSSU)(142)，用以加速用於許多傳統通訊演算法之Viterbi計算。

本實例中之DSP(130)包含大量的晶片內建記憶體資源，而記憶體/周邊裝置介面單元(145)經由資料匯流排C、D、E、及程式匯流排P而控制對晶片內建記憶體資源的使用。這些晶片內建記憶體資源包括隨機存取記憶體(RAM)(144)、用於儲存程式指令之唯讀記憶體(Read-Only Memory；簡稱ROM)(146)、以及若干位址暫存器(148)。程式控制器及位址產生器電路(149)亦連接到記憶體/周邊裝置介面單元(145)，並經由記憶體/周邊裝置介面單元(145)自ROM(146)或其他儲存裝置接收程式指令碼，且產生將施加到DSP(130)的每一執行單元之控制信號，以便控制對應於所接收程式指令碼的指令之執行。介面單元(158)亦連接到記憶體/周邊裝置介面單元(145)，以便控制外部通訊。電源、時脈、及控制機能(150)一般即是DSP(130)內的控制電路，用以處理諸如電源配送及穩壓、時脈產生、及對DSP(130)的整體控制等的功能；此外，電源、時脈、及控制機能(150)可進一步實施其他的介面機能，例如串列及主機埠、以及定時器、JTAG測試介面電路、及內建自我測

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

經濟部智慧財產局員工消費合作社印製

五、發明說明 (15)

試電路等的額外控制功能。

如前文所述，DSLAM控制器(25)執行在傳輸及接收方向上的資料封包之網路通訊協定處理。現在請參閱圖4，現在將說明利用根據本發明較佳實施例的DSP(130)實施之DSLAM控制器(25)軟體架構。

圖4所示之架構係基於即時核心(30)，即時核心(30)包含作為DSLAM控制器(25)軟體的其餘部分的基礎之若干功能元件。這些元件包含根據本發明較佳實施例的一即時排程器、一標準I/O管理系統、及一封包導向記憶體管理系統。

根據本發明較佳實施例的即時核心(30)之即時排程器部分使用基於優先順序的事件驅動式依序循環執行排程，以便處理在DSLAM控制器(25)的控制下通過DSLAM(10)之非同步網路介面資料通訊及訊息傳輸。在該實例中，係以一事件描述碼資料結構描述每一事件，該資料結構指示事件的類型及優先順序、特定事件之通知值、旗標、狀態、與裝置有關的資訊、及與事件相關聯的埠。

與標準I/O管理系統相關的硬體抽象層(31)包含DSLAM控制器(25)所回應的每一介面機能的驅動程式軟體之與硬體相關之部分。最好是如同將於下文中進一步詳述的可動態堆疊式通訊協定，而根據一標準I/O模型開發AFE機能(20)(圖2)支援的DSL頻道、經由NIC(26)的以太網路介面、及串列埠介面(27)、(28)之裝置驅動程式。如圖4所示，應用程式介面(Application Program Interface；簡稱

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明 (16)

API)層(32)位於硬體抽象層(31)“之上”，且包含EMS埠I/O驅動程式(32a)、DSL頻道I/O驅動程式(32b)、以太網路I/O驅動程式(32c)、及HSSP(高速串列埠)I/O驅動程式(32d)。API(32)中之這些驅動程式係作為由圖4所示架構中的信號通訊協定堆疊(36s)及資料通訊協定堆疊(36d)所代表的該架構中的通訊協定層之介面。

根據圖4所示之該軟體架構，可利用一特定應用程式或一可堆疊通訊協定驅動程式之方式實施一特定的網路通訊協定。提供較佳效能但較少彈性之網路通訊協定實施例是一較佳之網路通訊協定，但前提是該通訊協定必須在諸如PPP通訊的NPL階段等的幾乎每一資料封包上執行。

圖4所示架構的信號通訊協定堆疊(36s)是一種主要用於啟動在一用戶端位置H(圖1)上的一DSL數據機2與DSLAM(10)間的通訊期間之通訊協定堆疊。一旦建立了一雙向頻道之後，即將資料通訊協定堆疊(36d)用於儲存並處理資料封包及與資料封包相關聯的控管資料(亦即起始碼及結尾碼)，其中情形將於下文中說明之。

在通訊協定堆疊層(36s)、(36d)之上，各種系統管理程式軟體係位於圖4所示架構之API(38)。網路管理程式(38a)、主埠管理程式(38b)、及DSL頻道管理程式(38c)分別將一制式的API提供給DSLAM控制器(25)中最上層的應用程式，這些最上層的應用程式包括一EMS代理程式、DSL頻道IP路由、DSL頻道PPP處理、ARP及SNMP通訊協定等。如前文所述，管理程式機能(38a)、(38b)、(38c)全都共

五、發明說明 (17)

用標準I/O管理機能之共同程式碼庫，且同時提供與特定裝置相關的管理機能。

在作業中，經由DSLAM控制器(25)而接收及重新傳輸之資料封包有效地通過圖4所示之軟體架構。例如，硬體抽象層(31)接收一個AFE機能(20)(圖2)所接收的一DSL訊息，作為一組封包，然後由DSL頻道I/O驅動程式(32b)處理該等封包，並將該等封包放到資料通訊協定堆疊(36d)。根據各種通訊協定(而由封包緩衝區管理程式(40))處理資料通訊協定堆疊(36d)中之封包(將在下文中說明封包緩衝區管理程式(40))，然後將該等封包遞交到API(38)，以便根據訊息的特定目的地而由相關聯的管理程式處理。一旦適當的管理程式機能(38a)、(38b)、(38c)執行其處理之後，然後將該封包經由資料通訊協定堆疊(36d)傳送回封包目的地之適當I/O驅動程式(32)。

如圖4所示且如前文所述，DSLAM控制器(25)之軟體架構包含封包緩衝區管理程式(40)。封包緩衝區管理程式(40)對應於一組程式指令，當作為前為所述DSLAM控制器(25)的DSP(130)執行該組程式指令時，封包緩衝區管理程式(40)即根據本發明而針對諸如DSLAM(10)等的多通訊協定且多頻道的DSL通訊系統執行一零拷貝封包導向記憶體管理系統。封包緩衝區管理程式(40)實現的該零拷貝系統可讓DSLAM控制器(25)經由一系列不同的網路介面、通訊協定、及相關聯的處理而處理一資料區段，並同時維護置於記憶體的資料區段之內容。

五、發明說明 (18)

請再參閱圖3，最好勢將封包緩衝區管理程式(40)的程式指令儲存在用來實施DSLAM控制器(25)的DSP(130)之ROM(146)中，且經由存取DSP(130)的晶片內建RAM(144)而執行該等程式指令。因為封包緩衝區管理程式(40)執行的零拷貝記憶體管理所造成的記憶體需求減少，所以可根據本發明較佳實施例而將RAM(144)用來儲存資料封包(及其相關聯的控管資料)。因此，不只是可將記憶體需求降至最低，而且使用高速晶片內建RAM(144)時，可根據本發明而在DSLAM控制器(25)的控制下使資料流有更佳的效能。

根據本發明的較佳實施例，(於啟動DSLAM控制器(25)時)預先將一部分的RAM(144)分配為一記憶體共用區，以便於執行時供封包緩衝區使用。圖5示出此種預先分配的記憶體共用區(50)之一實例。記憶體共用區(50)最好是RAM(144)中的一連續記憶體區段，而其容量則是取決於諸如DSLAM(10)的頻道容量等的特定應用需求。也最好是將記憶體共用區(50)組織為一整數數目的連續最小容量之記憶體區段(50_j)。最小容量的記憶體區段之容量也取決於DSLAM控制器(25)所服務的特定應用，且可對應於所需應用的一最小或平均封包大小。記憶體區段(50_j)的容量最好是小於平均封包大小但大於最小封包大小，且可對應於二的乘冪倍數的資料字組。例如，當以太網路訊息的最小封包大小為62位元組時，可將最小容量的記憶體區段(50_j)設定為64位元組(稍微大於該最小封包大小，但小於平均封

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

經濟部智慧財產局員工消費合作社印製

五、發明說明 (19)

包大小，且具有一個二的乘幕的區段間偏移量)。我們認為用來選擇區段(50_j)的最小容量之精確度不是重要的；因此，諸如在前文所述64位元組的最小容量中，可能需要犧牲記憶體的使用，以便減少處理器作業及週期的數目。此外，如將於下文中說明的，將以一資料字組為單位而存取每一區段(50_j)，而字組的寬度係對應於用來實施DSLAM控制器(25)的DSP(130)之架構。

如圖5所示，記憶體共用區(50)具有一儲存在DSP(130)的一整體變數儲存區之起始位址資料庫，以使用於實體記憶體位址之計算。一旦分配了記憶體共用區(50)之後，最好是保護記憶體共用區(50)，使其他的應用不得使用該記憶體共用區(50)。

不論DSLAM控制器(25)根據本發明較佳實施例而處理的是來源資料或通訊協定，都利用一共用的資料結構P_BUF而將每一訊息資料封包儲存在封包記憶體共用區(50)中，其中係針對有效率地儲存可變長度的資料封包而特別設計該P_BUF。資料結構P_BUF的配置容許其(多個訊息封包的)多個實例在記憶體共用區(50)內進行有效率的等候、分配、及解除分配。根據本發明的一較佳實施例，可如下文(遵循一C語言類型定義)而說明封包緩衝區資料結構P_BUF：

```
Typedef_P_BUF
    Q_ENTRY   Pkt_Q;           /*封包佇列資料項*/
    WORD      B_Size;          /*封包緩衝區總容量*/
```

五、發明說明 (20)

```
WORD    D_Offset;          /* 資料起始偏移量 */  
WORD    D_Size;            /* 總有效資料大小 */  
WORD    D_Flag;            /* 資料旗標 */  
} P_BUF, *P_BUF
```

一特定封包緩衝區資料結構的名稱 *pPktBuf 將在其起始時被指定。

資料項結構 Pkt_Q 是由 Q_ENTRY 定義的佇列資料項結構，包含一特定資料封包緩衝區的順向及反向指標。這些順向及反向指標分別指向現行資料封包緩衝區的相鄰起始及結尾資料封包緩衝區，而有助於該等封包緩衝區之放入佇列及取出佇列作業。

