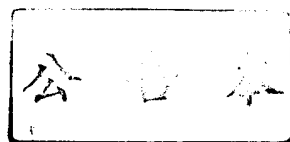
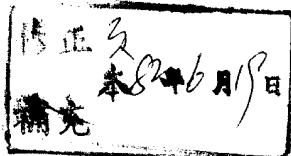


214588



申請日期	81. 6-20
案 號	81104859
類 別	G06F 15/02 G06K 9/36

A4
C4

(以上各欄由本局填註)

附件三

發明
專利說明書

一、發明 名稱	中 文	資訊處理系統之外部記憶系統
	英 文	EXTERNAL MEMORY SYSTEM FOR AN INFORMATION PROCESSING SYSTEM
二、發明 人	姓 名	(1)傑美 E·孫 (2)朋 齊士 (3)卡爾 N·庫拉母 (4)彼得 R·華尼斯
	籍 貫 (國籍)	英 國
	住、居所	(1)英國NW7 4PT倫敦米爾崗·上山路73號 (2)英國SG8 6AE哈特福舍·美爾本·樂斯頓·多爾芬巷23號 (3)英國E4 7ET倫敦·北青福·屋蘭路15號 (4)英國E6 1AR倫敦·東漢·偉爾波路12A號
三、申請人	姓 名 (名稱)	美商·愛恩股份有限公司
	籍 貫 (國籍)	美 國
	住、居所 (事務所)	美國·華盛頓州98052, 雷得蒙北東區150街4820號
	代 表 人 姓 名	荒川 實

經濟部中央標準局員工消費合作社印製

裝

訂

線

五、發明說明(3)

本發明係有關一個資訊處理裝置，其中包括一個唯一之外部記憶體單元，該單元具有一個可程式處理器。尤其是關於一個可移動之外部記憶體單元，其具有一個程式記憶體，以儲存將於一個主處理系統上執行的程式。主處理系統例如電視遊樂系統。部份設計係利用一個可程式微處理器，以增進主系統的高速繪圖處理能力。可程式處理器，包括硬體俾使用於從一個圖素基格式轉換成為字元基格式。

本申請案係與已申請案號-，-相關，其名稱“使用於一個電視遊樂器系統之具有可程式繪圖處理器的外部記憶體系統”(代理編號.1248-2)。並與申請案號-，-相關其名稱“使用於一個電視遊樂器系統之具有增強記憶體控制電路的繪圖處理器”(代理編號.1248-5)。

習知技藝之電視遊樂器，具有一個8位元微處理器，和一個相關於顯示處理的次系統，均實施於一個電視遊樂器控制板上。典型之方式係利用在卡匣上以8位元對8位元矩陣預先儲存字元，並且利用這預先儲存字元的各種不同可程式組合，以建立一個銀幕畫面顯示，因而產生繪圖。此種習知技術電視遊樂器系統，具有移動整個顯示背景，和多個受遊樂者控制之移動目標的能力。

此種習知技術系統，對於由多邊形組成的移動目標必須對每個圖框操作，例如從轉和再繪等，均無能力實際地實施這種電視遊樂功能。習知技術8位元處理器，與此系統相關顯示處理之電路，係不能夠執行例如有效地轉三度

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (4)

空間多邊形基目標，或者適當地比例於此種旋轉目標，以產生 3-D 型特殊效應。本發明則認知，精密繪圖需以一個像素-對-像素為基礎的銀幕畫面更新，並且在一個以即時為基礎而執行複雜數學。此種習知技術以字元為基礎之電視遊樂器，不能夠執行此種工作。

習知技術 8-位元電視遊樂器亦不能有效地執行其它繪圖技術，此技術需以一個像素-對-像素為基礎，能快速地更新畫面。例如，此系統不能有效地映至一個目標到一個顯示多邊形，該多邊形係為另一顯示目標（此後稱為“紋理 (texture) 映至”）於三度空間中之一部份。

改進習知 8 位元機器在繪圖能力方面的努力，為使用更有力之 16 位元處理器設計電視遊樂器系統。此 16 位元處理器所提供之電視遊樂器系統，可執行更精密圖形所需之數學的機構。此系統，例如，允許更精密彩色之產生與更佳繪圖解析度。此 16 位元電視遊樂器為字元基的系統，允許廣泛的電視遊樂功能的實施，能被預先繪製成為字元基的圖。此 16 位元電視遊樂器系統，亦可在高速多彩色背景平面之移動，及移動目標置於此平面的背面或前面。

但是此習知技術 16 位元電視遊樂器，不允許具有 3-D 型特別效應的先進電視遊樂器實際的實施。此效應顯示精密目標利用多邊形做成，其在每個圖框必需改變。例如，需許多完全旋轉目標的遊樂器，必需在一個圖框接一個圖框之基礎上放大和 / 或縮小，這在此種習知字元基 16 位元機器上實現是不實際的。發明人認知，欲有效實施此種完全旋轉和尺寸放大縮小之多邊形基目標的遊樂器，必需畫

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (5)

多邊形的邊緣，並且填入此多邊形基目標於一個像素-對-像素基上適當資料。此種工作，係在一個像素-對-像素基上執行，而消耗很多的處理時間。

於習知技術中，修改可移動遊樂器卡匣以改進遊樂器精巧性，係利用允許現存處理器定址一個較大程式記憶體位址空間，此位址空間大於主微處理器所允許的相關位址線的數目。例如，此習知技術8位元系統使用包含多記憶體控制器晶片的遊樂器卡匣，以晶片執行記憶體分組(bank)交換和其他增加功能。但此記憶體分組交換相關晶片，不能夠能電視遊樂系統執行高速圖形處理。

本發明解決習知技術之上述問題，利用提供一個唯一全部可程式之繪圖處理器，其係被設計於一個可移動外部記憶體單元內，和一個主資訊處理系統相連接。於此說明實施例中，本發明所實施的一個電視遊樂器系統，包含一個主電視遊樂器系統，與一個電視遊樂器卡匣，其內含此繪圖微處理器。

該繪圖微處理器與電視樂器系統，包含許多獨特和優點的特點，以下是某些概要。

根據本發明，一個獨特的繪圖微處理器可插入連接到一個主微電腦。欲獲得最大處理速度，繪圖處理器也可以平行於主微電腦操作。於一實施例中，繪圖處理器所在的遊樂器卡匣，亦包括一個儘讀記憶體(ROM)和一個隨機存取記憶體(RAM)。

本發明的繪圖共處理器，仲裁它自己需求和從主微電

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (6)

腦資料抓取間之記憶體變動(transaction)。該處理器能同時和主微腦執行程式，以允許高速處理，在此之前之習知遊樂器系統無法達到。

本發明的繪圖共處理器，係與遊樂器卡匣上所實施的一個三匯流排結構連接操作，其允許RAM和ROM卡匣記憶體之有效使用，利用最佳之主處理器卡匣處理器二者的能力，以有效地使用此記憶體裝置。