封包資料緩衝區結構的每一其餘欄位之 WORD 巨集描述該資料結構中的實際機械字組。在本發明的該第一實施例中，欄位 B_Size 指示分配給封包緩衝區而其中包括實際資料及保留給潛在通訊協定的預期起始碼及結尾碼的記憶體空間之資料字組總數。如前文所述，根據本發明較佳實施例而使用的資料字組或區段大小最好是匹配用來實施 DSLAM 控制器 (25) 的 DSP (130) 之自然資料字組大小。例如，在前文所述的 TMS320CL548 DSP 中，資料字組的大小為十六位元，或兩個位元組；因此，在欄位 B_Size 中 (及封包緩衝區資料結構的其他欄位中) 表示的資料字組數將指定十六位元資料字組之數目。欄位 B_Size 指示的記憶體大小應為上述最小區段大小的一整數倍，以便能得到本發明的完整效益。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

線

五、發明說明 (21)

欄位 D_Offset 指示自資料封包緩衝區的起始位址到該封包緩衝區中實際資料的第一有效字組之字組數。如將於下文說明中解說的，調整該欄位 D_Offset 的值，以便代表資料中將由一特定通訊協定考慮的部分。例如，只要調整欄位 D_Offset 的值，而不再將所要刪除的一起始碼包含作為其有效資料的一部分，即可根據本發明較佳實施例而完成該起始碼之刪除；相反地，改變欄位 D_Offset 值而容許額外的有效資料，然後將所需的起始碼插入該新區域，即可加入一起始碼。因此，根據本發明的較佳實施例，修改欄位 D_Offset 的值時，即可在不需要拷貝記憶體中的封包之情形下修改起始碼資料。當欄位 D_Size 指定資料封包緩衝區中以資料字組為單位的有效資料大小時，該欄位係配合欄位 D_Offset 而作業。因此，當加入或刪除起始碼時，係連同欄位 D_Offset 值的修改而修改欄位 D_Size 的值；此外，可以類似於前文中對欄位 D_Offset 所述之方式，經由欄位 D_Size 值的修改，而加入或刪除實際資料之結尾碼。

在根據本發明較佳實施例的資料封包緩衝區結構中設有欄位 D_Flag，用以指示欄位 D_Size 指示的有效資料的最後一個資料字組中之有效資料位元組數目。如前文所述，通常係將訊息封包組織成位元組，而 DSLAM 控制器 (25) 中 DSP (130) 所用的資料字組或區段之寬度為多個位元組；因此，根據本發明較佳實施例而組織的資料封包通常在不與資料封包相關聯的結尾資料字組中留有空的位元組，而如果傳送這些結尾資料字組作為資料封包的一部分，則可能

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

經濟部智慧財產局員工消費合作社印製

五、發明說明 (22)

產生一錯誤。欄位 D_Flag 的值因而指示資料封包緩衝區的結尾資料字組中之有效位元組數。我們考慮到，為了得到最佳的效率，欄位 D_Flag 中儲存的實際值將等於非有效資料位元組之數目。例如，如果一資料封包緩衝區的最後十六位元資料字組中只有一個位元組是有效的，則其欄位 D_Flag 的值將為 1；相反地，如果兩個位元組都是有效的，則欄位 D_Flag 之值將為 0。對於三十二位元的資料字組而言，欄位 D_Flag 的可能值將為 0 到 3 (分別對應於四到一個位元組是有效的)。

根據本發明的第一較佳實施例，封包緩衝區管理程式 (40) (圖 4) 包含在 API 層 (38) 上的一函式庫。這些函式執行封包緩衝區分配及解除分配之機能；其他的封包緩衝區作業意義亦係包含在該函式庫中。根據本發明第一實施例的這些函式庫機能之一份清單係如下文：

P_Alloc (WORD Size, WORD Offset)：該函式分配一部分的記憶體共用區 (50)，以供一封包緩衝區使用。運算元 Size 指定封包緩衝區的容量，且該緩衝區的位置是在距離位址 BASE 的開始處有偏移量 Offset 處。將一封包緩衝區的內容寫入所分配的該記憶體共用區 (50) 時，係將實際封包緩衝區的內容儲存在所界定封包緩衝區中欄位 B_Size (該欄位的值將是記憶體共用區 (50) 中的最小容量記憶體區段之一倍數) 指定的一部分。

P_Free (P_BUF *pPktBuf)：該函式將所分配的封包緩衝區內容送回到記憶體共用區 (50)。如將於下文中說明

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明 (23)

的，也將執行因執行該函式而造成的未使用的記憶體區段之任何合併，以便減少分割。

P_Append (P_BUF *pPktBuf, WORD *Buf, SIGNED WORD Count)：該函式將*Buf指定的資料添加到所指定封包緩衝區*pPktBuf的開始處或結尾處(由運算元Count的正負號決定)。

P_Remove (P_BUF *pPktBuf, SIGNED WORD Count)：該函式自封包緩衝器的開始處或結尾處(由Count的正負號指定)移除由運算元Count的大小所指示的資料量。

P_Catenate (P_BUF *pPktBuf1, P_BUF *pPktBuf2)：該函式連結指定的封包緩衝區*pPktBuf1及*pPktBuf2。

P_BUF*NetPktRead (IO_CHAN *pIODesc)：該函式是一API層級的函式，可讓工作中的應用程式擷取所指定裝置I/O頻道驅動程式*pIODesc使用的記憶體共用區(50)中的封包緩衝區*pPktBuf之一指標；當封包緩衝區*pPktBuf的內容通過資料通訊協定堆疊(36d)時，該讀取函式是用來識別該封包緩衝區之位置，此時封包緩衝區管理程式(40)根據所介入的通訊協定而修改或刪除該封包的起始碼及結尾碼，而留下呼叫本函式的應用程式所需之訊息資料。

NetPktWrite (IO_CHAN, *pIODesc, P_BUF *pPktBuf)：該函式也是一API層級的函式，用以將自NetPktRead函式送回的封包緩衝區*pPktBuf內容經由資料通訊協定堆疊(36d)而寫入所指定I/O頻道驅動程式*pIODesc之

五、發明說明 (24)

傳輸佇列。

這些函式可經由 DSLAM 控制器 (25) 而對訊息資料進行多個通訊協定的處理，而無須將資料本身拷貝到記憶體中。

現在請參閱圖 6 及 7，現在將詳細說明根據本發明第一較佳實施例的 DSLAM 控制器 (25) 之作業。圖 6 以流程圖的形式示出根據本發明該第一實施例而使用的各方法步驟，而流程控制程序的一圖示實例則係示於圖 7。如圖 7 所示，係參照一封包緩衝區 *PB₂ 而示出根據本發明的 DSLAM 控制器 (25) 所執行之一例示工作，該封包緩衝區 *PB₂ 內容係儲存在記憶體共用區 (50)，作為一接收佇列 RxQ 的一部分，且經由兩個通訊協定 (通訊協定 1 及通訊協定 2) 而處理該封包緩衝區 *PB₂ 內容，而且由一應用程式操作該封包緩衝區 *PB₂ 內容；然後經由通訊協定 2 及 1 重新處理封包緩衝區 *PB₂ 內容，並將該封包緩衝區 *PB₂ 內容傳送到一傳輸佇列 TxQ。接收訊息資料封包的 I/O 裝置通常將不同於傳送該訊息的 I/O 裝置；針對上行 DSL 通訊之例子，係由一個 DSL I/O (AFE(20)) 頻道接收該訊息資料封包，並將該訊息資料封包重新經由以太網路 NIC(26) 傳送到 LAN(14)。不論在任何狀況中，如圖 7 所示，封包緩衝區 *PB₂ 在整個該處理中都保持在記憶體共用區 (50) 中的相同位置。

請再參閱圖 6，在程序 (52) 中，DSLAM 控制器 (25) 的一個 I/O 頻道接收一訊息資料封包。如前文參照圖 2 及 4 所述，DSLAM 控制器 (25) 可用來接收一訊息資料封包的該組可能之 I/O 頻道包含 DSLAM(10) 服務的每一 DSL 頻道、經由串

五、發明說明 (25)

列介面(27)、(28)而耦合到EMS(17)的各串列埠、以及以太網路NIC(26)服務的以太網路I/O頻道。圖4中將這些I/O頻道的對應之驅動程式示為EMS埠I/O驅動程式(32a)、DSL頻道I/O驅動程式(32b)、以太網路I/O驅動程式(32c)、及高速串列埠I/O驅動程式(32d)。在程序(54)中，接收機驅動程式(32)分配記憶體共用區(50)中的一封包緩衝區，作為其接收佇列的一部分；所分配的該封包緩衝區*PB(圖7)將被用來儲存所接收的封包資料、以及在DSLAM控制器(25)所進行的整個通訊協定處理及傳送過程中所接收及處理的且並未被重新拷貝的該封包資料之起始碼及結尾碼。

圖8a及8b示出根據本發明第一較佳實施例的分配程序(54)之一實例。圖8a示出諸如在DSP(130)的晶片內建RAM(144)中的記憶體共用區(50)之配置，其中先前已分配了封包緩衝區*PB₀、*PB₁。如圖8a所示，封包緩衝區*PB₀、*PB₁係位於記憶體共用區(50)的終端(在自位址BASE算起的最遠偏移量處)；該配置當然只是舉例。在程序(54)中，接收驅動程式(32)為其接收的訊息資料封包執行函式庫函式P_Alloc (WORD Size, WORD Offset)，而選擇足以儲存實際的訊息資料以及可能在圖6所示處理過程中增加或修改的任何起始碼之一運算元Size值(形式為最小容量記憶體區段(50_j)的一整數)；然後運算元Offset之值係基於先前分配的封包緩衝區*PB₀、*PB₁之現行位置、以及所接收封包之運算元Size值。圖8b示出分配記憶體共用區

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

象

五、發明說明 (26)

以便容納封包緩衝區*PB₂的情形。在該實例中，係選擇運算元Size的值為十一個最小容量記憶體區段(50_j)，且將運算元Offset的值設定為小於十一(亦即在該實例中的運算元Size值)的封包緩衝區*PB₁運算元Offset值。由於適當的I/O頻道驅動程式(32)執行的程序(54)，而在記憶體共用區(50)內分配封包緩衝區*PB₂。

圖9a示出在分配之後所分配封包緩衝區*PB₂的一實例之內容。如圖9a的實例所示，函式庫函式P_Alloc (WORD Size, Word Offset)中之運算元Size值指定所分配封包緩衝區*PB₂之整體容量。圖9a中標示為“載送資料”(“PAYLOAD”)之實際的訊息資料係位於封包緩衝區*PB₂之內，而封包緩衝區*PB₂的額外保留部分係在載送資料之上及之下。該實際的訊息資料不只是包含訊息資料，也可包含最後必須要傳送的此種起始碼資訊(例如PPP封包起始碼資料)。欄位D_Offset的值指示此時封包緩衝區*PB₂內實際資料PAYLOAD開始的位置，而欄位D_Size的值被設定為此時在封包緩衝區*PB₂內儲存的實際資料之大小。此外，在分配之後，且一旦建立封包緩衝區*PB₂的欄位D_Offset及D_Size值之後，於I/O驅動程式(32)將資料儲存到記憶體共用區(50)作為一接收佇列時，將設定欄位D_Flag，用以指示實際資料的最後一個資料字組中之有效資料位元組數。將以與封包緩衝區*PB₂相關聯的方式，在DSLAM控制器(25)內的一本機儲存裝置或暫存器中維護這些欄位D_Offset、D_Size、及D_Flag。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明 (27)

如程序(56)所示，對封包緩衝區*PB₂進行作業的應用程式(38)開始執行前文所述函式庫函式NetPktRead(IO_CHAN *pIODesc)的一實例。函式NetPktRead的該實例指定先前在程序(52)、(54)中接收封包及所分配的封包緩衝區*PB₂之I/O頻道驅動程式(32)，且亦接收一個用來指示被界定及分配的封包緩衝區*PB₂之指標。且亦作為程序(56)的該起始之一部分，將一通訊協定指標i設定為序列中的第一通訊協定(圖7所示之通訊協定1)。執行該函式庫函式NetPktRead時，將使封包緩衝區管理程式(40)利用訊息所需各序列的通訊協定(例如圖7所示的通訊協定1及通訊協定2)處理封包緩衝區*PB₂，下文中將說明其中的情形。

NetPktRead函式的執行包括反覆地執行通訊協定處理程序(58)。在程序(58)中，DSLAM控制器(25)中之封包緩衝區管理程式(40)對所指定封包緩衝區*PB₂內容的起始碼或結尾碼部分執行作業，而無須拷貝該封包緩衝區*PB₂的整個內容；實際上，如下文中將參照圖9b而說明的，係利用上述的函式庫函式P_Append (P_BUF *pPktBuf, WORD *Buf, SIGNED WORD Count)及P_Remove (P_BUF *pPktBuf, SIGNED WORD Count)執行這些修改。

一般而言，通訊協定處理程序(58)通常包含：在不干擾實際的訊息資料本身的情形下，根據適當的通訊協定而對控管資料位元組進行增加、刪除、或修改。然後執行決定程序(59)，以便決定是否需要處理額外的通訊協定；如果確係如此(決定程序(59)的決定為“是”)，則在程序(60)中

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

經濟部智慧財產局員工消費合作社印製

五、發明說明 (28)

遞增通訊協定指標i，且控制回到程序(58)，以便進行額外的處理。

在通訊協定處理程序(58)的每一實例內，封包緩衝區管理程式(40)於執行P_BUF_NetPktRead時，可根據所要處理的現行通訊協定，而呼叫一個或多個下列函式。例如，要添加一個通訊協定起始碼時，將執行下列的函式：

P_Append (P_BUF *pPktBuf, WORD *Buf, SIGNED WORD Count)。添加一個起始碼時，需要使欄位*Buf的值等於所需的起始碼資料，且使欄位Count為正號，以便在開始處添加該起始碼。因此，指標D_Offset的值將被遞減該欄位Count值，且欄位D_Size的值將被遞增該欄位Count值。

圖9b示出於處理圖7所示實例中的通訊協定1時添加通訊協定起始碼PH1。如前文所述，係在記憶體共用區(50)中的所分配封包緩衝區*PB₂之內執行該作業，並不需要拷貝該封包本身。

亦可由封包緩衝區管理程式(40)將一結尾碼添加到現行封包緩衝區，而執行通訊協定處理程序(58)。藉由執行下列函式而執行一結尾碼的添加：

P_Append (P_BUF *pPktBuf, WORD *Buf, SIGNED WORD Count)。添加一個結尾碼時，需要使欄位*Buf的值等於所需的結尾碼資料，且使欄位Count為負號，以便在結尾處添加該結尾碼。因此，在該實例中，指標D_Offset的值將不會改變，但欄位D_Size的值將被遞增該欄位

五、發明說明 (29)

Count值。

圖9c示出於處理圖7所示實例中的通訊協定2時添加通訊協定結尾碼PT2。如前文所述，係在記憶體共用區(50)中的所分配封包緩衝區*PB₂之內執行該作業，並不需要拷貝該封包本身。如圖9c所示，封包緩衝區*PB₂包含通訊協定起始碼PH1及通訊協定結尾碼PT2。

如前文所述(且將利用下文中的實例示出)，利用利用函式P_Remove執行作為程序(58)一部分的一起始碼或結尾碼之刪除。於諸如根據位址決定通訊協定(ARP)而經常需要執行的修改一起始碼或結尾碼時，亦可在程序(58)中執行該修改，且最好是以前文所述之方式，先刪除一起始碼或結尾碼，然後將所需的起始碼或結尾碼添加到封包的開始處或結尾處。

如前文所述，係重複執行通訊協定處理程序(58)，直到沒有其他的通訊協定需要處理為止(決定程序(59)之決定為“否”)，然後如圖6中之程序(61)所示將控制轉一到應用程式式(38)，以供其以特定方式處理現行封包緩衝區。此種處理可包含單純將實際的訊息資料(載送資料)傳送到目的地I/O頻道；此外，在傳送訊息資料之前，某種預先處理可能是必要的(例如在PPP通訊的情形中)。不論在任何一種情形，應用程式式(38)將查詢封包緩衝區的欄位D_Flag之狀態，以便決定實際訊息資料的終端資料字組中之有效位元組數，以避免因在封包緩衝區儲存的過程中使用資料字組而不使用位元組產生的未預期零值位元組而造成之任何錯

五、發明說明 (30)

誤。

我們當了解，可能會有因應用程式(38)處理封包而產生的延遲，因而封包緩衝區管理程式(40)所要處理的次一封包緩衝區不是最近處理過的同一封包緩衝區(例如*PB₂)；事實上，我們考慮到在有多個應用程式(38)的情形中，DSLAM控制器(25)係同時處理來自其所服務的多個I/O頻道之訊息封包。如將於下文中詳細說明的，此種最終結果大多將直接影響到記憶體共用區(50)中各區段的空出及合併。

當應用程式(38)在程序(61)中完成其處理時，應用程式(38)執行 NetPktWrite (IO_CHAN, *pIODesc, P_BUF *pPktBuf)函式庫函式，使運算元*pIODesc參照到該封包的目的地I/O頻道，並使運算元*pPktBuf當然參照到將要傳送的特定封包緩衝區，即可開始傳送對應於所需封包緩衝區(例如*PB₂)之訊息資料。此時，係將通訊協定指標i設定為(或維持在)其最大值，而對應於上次在程序(58)中處理過的通訊協定。

於執行該傳送處理時，封包緩衝區管理程式(40)在程序(58)的反覆過程中，反覆地對封包緩衝區*PB₂執行該通訊協定處理。例如，可藉由執行下列函式而刪除一通訊協定結尾碼(例如圖9c之結尾碼PT2)：

P_Remove (P_BUF *pPktBuf, SIGNED WORD Count)：刪除一結尾碼時，需要使運算元Count的值等於所要刪除的資料字組之所需數目，且使運算元Count的值為負

五、發明說明 (31)

號，以便指示自封包緩衝器 *pPktBuf 中實際資料之結尾處移除。該作業將不會改變指標 D_Offset 的值；只須將欄位 D_Size 的值遞減欄位 Count 值，即可替代性地執行結尾碼的取消。

圖 9 d 示出該作業的結果。實際上，在程序 (64) 的該實例之後，對應於圖 9 c 所示實例中的結尾碼 PT2 之實際資料字組仍然保留在記憶體中，只是不理會該資料字組罷了。

在程序 (64) 的第一實例之後，封包緩衝區管理程式 (40) 執行決定程序 (65)，以便決定是否還有要處理的額外封包。如果確係如此 (決定程序 (65) 的決定為“是”)，則在程序 (66) 中遞減通訊協定指標 i，並將控制轉移回程序 (64)，以便進行額外的通訊協定處理。對於封包緩衝區 *PB₂ 的實例而言，執行下列函式即可刪除通訊協定起始碼 PH1：

P_Remove (P_BUF *pPktBuf, SIGNED WORD Count)：刪除一起始碼時，需要使運算元 Count 的值等於所要刪除的資料字組之所需數目，且使運算元 Count 的值為正號，以便指示自封包緩衝器中實際資料之開始處移除。指標 D_Offset 的值將遞減欄位 Count 值，欄位 D_Size 也將遞減相同的值。

圖 9 e 示出在完成程序 (64) 之該實例之後的封包緩衝區 *PB₂ 狀態。

當已經由程序 (64) 處理了所有的通訊協定時 (決定程序 (65) 之決定為“否”)，即警示應用程式 (38) 在函式庫函式 NetPktWrite (IO_CHAN, *pIODesc, P_BUF *pPktBuf) 中指定

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明 (32)