本發明之完全使用者可程式繪圖共處理器，包括一個唯一指令集俾允許高速處理。該指令集能有效地實施相關於3-D繪圖算術運算以及，例如，包括特殊指令利用專用硬體之執行，使用於主電視遊樂器系統的字元映至顯示上畫出個別像素。

指令集包括唯一像素基指令，從程式者觀點，係利用允許個別像素的定址創造一個“虛”位元映至---甚至若主系統為字元基者。像素資料繪圖處理器轉換字元資料，此格式典型地被主字元基16位元機器所使用。因此，例如，雖然程式者使用一個獨特“PLOT”指令以畫一個像素，當相關資料讀到RAM，資料轉換成為一個字元基格式，而使16位元主機能夠使用。

特別用途像素畫圖硬體執行此指令，以有效地允許實施高速3-D型繪圖。畫面硬體協助即時地轉換從像素座標定址成為字元映至定址，此定址主系統可自然地使用。其優點為，處理器利用指定X和Y座標寫程式，此座標定義在顯示畫面上每個像素的位置。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明(7)

因此，執行繪圖操作基於一個程式者所指定像素、及畫圖硬體快速地轉換像規格、成為適當地格式字元資料。然後，字元資料映至在主處理器的影像RAM期望的地方顯示出來。

畫圖硬體響應於各種不同畫圖相關指令，以允許可程式選擇在顯示畫面上之X和Y座標，並使用於一個特定像素之一個預先決定色彩與畫相關像素，因此，X和Y座標轉換成為相關於一個字元定義的位址，此方式使用於驅動主處理器的影像RAM。

經由下列本發明實施例之詳細說明及其圖式，本發明之優點與其它特性將更容易了解。

圖1係根據本發明實施例之一個外部記憶體系統之方塊圖。

圖2係實施例之主處理系統之方塊圖，目前之最佳實施例係使用具有一個繪圖共處理器者。

圖3係透視圖，顯示一個遊樂器卡匣之範例機械結構，其內含有一個繪圖共處理器與一個基座單元，其中包括主處理系統。

圖4A與4B係根據目前之最佳實施例之繪圖共處理器之方塊圖。

圖5係利用主處理系統執行操作起始繪圖共處理器操作之順序流程圖。

圖6係圖4所示算術與邏輯單元之更詳細方塊圖。

圖7係圖4A所示像素畫圖電路之更詳細方塊圖。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (8)

圖 8A 係一方塊圖，顯示利用畫圖控制器所接收的輸入信號，及利用畫圖控制器產生的輸出信號。

圖 8B 係像素畫圖電路於彩色矩陣內的一個彩色矩陣元素。

圖 8C 係相關於像素畫圖電路之時序，控制與資料信號。

圖 9 係圖 4A 所示 RAM 控制器之更詳細方塊圖。

圖 9A 係圖 9 所示相關於 RAM 控制器之時序，控制與資料信號。

圖 10 係圖 9 所示說明仲裁邏輯的一個電路圖。

圖 11 係本發明的繪圖共處理器實施例之再同步電路圖。

圖 12 係相關於圖 11 的再同步電路之時序信號說明。

圖 13 係本發明繪圖共處理器的 RAM 控制器之更詳細方塊圖。

圖 14 係根據本發明實施例之繪圖共處理器的快速控制器之方塊圖。

圖 15A 係一方塊圖，顯示本發明繪圖共處理器的指令解碼相關電路。

圖 15B 係時序信號，證明在圖 15A 中向前看邏輯之操作。

圖 16 與 17 係方塊圖，顯示根據本發明實施例之繪圖共處理器的暫存器控制邏輯。

圖 18 係實施例一個多邊形產生工作之繪圖共處理器的

(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)

操作範例流程圖之順序。

圖 19、20 及 21 係產生多邊形基目標的範例顯示，以說明根據本發明實施例之尺寸的比例與旋轉特色。

本發明之最佳實施例之詳細說明如下：

根據本實施例，本發明繪圖共處理器相互作用於一個 16 任元電視遊樂器系統。此系統係為美國的任天堂公司所發售的超級任天堂娛樂系統（超級 NES）。此超級任天堂娛樂系統於美國申請案號 07/651,265，名稱“影像處理裝置”中部份說明，其申請日期 1991 年 4 月 10 日。以及 1991 年 8 月 26 日申請，申請案號 07/749,530，其名稱“直接記憶體存取裝置與外部儲存裝置”。此等申請案於此當成參考。應了解，本發明未限於超級 NES 之相關應用，並且可許用於其它電視遊樂器系統或其它非電視遊樂器及資訊處理裝置。

為容易參考之目的，根據本發明實施例的繪圖處理器，以後稱為“瑪利歐晶片 (Mario chip)”。目前最佳實施例所說明的瑪利歐晶片，係被包裝在一個電視遊樂卡匣。應了解，本發明中之瑪利歐晶片不一定要和程式記憶體在相同卡匣內，只要在使用時連接到一個程式記憶體與主處理單元。

圖 1 係根據本發明最佳實施例中，一個範例電視遊樂器卡匣 / 外部記憶體系統。該遊樂器卡匣包括一個印刷電路板（未顯示），在此板上所有圖 1 之組件均安裝在上面。卡匣包括一系列連接器電極 1，係安置在印刷電路板的嵌入端，

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明 (10)

使用於傳輸信號到以及從超級NES主控制板。該列連接器電極1係利用一個配合連接器所收容，安置在超級NES主控制板。

根據本發明實施例，在遊樂器卡匣上所具有瑪利歐晶片(繪圖共處理器)2，係一個100到128支腳的積體電路晶片。瑪利歐晶片接收從主處理系統(例如超級NES)的許多控制，位址和資料信號。例如，瑪利歐晶片2，接收從主處理系統經腳P112的一個21MHZ時鐘信號輸入，和經腳P117一個21MHZ的系統時鐘信號輸入(或另一個預先決定頻率)。這系統時鐘信號輸入使於，例如，提供瑪利歐處理器，使用於主CPU記憶體存取之記憶體時序資訊，和提供時鐘信號在瑪利歐晶片內的時序操作。瑪利歐晶片2亦包括一個可選擇之外部時鐘信號輸入時鐘(腳110)，耦合瑪利歐晶片到一個外部石英4，以驅動瑪利歐CPU，例如，在一個較高於從主系統所接收21MHZ之頻率時鐘信號。

主CPU位址輸入(HA)從主處理(例如，超級NES CPU/圖處理單元PPU)位址匯流排經腳P37到腳P62連接到瑪利歐晶片2。類似地，從主系統資料輸入(HQ)從主CPU資料匯流排經腳P65-P72耦合到瑪利歐晶片2。此外，瑪利歐晶片2經P119，經腳P118的一個重置信號，和經腳P104，P105的讀和寫控制信號接收從主CPU的一個記憶體更新信號RFSH。瑪利歐晶片產生一個中斷要求信號IRQ，並且耦合這信號IRQ經腳P120到超級NES。從超級NES所接收其它控制信號，例如經腳106的一個ROMSEL信號，例如使用於起始一個主程

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 1(1)

式 ROM10 所存取。此外，卡匣亦包括一個確認處理器 3，在輸入 I，輸出 O，和重置 R 線上與一個超級 NES 確認處理器交器資料。使用於確認遊樂器卡匣之確認處理器 3 與安全系統，也可參考美國專利案號 4,799,635。

瑪利歐晶片耦合到 RAM6 和 8 係經 RAM 位址匯流排 (RAM A)，與 RAM 位址腳 P74-P91，與 RAM 資料匯流排 (RAM D) 和資料腳 P93-100。這些 RAM 可為動態記憶體裝置，其控制部份使用行位址和列位址選通脈衝信號 (RAS, CAS)，經腳 P90 和 P91 相連接。利用一或多個靜態 RAM 代替動態 RAM，並且腳 P90 和 P91 使用於連接位址信號到其個別的 RAM，而不需使用行位址和列址選通脈衝信號。一個寫致能控制信號 WE，經腳 107 適當地耦合到 RAM6 到 8。

利用主 CPU 產生讀和寫控制信號 (R, W)，並且經腳 104 和 105 連接到瑪利歐晶片。利用監看該等讀和寫信號線，瑪利歐晶片能決定此超級 NES CPU 企圖執行之記憶體存取操作的本質。類似地，從主系統幾乎所有位址和控制線，利用瑪利歐晶片監督，以追蹤該主 CPU 企圖做的軌跡。瑪利歐晶片所接收之 ROM 和 RAM 位址信號，被監督和通過到適當的記憶體裝置。在這方面，ROM 位址經 ROM 位址匯流排和腳 P2 及 P26，耦合到程式 ROM10，並且 RAM 位址經腳 P74 到腳 P91 耦合到 RAM6 和 9。從主 CPU 之 ROM 和 RAM 資料輸入，經 ROM 資料匯流排和腳 P28-P35，並經腳 P93 到 P100 適當地耦合到 ROM10，瑪利歐晶片可使用於連接到一個廣範圍的不同記憶體裝置，除了上述的 ROM 和 RAM 之外。例如，正計劃

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (12)

使瑪琍晶片使用於只有CD ROM的電視遊樂器系統。

例如，圖1中，一個CD ROM(未顯示)代替ROM10，使用於儲存字元資料，程式指令，影像，圖面，和聲音資料。一個傳統型CD讀頭(亦未顯示)適合於連接瑪琍歐晶片2，從位址匯流排P2-P26接收記憶體位址，使用於從資料匯流排P28-P35上存取資料和/或指令。CD讀取與CDROM儲存系統之細操作與特定指令，對該行業專業人士所習知。利用CD ROM儲存的優點，係大幅減低每位元組資料儲存的成本。該資料儲存成本，也許比在半導體ROM儲存成本少成100至1000百分比。然而不幸地，CD ROM記憶體存取/讀出時間，甚至於慢於半導體ROM。

瑪琍歐晶片使用三匯流排結構，允許資料至少平行地使用三個匯流排。在此方面，如圖1所示遊樂器卡匣，瑪琍歐晶片2耦合到一個ROM匯流排(包括ROM資料線、ROM位址線和控制線)，一個RAM匯流排(包括RAM位址線、資料線、和控制線)以及一個主處理器匯流排(包括主位址、資料和控制線)。

瑪琍歐晶片結構，允許平行管道式(pipelined)操作發生，以獲得最佳之輸出容量。在此方面，瑪琍歐晶片從ROM中讀出一個資料位元組，當處理其它資料時，仍進一步寫資料到RAM，以允許有效地執行3-D相關繪圖。如下進一步說明，瑪琍歐晶片2CD部使用一個16位元結構，並仍設計成界面於8位元ROM10和RAM6、8晶片。所有內部資料匯流排的內部暫存器均為16位元，從ROM10讀出，並且寫

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (13)

入 RAM6、8“緩衝”，並且典型地不使程式執行速度慢下來。

類似地，瑪利歐晶片2也可從CD ROM中，存取指令和繪圖資料。並且寫該資訊進入RAM6、8中，使用於後續DMA傳送進入主處理器的影像RAM，例如超級NES圖形處理單元(PPU)。該行業專業人士將予欣賞者，係瑪利歐晶片2可程式資料的座標轉換，從CD ROM直接到PPU的影像RAM。省略通過RAM儲存和存取操作。

瑪利歐晶片2的極端快速處理速度，使得CD ROM儲存能實際地使用於繪圖應用，雖然CD ROM需長讀存取時間。影像及聲音在儲存CD ROM之前，利用傳統資料壓縮技術壓縮。資料壓縮與復原技術對該行業人士是為習知。在從CD ROM存取壓縮資料後，瑪利歐晶片2使用傳統資料復原演算法復原資料，其所使用時間係很短於利用傳統繪圖處理器所能完成者。因為它用一個21MHZ時鐘信號操作，瑪利歐晶片2完成復原，在預定時間週期內使資料傳送到RAM6、8。

因此，典型CD ROM之存取週期，能大量存取影像和聲音資料(壓縮形式)。但是，此相對長存取時間的效應縮小因為在利用瑪利歐晶片2資料復原後，每個資料位元組的實際存取時間大幅減少。利用瑪利歐晶片2執行復原時，主繪圖處理器，例如超級NES PPU，則可以執行其它處理工作。當然，假如速度對一個特別應用不需考慮，則瑪利歐晶片2能從CD ROM中存取未壓縮形式的資料。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (4)

當使用靜態 RAM，卡匣亦可包括一個備用電池。一個備用電池 12 經一個電阻 R，耦合到一個傳統備用電池電路 14，提供靜態 RAM 和一個靜態 RAM 晶片選擇信號 RAMCS 的一個備用電壓 (RSRAM)，假如電供應喪失時可提供一個資料儲存功能。

此外，耦合到 RAM 位址匯流排，可選設定電阻 16。在正常操作時，瑪利歐晶片位址線是輸出到 RAM6 和 8。但在重置和開機操作期間，使用這位址線當成輸入線，而產生一個高或低信號係依是否其連接到一個預先決定之電壓 VCC 或接地而定。由此方式，一個 "1" 或 "0" 適當地讀進一個內部瑪利歐晶片暫存器。在重置後，依這些電阻的不同設定，瑪利歐晶片能決定 (在程式執行期間)，例如，多工器時鐘信號速率，到瑪利歐晶片耦合的 RAM 存取時間，使用於瑪利歐其它操作之時鐘信號速率等。經由這些選擇設定暫存器的設定，瑪利歐晶片，例如，可使用於多種不同型式記憶體裝置，而不需修改任何瑪利歐晶片之設計。例如，假如檢視測到一個動態 RAM 設定，則再更新信號於適當次數。此外，選擇設定使用於控制速度，例如，處理器多工器電路操作，並允許利用繪圖處理器執行其它指令，其速率較快而可執行某些乘法指令。因此，利用起始一個延遲乘法執行，其餘指令能在較快時鐘信號速率執行，(例如，處理器也許設定在時鐘信號 30MHz，因此選擇設定將有效地使乘法指令於 15MHz 執行)。