的適當I/O頻道驅動程式(32)：該I/O頻道在記憶體共用區(50)中有一與本實例中的封包緩衝區*PB₂相關聯之封包。傳出驅動程式(32)然後在程序(68)中有效地使用封包緩衝區*PB₂作為其傳輸佇列，而在適當的時間將該資料傳送到其對應的I/O頻道(例如以太網路NIC(26))。

當該適當的I/O頻道驅動程式(32)完成使用本實例中的封包緩衝區*PB₂之內容時，然後執行程序(70)，以便空出記憶體共用區(50)中與封包緩衝區*PB₂相關聯的被分配部分。執行P_Free (P_BUF *pPktBuf)函式，以便參照到所要空出的封包緩衝區，即可執行該封包緩衝區的空出。然後歸還記憶體共用區(50)中先前分配給現行封包緩衝區*PB₂的部分，以供重新分配。

根據本發明的較佳實施例，係以一種合併所空出各區段之方式，執行自記憶體共用區(50)空出一封包緩衝區，因而避免在後續的封包緩衝區分配中造成分割的情形。現在將參照圖8c而說明於執行該合併時程序(70)的一作業實例，其中在記憶體共用區(50)中仍然分配有封包緩衝區*PB₀、*PB₂，但業已空出了封包緩衝區*PB₁(請參閱圖8b)。根據本發明的較佳實施例，配合封包緩衝區*PB₂的空出，封包緩衝區管理程式(40)檢查記憶體共用區(50)的狀態，以便決定封包緩衝區*PB₂是否連續到記憶體共用區(50)中已被分配的各區段；如果並非如此(如同在圖8c中之情形)，則封包緩衝區管理程式(40)最好是改變前文所述其佇列資料項P_ENTRY Pkt_Q中封包緩衝區*PB₂內含的各

五、發明說明 (33)

指標，而合併被認定已空出的記憶體共用區(50)之各區段。在此種情形下於程序(70)中空出封包緩衝區*PB₂時，即可使記憶體共用區(50)內的各可用區段不會被分割。

因此，根據本發明的該第一較佳實施例，由於免除了拷貝訊息資料封包內容之需求，所以苦在最小的記憶體需求下完成經由一DSLAM對資料流的控制。此外，可根據本發明而節省在傳統DSLAM控制器中產生這些拷貝(事實上為多份拷貝)所需的處理作業及機器時脈週期，因而空出了額外的處理容量。此種額外的處理容量以及所得到電路消耗的減少可讓每一DSLAM線路介面卡支援更多的DSL頻道，因而提高了DSLAM的埠密度。因為較低的記憶體需求，所以亦可使用諸如作為控制器的DSP的晶片內建RAM等的晶片內建記憶體，因而進一步提高了DSLAM控制器的效能。由於根據本發明而利用DSP或作為DSLAM控制器的其他處理器所用之資料字組，而不採用傳統的訊息傳輸通訊協定之位元組導向，所以亦提高了DSLAM的處理效率。此外，由於根據本發明較佳實施例而自動合併封包緩衝區記憶體共用區中空出的各記憶體區段，而降低記憶體的分割，因而進一步提高了DSLAM控制器的效能。

現在請參閱圖10，現在將說明根據本發明第二較佳實施例的封包緩衝區*PB之邏輯配置。在處理封包時所碰到的起始碼及結尾碼之長度是未知的情形中，根據本發明該實施例的圖10所示之配置尤其有其效益。經由以此種方式使用一連結表，即可避免對記憶體共用區(50)相當大的部分

五、發明說明 (34)

作不必要的分配，且因而可避免對記憶體共用區(50)作累贅的管理。

根據本發明的第二實施例，封包緩衝區資料結構P_BUF的類型定義係如下：

```
Typedef_P_BUF
    Q_ENTRY  Pkt_Q;           /* 封包佇列資料項 */
    Q_ENTRY  P_List;          /* 封包表 */
    WORD     B_Size;          /* 封包緩衝區總容量 */
    WORD     D_Offset;        /* 資料起始偏移量 */
    WORD     D_Size;          /* 總有效資料大小 */
    WORD     D_Flag           /* 資料旗標 */
} P_BUF, *P_BUF
```

除了有佇列資料項出現以外，該封包緩衝區資料結構語前文所述的封包緩衝區資料結構P_BUF相同。根據本發明的該實施例，一單一封包緩衝區可能位於記憶體共用區(50)的數個封包緩衝區位置，而實際上係以一連結表之形式出現。佇列資料項P_List針對每一封包緩衝區分配提供了該分配所屬的連結表之一識別碼。

請再參閱圖10，圖中示出該連結。在該實例中，封包緩衝區表*PB_{0,0}到*PB_{0,2}對應於在分配時無法可靠地預測資料字組數目時(例如在無法當場知道特定序列的通訊協定時)之一單一訊息資料封包。在該實例中，該單一訊息資料封包係儲存在相互參照的三個連結封包緩衝區表資料項*PB_{0,0}到*PB_{0,2}中。因此，當以前文所示之方式執行對每一

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

經濟部智慧財產局員工消費合作社印製

五、發明說明 (35)

資料項*PB_{0,j}進行通訊協定處理時，I/O驅動程式(32)及應用程式(38)都可能重新排列封包緩衝區表資料項*PB_{0,0}到*PB_{0,2}內容的順序，以便進行傳輸或應用處理。

根據本發明的該第二實施例，記憶體共用區(50)以與其維護前文所述一序列的封包緩衝區本身之類似方式，維護一序列的封包緩衝區表。在圖10所示之實例中，連結封包緩衝區表*PB_{1,0}及*PB_{1,1}之資料項累積地對應於該序列的所分配封包緩衝區中之次一封包緩衝區資料項，但係個別地位於記憶體共用區(50)中相互不連續的區域。可將該序列中的第三封包緩衝器*PB_{2,0}儲存在記憶體共用區(50)的一單一連續區中，這是因為該封包的起始碼及結尾碼資料需求是固定的(例如在PPP封包的情形中)；根據該封包緩衝器*PB_{0,2}的資料結構因而將是無效。

根據本發明的該第二實施例，最好是提供額外的函式庫函式，用以管理記憶體共用區(50)中的連結封包緩衝區資料項表。這些函式包括：

P_AllocList (WORD Size, WORD Offset)：該函式以前文所述的一連結表之形式而分配一部分的記憶體共用區(50)，以使用來作為一封包緩衝區。因而將記憶體共用區(50)中利用該函式分配的部分保留給一組資料項。

P_FreeList (P_BUF *pPktBuf)：該函式將封包緩衝區位置的分配表送回到記憶體共用區(50)，而以前文所述之方式對空出的記憶體區段執行適當的合併，以便減少分割。

五、發明說明 (36)

如同前文中參照本發明第一較佳實施例所述的，本發明的第二較佳實施例提供了下列類似的優點：減少必要的記憶體資源、提高處理器效率、提高埠的密度、可使用晶片內建RAM以供DSLAM流程控制、以及減少了分割。然而，本發明的第二較佳實施例額外地提供了下列效益：可提供封包緩衝區部分的連結表，因而可在線上定義並處理各種通訊協定。

雖然已參照一些較佳實施例而說明了本發明，但是我們當了解，對本門技術具有一般知識者在參閱本說明書及其圖示之後，將可易於作出這些實施例的修改及替代、以及具有本發明優點及效益的此類修改及替代。我們當了解，此類修改及替代係在後文申請專利範圍的本發明範圍內。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

四、中文發明摘要 (發明之名稱：數位用戶線路接取多工器之有效率封包緩衝管理)

本發明揭示了一種用於一數位用戶線路通訊網路而處理一數位用戶線路接取多工器(DSLAM)中的訊息資料封包之方法。該DSLAM包含一DSLAM控制器，且可將該DSLAM控制器實施為一具有其本身晶片內建隨機存取記憶體(RAM)之單晶片數位信號處理器。該DSLAM控制器預先分配一部分的其RAM作為一封包緩衝記憶體共用區，而該記憶體共用區包含一所選擇複數個最小容量記憶體區段。所接收的每一訊息資料封包都造成在該記憶體共用區內分配(54)一封包緩衝區，且該封包緩衝區之資料結構具有一有效資料指標、及一資料大小值。將足夠的記憶體空間分配給該封包緩衝區之載送資料、以及該封包之預期通訊協定起始碼及結尾碼；在必要時，可利用封包緩衝區資料項的一連

英文發明摘要 (發明之名稱：EFFICIENT PACKET BUFFER MANAGEMENT IN A DIGITAL SUBSCRIBER LINE ACCESS MULTIPLEXER)

A method for processing message data packets in a digital subscriber line access multiplexer (DSLAM), for use in a digital subscriber line communications network. The DSLAM includes a DSLAM controller, which may be realized as a single-chip digital signal processor having its own on-chip random access memory (RAM). Flow control operations are carried out by the DSLAM controller by pre-allocating a portion of its RAM as a packet buffer memory pool, comprised of a selected plurality of minimum-size memory blocks. Each received message data packet causes the allocation 54 of a packet buffer within memory pool as a data structure having a valid data pointer and a data size value. Sufficient memory space is allocated for the packet buffer payload and its expected protocol headers and tails; if necessary, the packet buffer may be allocated in the memory pool by way of a linked list of packet buffer entries. The

經濟部智慧財產局員工消費合作社印製

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

四、中文發明摘要（發明之名稱：_____）

結表，而在該記憶體共用區中分配封包緩衝區。主要係以修改該有效資料指標及資料大小值欄位之方式，且在不重新拷貝訊息資料本身的情形下，增添、刪除、及修改起始碼及結尾碼資訊，而經由一系列進出應用層的通訊協定(58)、(59)、(60)處理訊息資料封包。執行對封包緩衝區的解除分配(70)，以便自動將各連續未分配的記憶體區段相互合併，因而避免該記憶體共用區之分割。

英文發明摘要（發明之名稱：_____）

message data packet is processed through a series of protocols 58, 59, 60, both to and from the application level, by adding, deleting, and modifying header and tail information, primarily by modification of the valid data pointer and data size value fields, and without recopying the message data itself. De-allocation 70 of the packet buffer is carried out so as to automatically merge contiguous non-allocated memory blocks with one another, thus avoiding fragmentation of the memory pool.