圖 2 係一個實施例之主電視遊樂器系統的方塊圖，其

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (15)

中範例遊樂器卡匣如圖1所設計者。圖2，例如，代表美國的任天堂所售的超級NES。但本發明未限制在如圖2所示之超級NES相關之應用或系統。

超級NES包括它的控制板20內，一個16位元主CPU，例如，其也許是一個65816相容微電腦。CPU22耦合到一個工作RAM32，例如，其或許包括128K位元組的儲存。CPU22耦合到一個圖形處理單元(PPU)24，依次耦合到一個影像RAM30，其也許包括例如儲存的字組。在垂直或水平熄滅脈衝間隔間內，CPU22經PPU24存取影像RAM30，除了在有效線掃描期間內，當PPU24存取影像RAM時。PPU24從影像RAM30，產生一個影像顯示在使用者的電視36上。CPU亦耦合到一個聲音處理單元APU26，該單元耦合到一個工作RAM28。APU也許包含商業用的聲音晶片，產生相關於電視遊樂器程式之聲音，此程式儲存在遊樂器卡匣之ROM10內。CPU經APU26只能存取工作RAM28。PPU24和APU26經RF調變器單元34，耦合到使用者家用電視36。

在超級NES之影像RAM30，必需載入儲存在卡匣程式ROM11中適當字元資料。(卡匣不只儲存遊樂器程式，並且在玩遊樂器期間使用字元資料)。任何移動目標，例如幽靈資訊，或背景資訊的顯示，在使用前需放在影像RAM30內。程式ROM10存取利用CPU22主位址與資料匯流排，經一個配合連接器18耦合到如圖1所示印刷電路板邊緣連接器1。PPU24連接到遊樂器卡匣，經共用主CPU資料與位址匯流排與連接器23，因此提供一個路徑，使用於耦合PPU

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (16)

資料與控制信號到卡匣。APU26共用主CPU匯流排及聲音匯流排27，連接到遊樂器卡匣。

CPU22位址空間係被映至，因此程式ROM10位置在開始位置0，並典型地分割成為32位元組段。程式ROM大約使用一半的CPU位址空間。在每個CPU位址空間32K位元段之最上面位置，典型地使用位址工作RAM32和各種不同暫存器。程式ROM10典型地為4個百萬位元組。使用在超級NES之CPU22能夠定址整個程式ROM10。在另一方面，瑪利歐晶片2只包括一個16位元程式計數器，並且因此包括組暫存器，使用於選擇在程式ROM10中之32K位元組。

於本實施例中，瑪利歐晶片只有一個全24位元空間，相對於超級NES記憶體映至。其包含ROM10起始位址在\$00:800與卡匣上RAM晶片6,8起始位址在\$70:000。

由於ROM10和與卡匣上RAM6,8是分開匯流排，因此瑪利歐晶片能平行地存取它們。並且RAM6,8存取速度快於ROM，並且瑪利歐晶片設計亦使用此執行優點。瑪利歐晶片不存取在超級NES內任何記憶體，亦即，不存取工作RAM32或PPU影像RAM30。

欲使瑪利歐晶片處理資料或畫成一個位元映至，則資料需包含在瑪利歐卡匣RAM晶片6,8內。因此，任何NES CPU程式和瑪利歐晶片程式所共同之變數，必需放在瑪利歐卡匣RAM晶片6,8內。任何瑪利歐晶片程式所需使用的預先儲存資料可以在ROM10，並且任何變數將在RAM6,8內。

任何超級NES程式需的專用變數，不需在卡匣RAM6,8

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (17)

中。事實上，因此 RAM6,8 是位於記憶體空間的較高位址。最好放置卡匣 RAM6,8 在高優先需求的基礎上。任何非必要變數應儲存在超級 NES 內部 RAM32。

瑪利歐晶片所寫入的位元映至，係在瑪利歐卡匣 RAM 6,8，並且將在超級 NES 的控制下執行 DMA，傳送進入 PPU 的影像 RAM30，當每個位元映至圖框完全繪出時。

假如當瑪利歐晶片不出現時超級 NES 的 CPU22 存取在超級 NES 控制板內所有內部 RAM。瑪利歐晶片未存取這 RAM，因此瑪利歐 ROM/RAM 晶片與內部超級 NES 之間所傳送所有資料，必需利用 CPU22 其本身起始。資料傳送可利用 CPU22 的程式，或利用 DMA 傳送移動整塊資料。瑪利歐卡匣 ROM10 和 RAM6,8，如一般使用於所有遊樂器程式者。

CPU22 只有控制器，使得 CPU 可暫時存取到卡匣 ROM 或 RAM 晶片。在電源開啓或重置之條件下，瑪利歐晶片關閉，並且 CPU22 全部存取於卡匣 ROM 和 RAM 晶片。欲使瑪利歐晶片執行一個程式，必需使 CPU22 程式放棄其到 ROM 或 RAM 晶片之存取，最好是二者。並且等待瑪利歐晶片完成它已指派之工作，或另一方式，CPU22 能拷貝某些程式碼到內部工作 RAM32，並且在那裡執行之。

瑪利歐晶片只有多個暫存器，並且可超級 NES CPU 之側邊程式或可讀出。映至到 CPU22 記憶體映至，其起始位址在 \$ 00:3000。

如圖 2 所示，超級 NES 產生和接收不同的控制信號。當超級 NES CPU22 需存取程式 ROM10 時，它產生一個控制信

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (18)

號 ROMSEC。欲起始一個記憶體更新，超級 NES 則產生一個更新信號 RFSH。當瑪利歐晶片完成一個操作，它傳輸一個中斷信號 IRQ 於中斷要求線上，係相關於超級 NES CPU。此外，CPU22 產生讀和寫信號。

從在控制板 20 內的時序波道電路 21，產生系統時序信號。在主控制板 20 內亦產生一個開機/重置信號，並且耦合到遊樂器卡匣。

超級 NES 亦包括一個確認處理裝置 25，係交換在輸入 I 和輸出 O 的資料。並且根據上述美國專利 4,799,635，在電視遊樂器上重置 R 導線與一個確認處理裝置 3。在美國專利 4,799,635 所示處理裝置 25 中，保持 CPU22 在重置狀態，直到建立確認。

如圖 2 所示方塊圖之超級 NES 電視遊樂器，在此已一般性地說明。對於超級 NES 進一步詳情包括 PPU24，可在美國申請案號 07/651,265 發現到，其名稱“影像處理裝置”，其申請日 1991 年 4 月 10 日，此申請案在此當成參考。超級 NES 和遊樂器卡匣之間如何傳送資訊，可在美國申請案號 07/749,530，申請日 1991 年 8 月 26 日，名稱“在影像處理系統與外部儲存裝置，所使用之直接記憶體存取裝置”，以及美國申請案號 07/793,735，申請日 1991 年 11 月 19 日，其名稱“嵌鑲圖形顯示裝置與在此使用外部儲存單元”。這些申請於此當成參考。

某些應用方面，發明人已認知，在垂直熄滅期間需傳送多資訊，多於主處理 DMA 電路實際所能傳送者。因此，

(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 }

也許期望的延伸垂直熄滅時間。甚至可引起稍微縮小圖形的大小。利用此方法，最大的優點在於處理速度和圖形的更新速率。

圖 3 係範例機械設計的透視圖，一個遊樂器卡匣盒 19，使用於安置瑪利歐晶片和如圖 1 所示其它卡匣結構。類似地，圖 3 係一個範例外殼的透視圖，使用於一個電視遊樂器控制板 20，與安置如圖 2 所示超級 NES 電視遊樂器硬體。使用於此種電視遊樂器控制板 20，與相關可移動遊樂器卡匣 19 之機械設計，可參考美國申請案號 07/748,938，申請日 1991 年 8 月 2 日，名稱“TV 遊樂器”，此申請案在此當成參考。

圖 4A 和 4B 係如圖 1 所示瑪利歐晶片 2 的方塊圖。首先注意集中在圖 4A 和 4B 各種不同之匯流排。指令匯流排 INSTR 係一個 8 位元匯流排，其耦合指令碼到各種不同瑪利歐晶片組件。X，Y 和 Z 匯流排是 16 位元資料匯流排。HA 匯流排是一個 24 位元主系統位址匯流排。在目前最佳實施例中，使用於耦合到超級 NES 位址匯流排。HD 匯流排是一個 8 位元主資料匯流排，使用於耦合到超級 NES 資料匯流排。PC 匯流排是一個 16 位元匯流排，其耦合瑪利歐晶片程式計數器的輸出（例如，在一般暫存器方塊 76 暫存器 R15），到各種不同系統的組件。ROMA 匯流排是一個 20 位元 ROM 位址匯流排。ROM D 匯流排是一個 8 位元 ROM 資料匯流排。RAM A 匯流排是一個位元 RAM 位址匯流排。RAMD-IN 匯流排是一個 8 位元 RAM 讀資料匯流排。以及 RAMD-OUT 是一

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明 (20)

個 8 位元 RAM 寫資料匯流排。

瑪利歐晶片與超級 NES 共用卡匣 RAM 6, 8, 此 RAM 當成瑪利歐晶片與超級 NES 間傳送資料的主要機構。超級 NES 經位址和資料匯流排 HA 和 HD, 存取瑪利歐晶片。瑪利歐晶片暫存器 76, 經超級 NES 位址匯流排 HA, 利用超級 NES 存取之。

超級 NES 經瑪利歐晶片 2, 存取卡匣程式 ROM 10 和 RAM 6, 8。ROM 控制器 104 和 RAM 控制器 88, 接收利用超級 NES 所產生之記憶體存取相關信號, 以個別起始 ROM 和 RAM 記憶體存取。以實施例說明之, 瑪利歐晶片 2 使用一個 RAM 選擇信號 RAMCS, 以認真超級 NES 正企圖定址此 RAM。

在圖 4A 和 4B 所示 X, Y 和 Z 匯流排, 係內部瑪利歐晶片資料匯流排。X 和 Y 匯流排是來源資料匯流排, 並且 Z 資料匯流排為一個目的地匯流排。這些匯流排攜有 16 位元平行資料。

當執行指令時, 瑪利歐晶片 2 把一個指令的資料的來源放在 X 和 / 或 Y 匯流排上, 並在 Z 匯流排上放目的資料。例如, 在執行一個指令時, 把二個暫存器的內容相加, 並且把結果放在第三個暫存器。算術和邏輯單元 (ALU) 50, 經 X 和 Y 匯流排接收二個來源暫存器的內容。並且耦合其結果到 Z 匯流排 (其次耦合到在方塊 76 內的一個特定暫存器)。在瑪利歐晶片 2 上利用指令解碼電路 60, 從一個指令操作碼的解碼所得到控制信號, 耦合至 ALU 50 以起始一個 ADD 操作。

從對圖 1 的說明註解, 瑪利歐晶片耦合到一個 ROM 匯

(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)

流排，一個 RAM 匯流排，和一個超級 NES 主匯流排，均能平行地通信。瑪利歐晶片 2 監督，經主超級 NES 匯流排傳輸之控制，位址與資料信號，以決定主系統正執行那一個操作。卡匣 ROM 匯流排和卡匣 RAM 匯流排平行地存取，依任何已知時間執行中的超級 NES 操作而定。在傳統超級 NES 電視遊樂器卡匣中，主 CPU 位址與資料線直線耦合到 RAM 和 ROM，因此，RAM 和 ROM 不能平行地存取。

根據本發明的一個觀點，瑪利歐晶片 2 實如圖 1 所示，從超級 NES 匯流排分離 ROM 匯流排和 RAM 匯流排。瑪利歐晶片 2 監督在超級 NES 匯流排上傳輸的信號。並且決定那個信號需耦合到 ROM 晶片和 RAM 晶片，經過二個分離 ROM 和 RAM 匯流排，由於此匯流排不是時間分割的。利用分離 ROM 和 RAM 匯流排，瑪利歐晶片 2 能同時地從 ROM 讀和寫到 RAM 中。由此方式，瑪利歐晶片能有效地使用於較便宜 ROM 晶片，因此晶片的存取時間大幅地慢於 RAM 存取時間，且在存 RAM 之前，不用等待 RAM 存取的完成。

圖 4A 中，如上所述，瑪利歐晶片 2 係完全可程式處理器，並且包括一個 ALU50。ALU50 執行所有算術功能均在瑪利歐晶片中，除了乘法是操作乘法器 64 執行，並且利用畫圖硬體處理某些像素畫圖操作。從指令解碼器 60 接收一個適當控制信號，此 ALU50 執行加法、減法、互斥或、移位與其它操作。如圖 4A 所示，ALU50 從 X、Y 匯流排接收將操作之資訊，執行此操作的起始，利用從指令解碼器 60 所接收的一個控制信號。並且耦合操作的結果到 Z 匯流排。

五、發明說明 (2)

圖 6 將進一步詳細說明 ALU。

此外，瑪琍歐晶片 2 包括特殊目的硬體，以致能 3-D 型特殊效應和其它繪圖操作有效執行，因此使用這些功能的電視遊樂器能實際達成。瑪琍歐晶片 2 包括畫圖硬體 52，協助在即時從像素座標位址，轉換成為超級 NES 自然使用的字元映至位址。其優處為，在瑪琍歐晶片上寫程式，可利用指令 X, Y 座標，以定義在顯示銀幕畫面上每個像素的位置。