六、申請專利範圍

1. 一種處理一耦合到複數個輸入/輸出頻道的網路多工器中的訊息資料封包之方法，包含下列步驟：

分配一記憶體中之一記憶體共用區；

回應自一第一輸入/輸出頻道接收到一訊息資料封包，而將該記憶體共用區中的一封包緩衝區分配給該訊息資料封包；

將該訊息資料封包儲存在該記憶體共用區內的該封包緩衝區之一有效資料部分；

修改該封包緩衝區的該有效資料部分之大小，而處理至少一個通訊協定之所接收訊息資料封包；

在該處理步驟之後，使該封包緩衝區的訊息資料封包在該記憶體共用區中排隊等候，以便經由一第二輸入/輸出頻道而傳輸；以及

將該記憶體共用區中所分配的該封包緩衝區解除分配。

2. 如申請專利範圍第1項之方法，其中該處理步驟包含下列步驟：

執行其中包括修改該封包緩衝區的該有效資料部分的大小之一序列之作業，而處理複數個通訊協定的所接收訊息資料封包。

3. 如申請專利範圍第2項之方法，其中該分配一封包緩衝區之步驟包含下列步驟：

將該記憶體共用區中的一所選擇連續系列的記憶體區段保留給該封包緩衝區；以及

六、申請專利範圍

指定該封包緩衝區的一偏移位址及資料大小值，而在該連續系列的記憶體區段內界定一有效資料部分，以供儲存所接收的訊息資料封包。

4. 如申請專利範圍第3項之方法，其中該等序列的處理步驟包含下列步驟：

修改該偏移位址及資料大小值，以便增加該封包緩衝區的該有效資料部分之大小，並將一通訊協定起始碼添加到該新增加的有效資料部分之開始處，而增加該通訊協定起始碼。

5. 如申請專利範圍第3項之方法，其中該等序列的處理步驟包含下列步驟：

修改該偏移位址及資料大小值，以便減少該封包緩衝區的該有效資料部分之大小，而刪除一通訊協定起始碼。

6. 如申請專利範圍第3項之方法，其中該等序列的處理步驟包含下列步驟：

修改該資料大小值，以便增加該封包緩衝區的該有效資料部分之大小，並將一通訊協定結尾碼添加到該新增加的有效資料部分之結尾處，而增加該通訊協定結尾碼。

7. 如申請專利範圍第3項之方法，其中該等序列的處理步驟包含下列步驟：

修改該資料大小值，以便減少該封包緩衝區的該有效資料部分之大小，而刪除一通訊協定結尾碼。

六、申請專利範圍

8. 如申請專利範圍第3項之方法，其中每一記憶體區段具有一對應於所選擇一數目的資料字組之大小，而每一資料字組對應於一整數數目的位元組；

其中該儲存訊息資料封包的步驟進一步包含下列步驟：

設定一對應於該封包緩衝區的一終端資料字組中的有效資料位元組數之旗標值。

9. 如申請專利範圍第3項之方法，其中該解除分配步驟包含下列步驟：

使該記憶體共用區中的該封包緩衝區可供重新分配；
以及

將先前分配給該封包緩衝區的所選擇連續系列之記憶體區段與該記憶體共用區中相鄰可用的記憶體區段合併。

10. 如申請專利範圍第2項之方法，其中該分配一封包緩衝區之步驟包含下列步驟：

為該封包緩衝區選擇該記憶體共用區中一非連續複數個區域；

以一連結表之形式使每一該等非連續區域彼此相關聯；以及

指定該封包緩衝區的一偏移位址及一資料大小值，而為每一該等非連續區域界定該記憶體共用區內的一有效資料部分，以供儲存一部分的所接收訊息資料封包。

11. 如申請專利範圍第1項之方法，其中該分配一記憶體共

(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

用區之步驟包含下列步驟：

分配一整數數目的固定大小記憶體區段，作為一連續記憶體區域。

12. 一種用來在一網路與複數個用戶端位置之間傳送訊息資料之多工器，包含：

耦合到該網路的一網路介面；

複數個輸入/輸出頻道介面，每一輸入/輸出頻道介面係耦合到該等複數個用戶端位置中之一用戶端位置；

耦合到複數個輸入/輸出頻道介面中相關聯的一些輸入/輸出頻道介面之至少一個收發機機能，用以對經由該等相關聯的輸入/輸出頻道介面而傳送之訊息資料執行數位作業；

耦合到該網路介面及該等至少一個收發機機能之一控制器，用以處理所傳送的訊息資料之訊息資料封包，該控制器包含：

一記憶體；

邏輯電路，用以執行一序列的作業，該序列的作業包含下列步驟：

分配該記憶體中之一記憶體共用區；

回應自一第一輸入/輸出頻道接收到一訊息資料封包，而

將該記憶體共用區中的一第一封包緩衝區分配給該第一訊息資料封包；

將該訊息資料封包儲存在該記憶體共用區內的該

(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

第一封包緩衝區之一有效資料部分；

修改該第一封包緩衝區的該有效資料部分之大小，而處理至少一個通訊協定之所接收第一訊息資料封包；

在該處理步驟之後，使該第一封包緩衝區的訊息資料封包在該記憶體共用區中排隊等候，以便傳送到該網路介面；以及

將該記憶體共用區中所分配的該第一封包緩衝區解除分配。

13. 如申請專利範圍第12項之多工器，其中該控制器的該邏輯電路執行的該序列之作業包含下列步驟：

回應自該網路介面接收到一訊息資料封包，而將該記憶體共用區中的一第二封包緩衝區分配給該訊息資料封包；

將該訊息資料封包儲存在該記憶體共用區內的該第二封包緩衝區之一有效資料部分；

修改該第二封包緩衝區的該有效資料部分之大小，而處理至少一個通訊協定之所接收訊息資料封包；

在該處理步驟之後，使該第二封包緩衝區的訊息資料封包在該記憶體共用區中排隊等候，以便傳送到一輸入/輸出頻道；以及

將該記憶體共用區中所分配的該第二封包緩衝區解除分配。

14. 如申請專利範圍第12項之多工器，其中係將該控制器的

(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

該記憶體及邏輯電路實施於一可程式數位信號處理器積體電路。

15. 如申請專利範圍第14項之多工器，其中係將該數位收發機實施於一可程式數位信號處理器積體電路。

16. 如申請專利範圍第12項之多工器，其中分配一封包緩衝區之該作業包含下列步驟：

將該記憶體共用區中的一所選擇連續系列的記憶體區段保留給該第一封包緩衝區；以及

指定該第一封包緩衝區的一偏移位址及資料大小值，而在該連續系列的記憶體區段內界定一有效資料部分，以供儲存所接收的訊息資料封包；

且其中該處理作業包含下列步驟：

執行其中包括修改該第一封包緩衝區的該有效資料部分的大小之一序列的作業，而處理複數個通訊協定的所接收訊息資料封包。

17. 如申請專利範圍第16項之多工器，其中該序列的處理步驟包含下列步驟：

修改該偏移位址及資料大小值，以便增加該第一封包緩衝區的該有效資料部分之大小，並將一通訊協定起始碼添加到該新增加的有效資料部分之開始處，而增加該通訊協定起始碼。

18. 如申請專利範圍第16項之多工器，其中該序列的處理步驟包含下列步驟：

修改該偏移位址及資料大小值，以便減少該第一封包

(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

緩衝區的該有效資料部分之大小，而刪除一通訊協定起始碼。

19. 如申請專利範圍第16項之多工器，其中該序列的處理步驟包含下列步驟：

修改該資料大小值，以便增加該第一封包緩衝區的該有效資料部分之大小，並將一通訊協定結尾碼添加到該新增加的有效資料部分之結尾處，而增加該通訊協定結尾碼。

20. 如申請專利範圍第16項之多工器，其中該序列的處理步驟包含下列步驟：

修改該資料大小值，以便減少該第一封包緩衝區的該有效資料部分之大小，而刪除一通訊協定結尾碼。

21. 如申請專利範圍第16項之多工器，其中每一記憶體區段具有一對應於所選擇一數目的資料字組之大小，而每一資料字組對應於作為該控制器的該邏輯電路的操作對象之一整數數目的位元組；

其中分配該第一封包緩衝區的作業進一步包含下列步驟：

設定一對應於該第一封包緩衝區的一終端資料字組中的有效資料位元組數之旗標值。

22. 如申請專利範圍第16項之多工器，其中該解除分配步驟包含下列步驟：

使該記憶體共用區中的該第一封包緩衝區可供重新分配；以及

六、申請專利範圍

將先前分配給該第一封包緩衝區的所選擇連續系列之記憶體區段與該記憶體共用區中相鄰可用的記憶體區段合併。

23. 如申請專利範圍第12項之多工器，其中該分配一封包緩衝區之步驟包含下列步驟：

為該第一封包緩衝區選擇該記憶體共用區中一非連續複數個區域；

以一連結表之形式使每一該等非連續區域彼此相關聯；以及

指定該第一封包緩衝區的一偏移位址及一資料大小值，而為每一該等非連續區域界定該記憶體共用區內的一有效資料部分，以供儲存一部分的所接收訊息資料封包。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