因此，執行圖形操作基於一個程式者所指定像素，並且畫圖硬體電路 52 可立即轉換所指定像素成為適當格式的字元資料。然後，字元資料映至到所期望位置，使用於顯示在超級 NES 影像 30，如圖 2 所示。於此方式中瑪琍歐晶片程式者只需把超級 NES 影像 RAM 30 當成一個位元映至，而實際上它是一個字元映至。

畫圖硬體 52 響應各種不同畫圖相關指令，以允許可程式的選擇在顯示銀幕畫面上之 X 和 Y 座標，以及對一個特定像素的一個預先決定色彩，與畫到相關像素，因此，X 和 Y 座標轉換相關於一個字元定義之一個位址，此字元定義的形式使用於驅動超級 NES 影像 RAM 30。

畫圖硬體 52 具有相關資料開門，可允許在寫到卡匣 RAM 之前，儘可能緩衝像素資料，以使 RAM 資料變動最小。在 X 和 Y 座標資料轉換與緩衝在畫圖硬體 52 後，字元界定資料傳送到卡匣 RAM。

畫圖硬體經一個 PLOTX 暫存器 56 和 PLOTY 暫存器 58，個

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (3)

別接收 X、Y 座標資料。在目前最佳實施例中，PLOTX 和 PLOTY 暫存器不是分開暫存器（如圖 4A 所示），但是瑪琍歐晶片的一般暫存器（亦即，圖 4B 所示，在暫存器方塊 76 中之暫存器 R1 與 R2）。

畫圖硬體 52 亦經一個色彩暫存器 54，接收像素色彩資訊。如下所說明，每個像素的色彩是儲存在一個 8×8 暫存器矩陣，其中每個像素色彩規格佔一列的矩陣。

畫圖硬體 52 處理與耦合相關於 X、Y 之字元位址和資料，及色彩輸入到字元 RAM 6, 8。字元位址經輸出線 53 向前到 RAM 控制器 88，及到一個 RAM 位址匯流排 RAMA。字元資料經輸出線 55，多工器 93 和 RAM 資料匯流排 RAMD-out，到字元 RAM。畫圖硬體 52 允許在字元內能個別定址到像素，因此提供程式者一個“虛”位元映至顯示系統。當保持和超級 NES 字元格式相容時。“虛”位元映至保持在卡匣 RAM 中，在每個圖框的顯示完成傳送到超級 NES 影像 RAM 30，係使用，例如，一個 DMA 電路於上述申請案號 07/749,530 中。畫圖硬體 52 允許高速個別像素控制，因此某些 3-D 繪圖效應包含旋轉和尺寸大小變化，而成為實際可行。

因為由像素轉換成為字元格式，畫圖硬體 52 亦從一個卡匣 RAM 6, 8 經 RAMD-in 資料門 82 和輸入線 83，接收相關於目前像素 X、Y 的週圍其它像素之資訊。利用從 RAM 6, 8 所取出先前像素資料，並且暫時儲存在 RAM 資料門 80 中，則寫到 RAM 的次數是最小的。如圖 4A 所示 RAM 資料門 80, 84 和 86，亦當成緩衝所接收相對，個像素之色彩資料；此像

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明 (24)

素已儲存在卡匣 RAM 中之多位元平面，以提供畫圖硬體 52 之資料。

RAM 資料門 80 耦合到超級 NES 匯流排，因此超級 NES 能讀出資料門的內容。利用 RAM 控制器 88 控制 RAM 資料門 80，82，84 及 86。RAM 資料門 84 和 86 操作，從 RAM 6.8 接收資料，及從 RAM 6.8 耦合資料到目的地 Z 匯流排，使用載入暫存器方塊 76 中一個預先決定暫存器。此外，耦合到 RAM 控制器 88 是一個門 90，係緩衝 RAM 位址。在門 90 所儲存位址，係 RAM 控制器 88 經 RAM A 匯流排，使用於定址 RAM 6.8。RAM 控制器亦可利用超級 NES 經位址匯流排 HA 存取。

畫圖硬體 52 亦響應一個 READ PIXEL 指令，係讀此像素色彩資訊，利用暫存器 R1 的內容定義一個水平位置，與暫存器 R2 的內容定義一個垂直位置，並且所得結果經由目的地 Z 匯流排和輸出線 87，放在暫存器方塊 76 的一個預先決定暫存器。PLOT 硬體 52 將以圖 7、8A 和 8B 進一步詳細說明。

平行管道緩衝暫存器 62 和一個 ALU 控制器指令解碼器 62，耦合到指令匯流排 INSTR，並操作以產生控制信號 CTL (使用輸出此瑪利歐晶片)，響應於放置在指令匯流排上的命令以起始操作。瑪利歐晶片 2 係一個平行管道微處理器，當它在執行目前指令時，它抓取下一個將要執行的指令。平行管道暫存器 2 儲存將要執行的下一個指令，因此假如可能的話，允許一個週期執行指令。放在指令匯流排的指令，係利用儲存在一個暫存器內的程式計數器的內

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (5)

容定址，此暫存器例如圖4B所示在暫存器方塊76中之暫存器R15。

利用瑪利歐晶片2執行指令，也許從如圖1所示程式ROM10所獲得，或瑪利歐內部快速RAM94，或者從卡匣RAM6.8。假如將執行程式從ROM10取出，ROM控制器104(如圖4B所示)將抓取這指令，並且將它放在瑪利歐晶片指令匯流排INSTR。假如程式指令儲存在快速RAM94中，則此指令直接從快速RAM94經快速RAM輸出匯流排95，放在指令匯流排上。

主CPU，亦即，超級NES，被規劃配置部份的程式ROM10，使用於瑪利歐晶片程式指令。超級NES程式命令瑪利歐晶片，執行一個預先決定之功能，並且提供瑪利歐晶片在ROM10的位址，使用存取瑪利歐晶片的程式碼。平行管道暫存器62，在指令被執行的一個位元之前抓取指令，提供指令解碼器60之指令相關資訊，使得解碼器能夠預測在程式執行期間將發生什麼，因此允許向前看到相關處理。方塊60解碼和控制電路產生控制信號，用於命令ALU50，畫面硬體52，快速控制68等，以執行利用將被執行指令所指示之操作。

瑪利歐晶片亦包括一個高速平行乘法器64，係從ALU50分離出。乘法器64響應預先決定指令，操作從X和Y來源之匯流排所接收2個8位元數目相乘，並且載入此16位元結果到目的地Z匯流排。假如可能在一個週期執行此乘運算。輸入到乘法器64之數目可以有正負號或無正負號

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (26)

。乘法器亦能執行長乘法運算，其中二個16位元數目相乘，以產生一個32位元的結果。乘法器64亦包括相關部份乘積暫存器66，在乘運算期間俾儲存部份的乘積。從指令解碼器60利用一個控制信號致能乘法器64，當解碼到一個乘法運算時。乘法器64將執行16位元的乘法之長乘法指令，在一個最小的4個時間信號週期內執法。