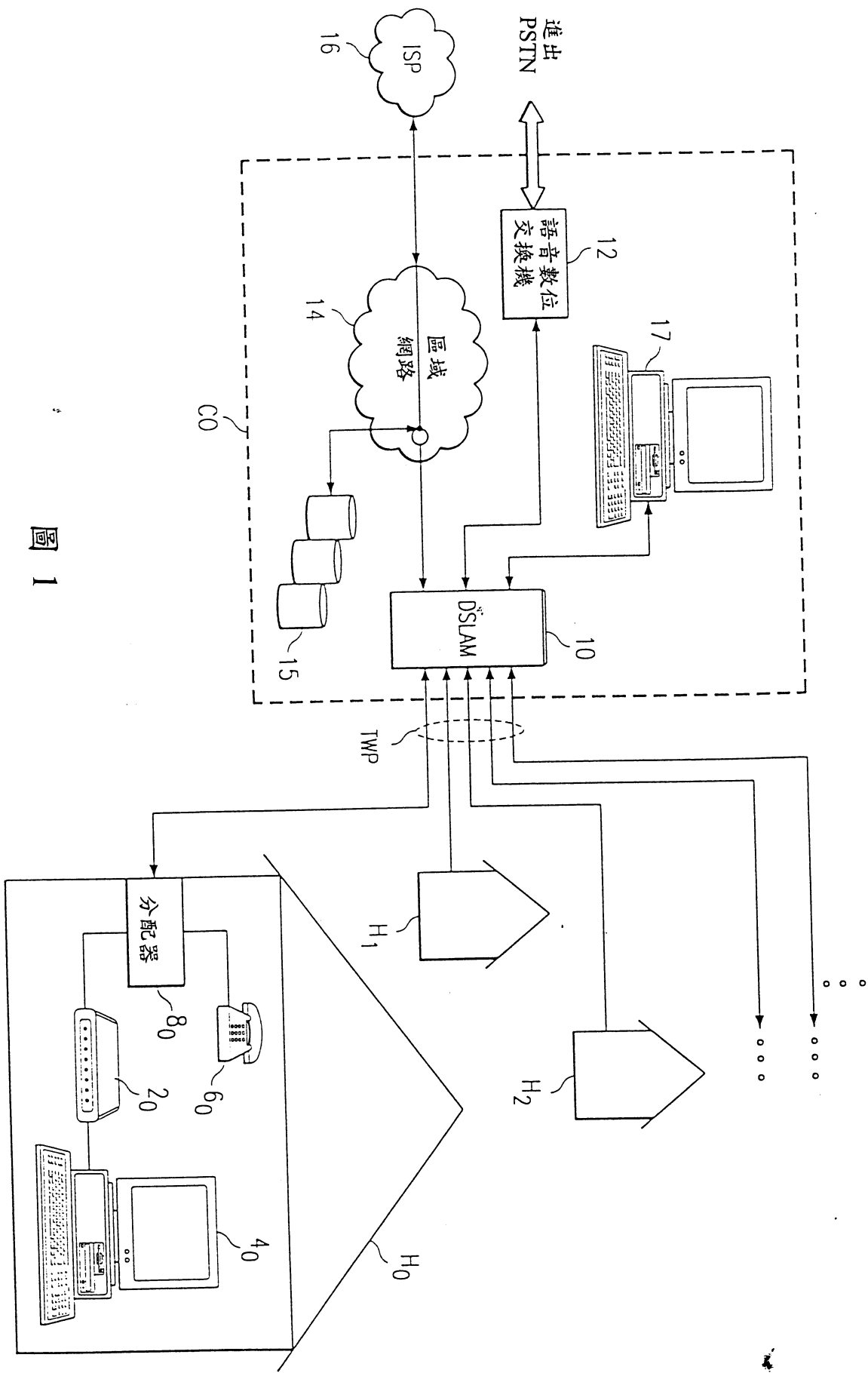


圖 1

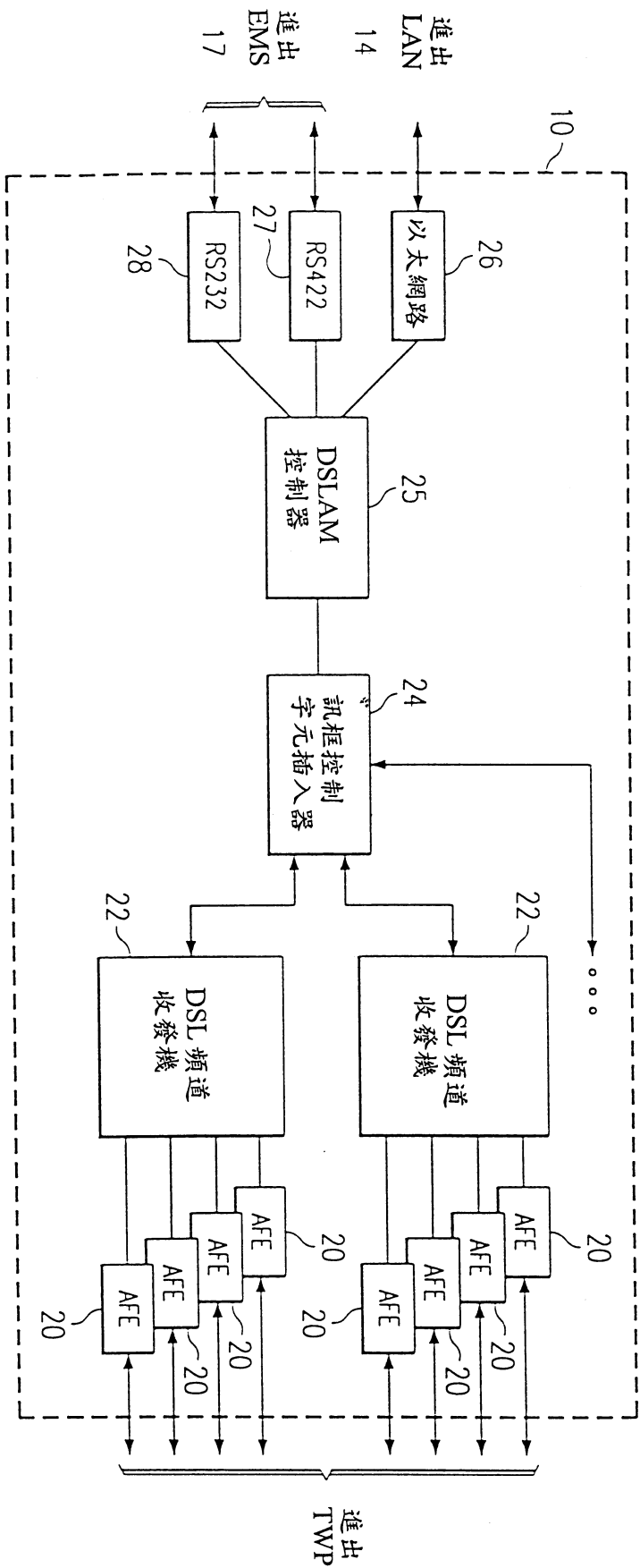


圖 2

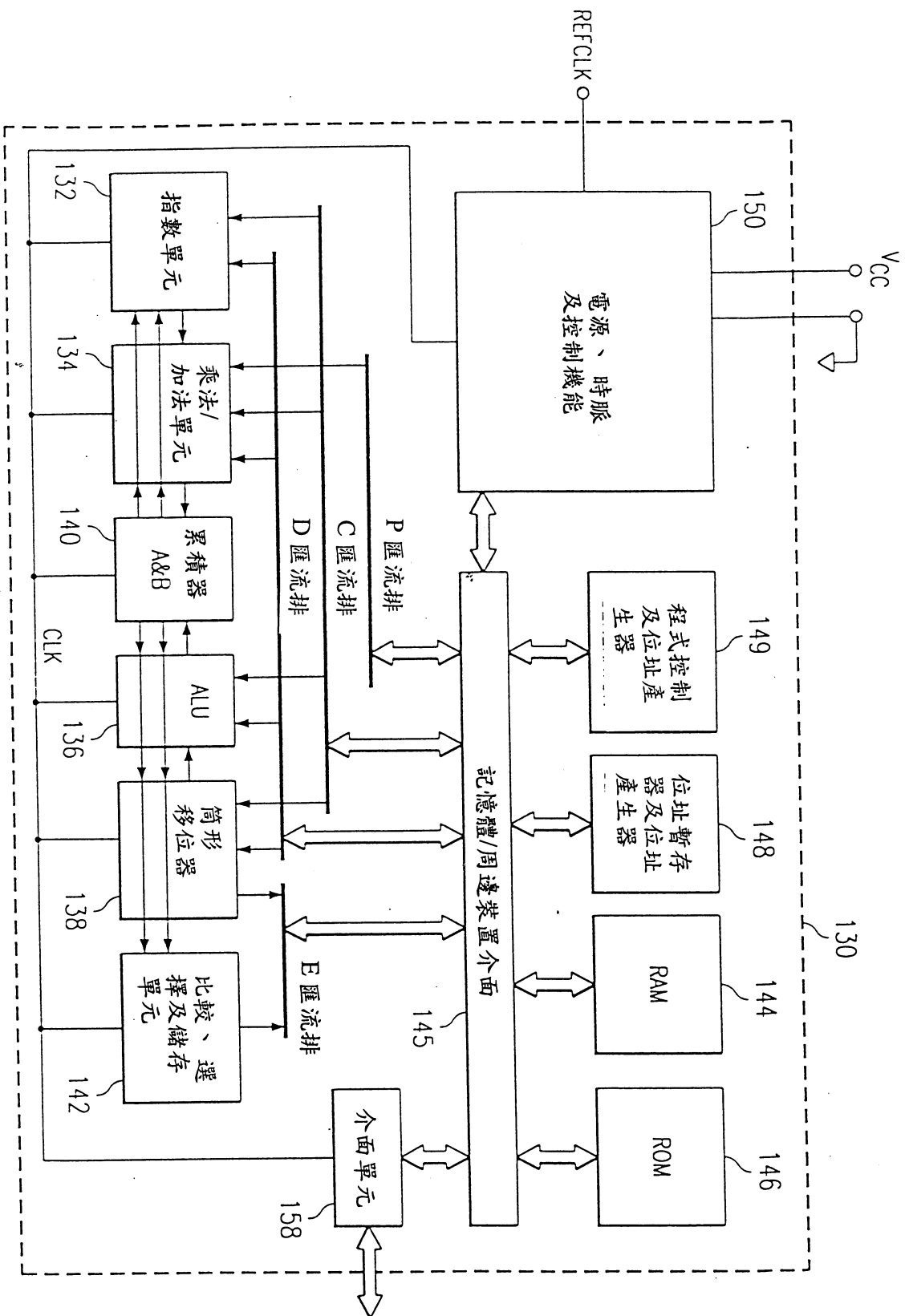


圖 3

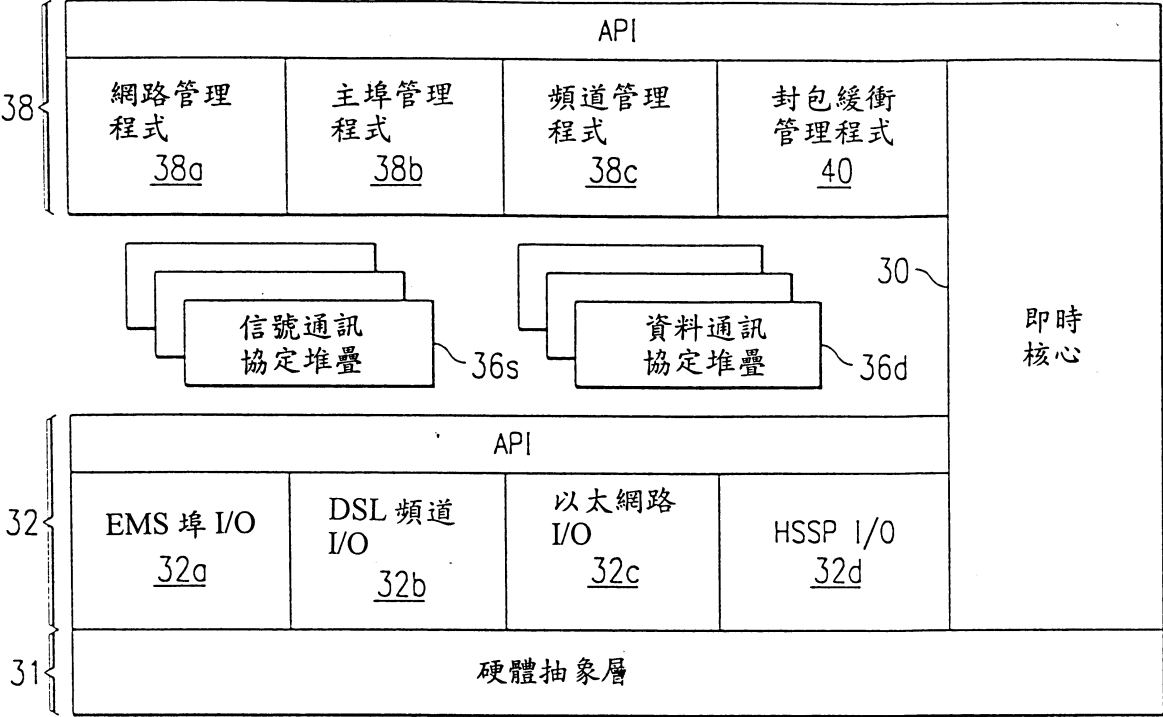


圖 4

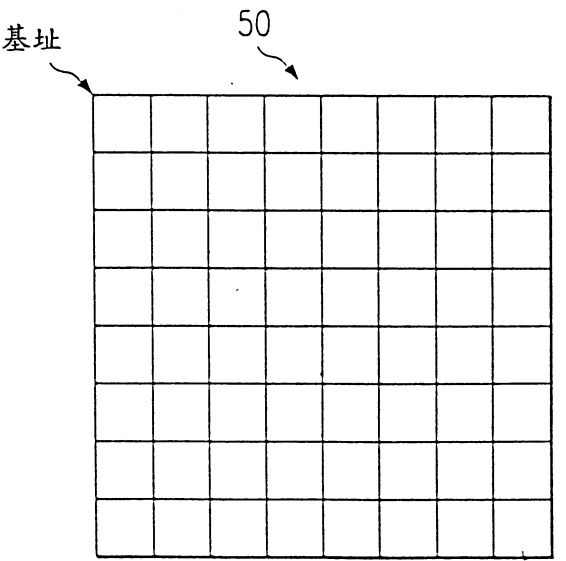


圖 5

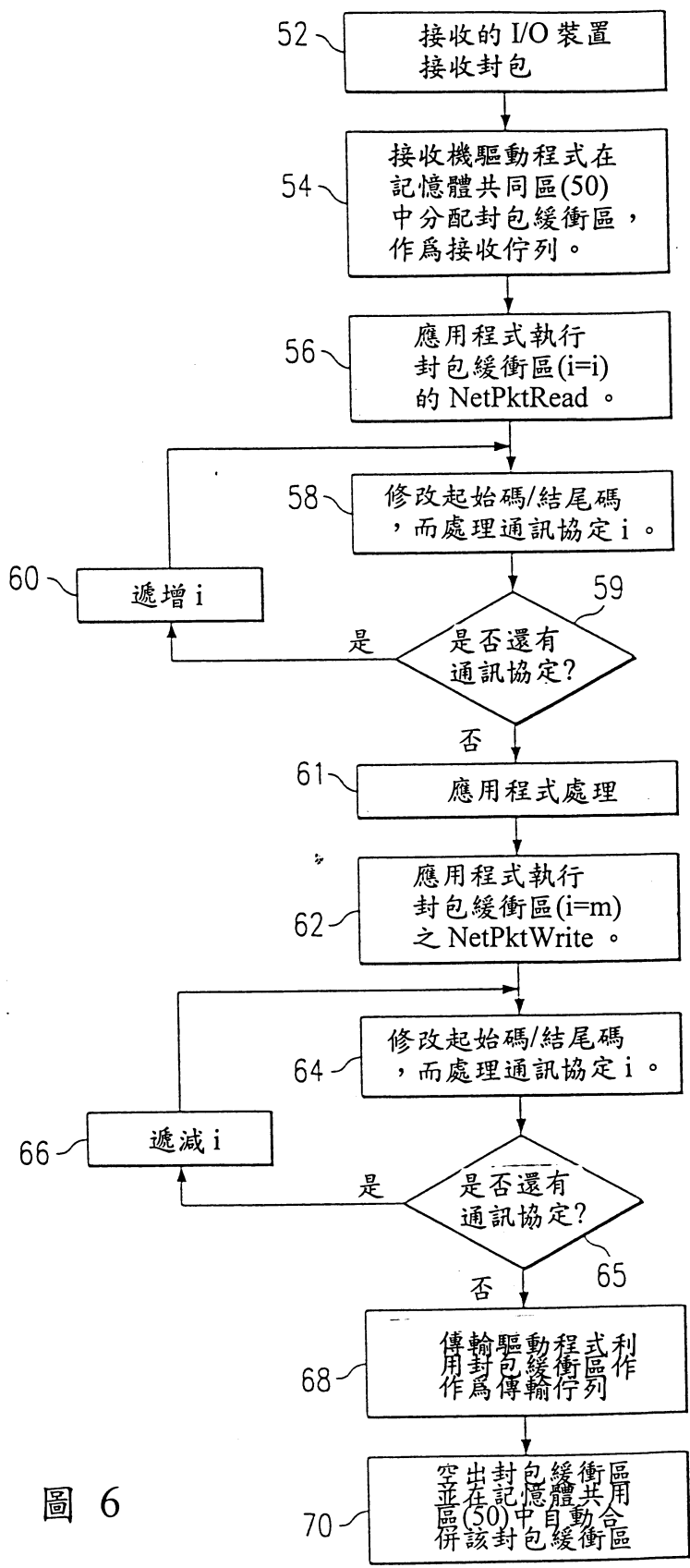


圖 6

圖 7

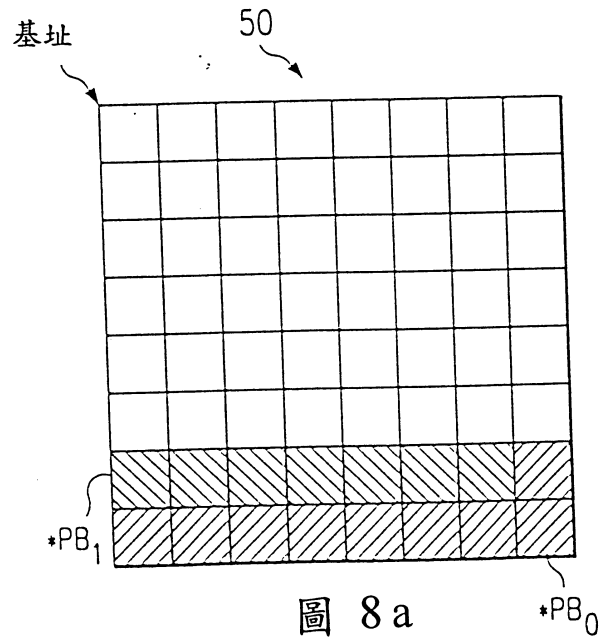
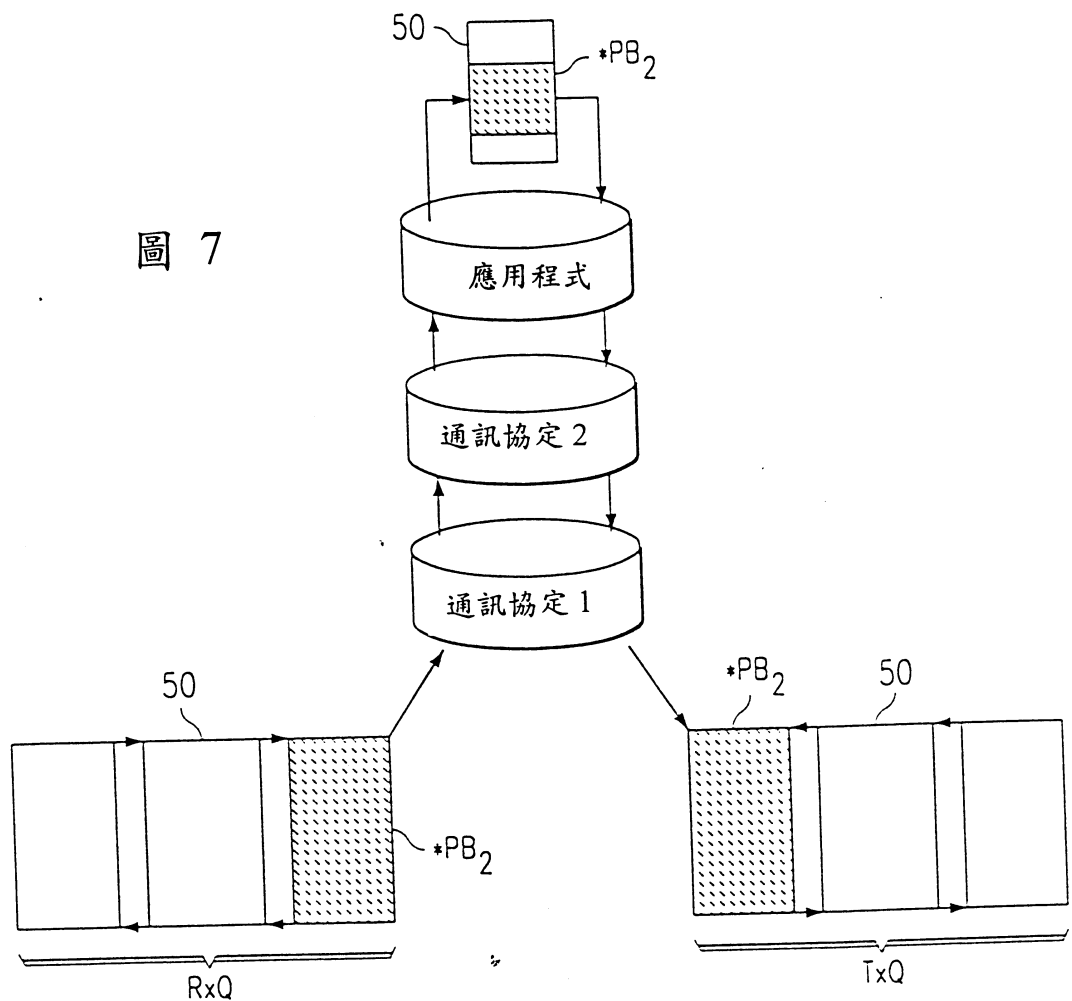