長乘法指令的格式：

$R4$ (較低字組)， $DREG$ (較高字組) = $Sreg * R6$ 。執行這指令，係利用暫存器 $R6$ 的內容乘法以來源暫存器，並且儲存一個32位元結果在暫存器 $R4/DREG$ (較低/較高) 中。乘法有正負號，則在32位元結果上設定零和信號旗標。

根據下列6個步驟執行運算：

步驟1：無正負號乘 $R4[0 \cdots 15] = SREG[0 \cdots 7] * R6[0 \cdots 7]$ 。

步驟2：X 有無正負號。 $R4[0 \cdots 15] = R4[0 \cdots 15] + 256 * SREG[8 \cdots 15] * R6[0 \cdots 7]$ 。忽略乘積的較高8位元，但從加法的進位保留。

步驟3：X 有正負號。 $R5[0 \cdots 15] = CY + (R6[8 \cdots 15] * SREG[0 \cdots 7]) \div 256$ ；正負號延伸。

步驟4：X 無正負號。Y 有正負號。 $R4[0 \cdots 15] = R4[0 \cdots 15] + 256 * SREG[0 \cdots 7] * R6[8 \cdots 15]$ 。忽略乘積的較高8位元，但保留從加法的進位。

步驟5：Y 有正負號。 $R5[0 \cdots 15] = R5[0 \cdots 15] + CY + (SREG[0 \cdots 7] * R6[8 \cdots 15]) \div 256$ ；正負號延伸。

步驟6：X, Y 有正負號。 $R5[0 \cdots 15] = R5[0 \cdots 15] + RY[8 \cdots 15] * R6[8 \cdots 15]$ 。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (27)

本實施例所使用之乘法器 64，在“數位計算機算術”書中有說明。

參照圖 4B，快速控制器 48 (於圖 14 有進一步詳細顯示)，允許一個程式者，對於部份程式期望於高速執行，則有效起始載入快速 RAM94 中。此種“快速”典型地使用於執行小的程式循環，係經常在繪圖處理時發生。瑪琍歐晶片指令集包括一個“~~CACHE~~^{CH}”指令。任何在 CACHE 指令後指令，均載入到快速 RAM 中直到快速 RAM 滿為止。當執行 CACHE 指令時，目前程式計數器狀態載入到快速基底暫存器 70。因此，快速基底暫存器 70 的內容，定義此快速將被起始的開始位置。

大部份指令在一個週期內執行。從相對之低外部記憶體，例如 ROM10，或 RAM6，8 的指令，在它們被執行之前就先抓取者。此將花費一個額外 6 的週期。欲增進程式執行速度，應使用在瑪琍歐晶本身內的快速 RAM94。

快速 RAM94 可為一個 512 位元組指令快速記憶體。比較於平均程式的大小，前者係相對地小。因此，程式者必需決定如何適當使用快速記憶體 94。任何程式循環能在 512 位元組快速記憶體之大小，能以全速度執行，亦即抓取和執行均在一個週期。因為分開的匯流排，當執行從內部快速記憶體 94 之碼時，能同時存取 ROM 和 RAM 二者。

快速 RAM94 宜使用於旋轉一個幽靈 (Sprite)，利用執行在快速記憶體 94 內的一個循環。當它執行旋轉和尺寸大

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 28)

小變化的計算時，將從ROM10中讀出每個像素的色彩。它使用PLOT指令（將於下面說明），以寫入像素到RAM6，8。所有皆平行地發生，最慢操作將使其快速慢下來。最慢操作經常是ROM資料抓取，就是所設計的瑪利歐晶片，使用緩衝存取於ROM和RAM中。

當和相對地慢ROM10的執行相比較，一個程式在快速RAM94執行速度有6倍之快。但首先它必須從ROM載入到快速記憶體94內。執行這項工作係利用任何要收到快速記憶體的循環開始，放置一個指令。只有循環的第一個512位元組，從CACHE指令的位址取出和放入快速記憶體。當執行第一個重覆的循環碼時，程式從ROM10來，並且以16位元組拷貝到快速RAM中。所有進一步循環之重覆將來自快速RAM94，而非ROM10中。

CACHE指令使用於任何重覆程式循環的前面。只有循環的後續重覆從快速記憶體獲得好處。假如程式循環大於512位元組，並且超過快速記憶體94，它將仍正確地工作，只不過第一個512位元組從快速記憶體94執行，但其餘仍和平常一樣從ROM10執行。這可提昇部份速度，但並非理想的。

在最佳實施例有一個快速標識位元暫存器72，係快速控制器68的一部份，用於識別已載入快速RAM94記憶體位置。快速標識位元允許瑪利歐晶片快速地決定，是否從較快的快速RAM取得可執行的程式指示，而不是從程式ROM10。利用快速控制器68或超級NES經超級NES位址匯流

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明 (29)

排 HA 經多工器 96，以存取快速 RAM。

快速控制器 68 耦合到程式計數器匯流排 PC，以載入快速基底暫存器 70，並且執行快速記憶體位址超出範圍之檢查操作。

類似於從 ROM10 可完成的平行讀出，瑪利歐晶片亦提供平行寫到 RAM6，8 的方式。不論何時瑪利歐暫存器寫到 RAM6，8，它將起始，例如在 RAM 控制器 88 的一個分離 RAM 寫入電路，以執行記憶體變動。這典型地需 6 個週期，當它如此執行時，它將不延遲處理器。假如程式者在此時間避免另一個 RAM 變動。例如，在每個儲存指令間交叉間隔其它處理是較快的。以這方式 RAM 寫電路才有時間做它的工作。假如二個寫使用在一行內，當第一個在寫時，第二個將延遲處理器。

例如 (所使用的從指令集而來之指令，將在下面說明)

:

```

FROM   R8           ; 儲存 R8 到 (R13)
SM      (R13)
SM      (R14)       ; 儲存 R0 到 (R14)
TO      R1
FROM    R2
ADD     R3           ; 執行 : r1 = r2 + r3
TO      R4
FROM    R5
ADD     R6           ; 執行 : r4 = r5 + r6

```

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (30)

注意，二個儲存指令相互太靠近。第二個將需6個週期，因為RAM匯流排正忙於試著完成第一個儲存指令。

一個比較好寫程式的方式，係二個儲存指令之間有其它有用碼分開，則能達成較快的執行速度。例如：

```
FROM R8          ; 儲存 R8到 (R13)
SM      (R13)
TO      R1
FROM R2
ADD R3          ; 執行： r1=r2+r3
TO      R4
FROM R5
ADD R6          ; 執行： r4=r5+r6
SM      (R14)    ; 執行 R0到 (R14)
```