圖 8a

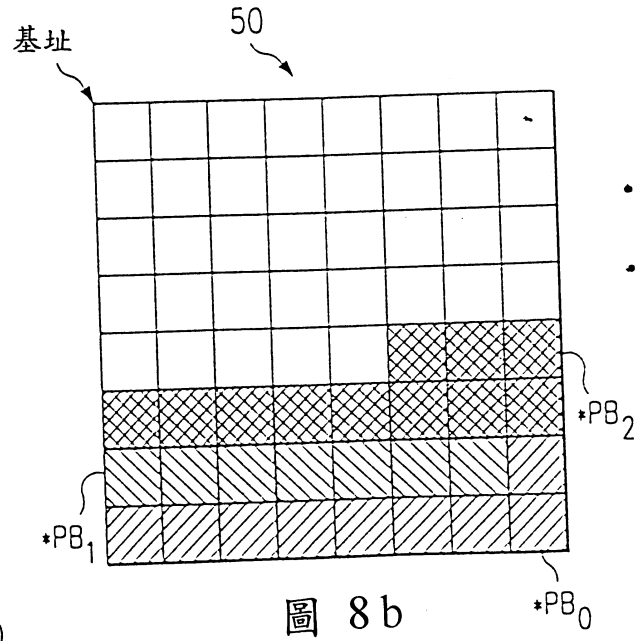


圖 8b

基址

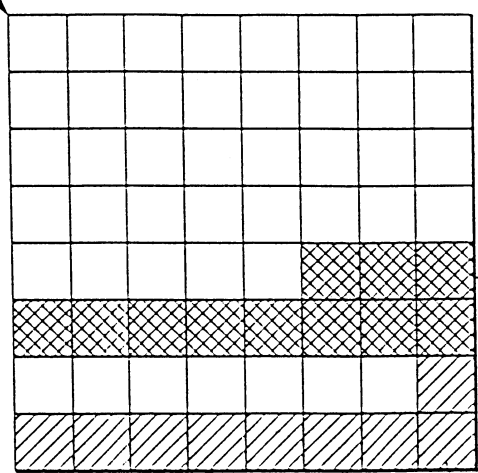


圖 8 c

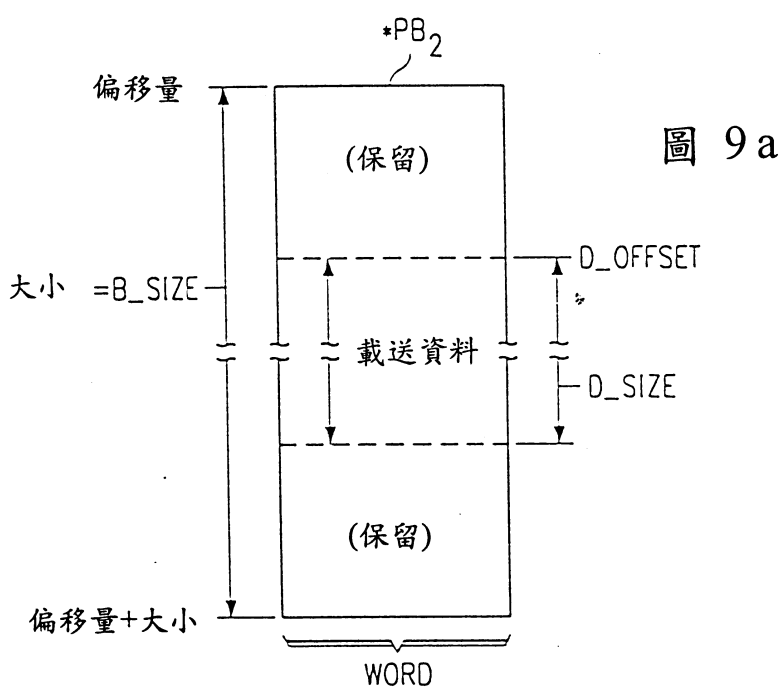
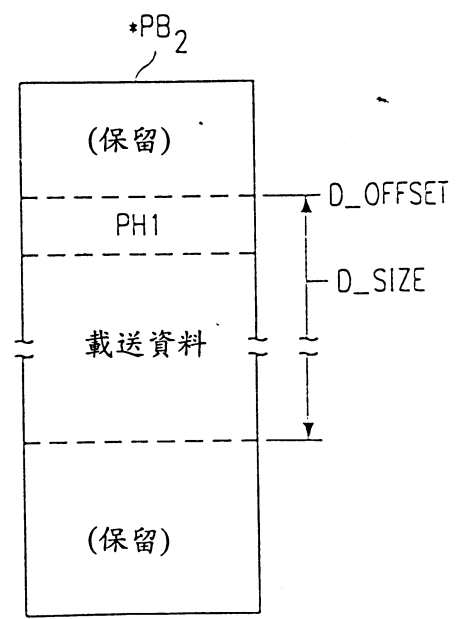


圖 9 a

圖 9 b



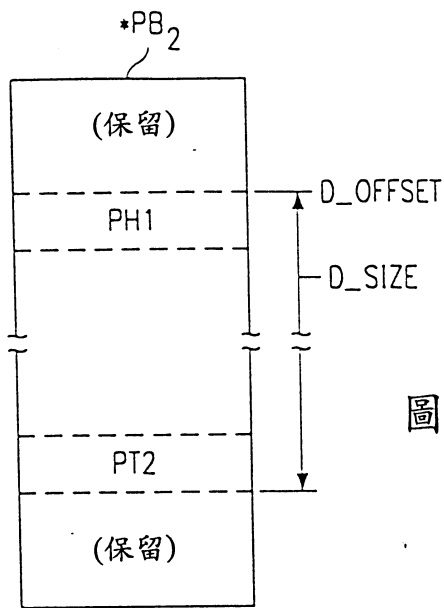


圖 9 c

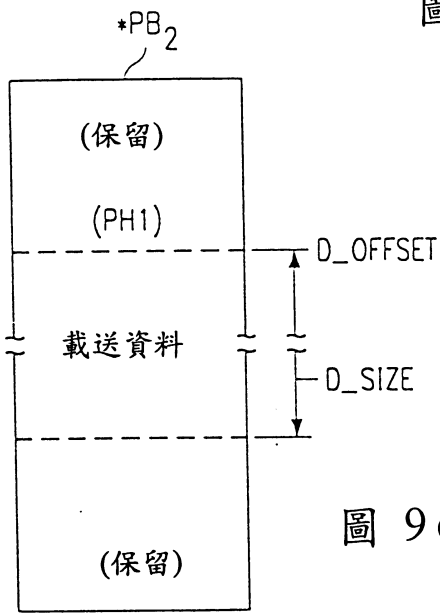


圖 9 d

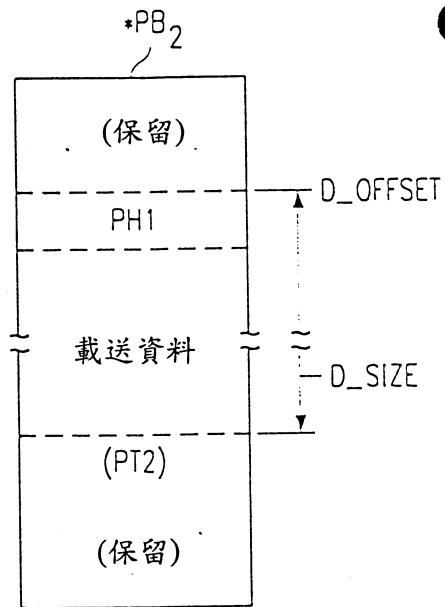


圖 9 e

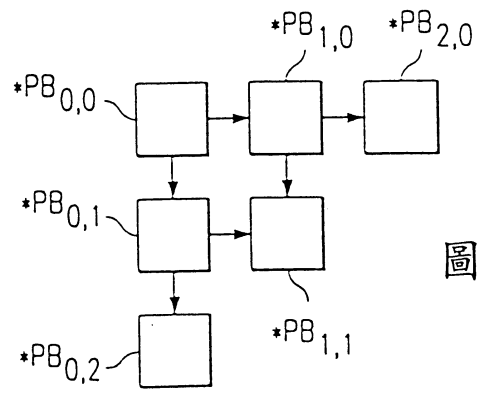


圖 10