依此方式，同時平行執行幾個更多指令，第一個儲存指令結果寫入RAM，然後，第二個儲存指令在幾個週期後執行。

以下說明指令集，包括一個快速指令，係用於寫回一個暫存器到最後使用RAM位址。這允許“大量”資料處理，利用從RAM載入之值，在它上面做某些處理，然後再快速儲存回去。

如圖4B，一個立即資料門門74耦合到指令匯流排。此資料門門74允許指令它本身提供資料的來源，因此這指令不需指令來源暫存器。立即資料門門74的輸出耦合到目的地匯流排Z。依序地耦合到暫存器方塊76之一個預先決定暫

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (31)

存器。指令解碼電路60解碼一個“立即”資料指令，並且起始適當的傳送之執行到暫存器操作。

如圖4B所示，GETB暫存器98用於連接上述顯示/緩慢讀操作。在此方面，假如廣泛使用相對之低存取時間ROM，習知微處理器典型地必須等待，直到完成一個資料抓取，不論何時執行一個ROM。利用下述延遲/緩衝抓取機構，當資料抓取完成時，也可執行其它操作。根據該機構，假如在暫存器方塊76的暫存器R14以任何方式存取或修改，自動地起始ROM或RAM抓取，其位址則利用R14的內容所識別。

如圖4B所示，暫存器R14耦合到ROM控制控制104。在任何時間以任何方式修改暫存器R14的內容時，ROM控制器104操作俾起始一個ROM存取。存取ROM的結果經多工器102載入GETB暫存器98。耦合到ROM資料匯流排ROM D。指令識別以下允許存取緩衝在GETB暫存器98之資訊。這資訊經多工器100載入目的地匯流排Z，並且然後進入在暫存器方塊76的一個暫存器。

依此方式，假如從ROM抓取的資料，係已知其需多少個預先決定的處理週期。因此能起如抓取並且不用等待與執行其它操作。瑪利歐晶片能執行，例如，不相關碼，在此種資料抓取已起始後。GETB暫存器亦用於儲存從ROM6，8經多工器102所取出的資訊，如圖4B所示。

暫存器方塊76共有16個16位元暫存器(R0~R15)。暫存器R0~R13是一般目的地暫存器(雖然某些暫存器用於特殊目的，如下所說明)。如上所述，暫存器R14使用於當成讀

(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)

記憶體 的 指標，並且，當修改時，從 ROM(或 RAM)起始一個讀週期。所讀位元組儲存在一個暫時緩衝區(GET B暫存器 98)，係於較後利用一個 GET L或 GET H命令存取。暫存器 R15是程式設~~設~~₂₁數器。它指到將被抓取的下一個指令的開始。

暫存器 R0是一般目的暫存器，係典型地當成一個累積器。對於大部份單週期指令言，它亦預設來源和目的地暫存器。假如，例如，期望將 R0和 R4的內容相加在一起，它只需指示 R4暫存器。

R11，R12和 R13暫存器特別使用於執行一個循環指令。R13暫存器儲存將執行循環的最上面的位址，與 R12暫存器儲存將被執行循環的次數的數目。假如暫存器 R12的內容是非零的，然後在 R13的內容所指定住址之指令，被載入程式計數器(R15)且執行。在循環完成後，返回的位址儲存在暫存器 R11。

暫存器控制邏輯 78耦合到暫存器方塊 76，並且控制存取一般暫存器 R0到 R15。依將被執行特定指令的格式而言，指令解碼邏輯 60將指令 1或多個暫存器 R0~R15。暫存器控制邏輯 78指定暫存器，這些暫存器在執行下一個指令時，都將需要使用者。暫存器控制邏輯 78耦合適當暫存器的輸出，到 X 和 Y 匯流排。此外，如圖 4B所示，這適當暫存器 R0~R15從 Z 匯流排，於暫存器控制 78的控制下接收資訊。

ROM控制器 104接收從超級 NES位址匯流排 HA的位址，

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (33)

或瑪利歐晶片將存取該位址。ROM控制器104如圖13所示將近一步詳細說明。從ROM10所存取資訊，被載入快速RAM94，係用於快速指令執行。ROM和RAM控制器104，108只有匯流排仲裁單元，其仲裁超級NES與瑪利歐晶片之間的存取企圖。

底下將進一步說明，瑪利歐晶片亦使用狀態暫存器，(亦即，在暫存器方塊76或在RAM6，8內)，它可利用超級NES CPU存取，並且儲存旗標用於識別狀態條件，例如零旗標，進位旗標，正負號旗標，溢位旗標，“GO”旗標(當1指示瑪利歐晶片正執行中，以及0指示瑪利歐片停止)；一個ROM位元組抓取正在中的旗標(指示暫存器R14已被存取)。各種不同模式指示旗標，包括一個ACT1旗標，ACT2旗標；立即低位元組和立即高位元組旗標；與旗標指示來源及目前暫存器，已被一個“with”前置指令及一個中斷旗標設定1。

瑪利歐晶片代表方塊圖在圖4A與4B之形式，利用超級NES在一秒中內多次打開或關閉瑪利歐晶片以執行工作。起始時，當超級NES打開，儲存在ROM10的遊樂器程式開始啓動。須注意，在利用超級NES和瑪利歐晶片處理器之遊樂器程式執行之前，首先確認遊樂器卡匣。如實施例的方式，此確認的發生係利用起始而使超級NES CPU在一個重置狀態下，並且確認處理器執行相關於遊樂器卡匣和超級NES主控制板之確認程式。即根據美國專利案4,799,635的方式。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (34)

起始地，瑪利歐晶片係在關閉狀態。在這時間，超級NES具有無任何限制存取到遊樂器卡匣程式ROM，及遊樂器卡匣RAM。當超級NES需使用瑪利歐晶片處理能力，以執行繪圖操作或數學計算時，超級NES在卡匣RAM中，儲存給瑪利歐晶片處理的適當資料（或預先決定瑪利歐暫存器）；並且在瑪利歐晶片程式計數器中，載入將被執行瑪利歐程式的位址。利用瑪利歐晶片處理資料，也許是目標的預先決定X，Y座標資料，該目標需旋轉和放大或縮小。瑪利歐晶片能執行程式，其實施演算法以操縱不同數目移動目標的前景和背景。使用瑪利歐晶片速度增進硬體與軟體，以獲得此操作可在很高速度執行。

使用瑪利歐晶片處理幽靈，能大幅地擴張整體電視遊樂器系統的能力。例如，超級NES限制每個圖框顯示128個幽靈，但使用瑪利歐晶片可達到數百個幽靈顯示，以及例如旋轉。

當瑪利歐晶片完成超級NES所要求功能。執行一個STOP指令。並且產生中斷信號，並被傳輸到超級NES，以指示瑪利歐晶片完成它的操作，依序地指示它準備執行下一個工作。

亦可使用瑪利歐晶片執行小的工作，例如一個高速乘法工作，或者使用於畫出全銀幕的幽靈。在這工作中，超級NES自由地和瑪利歐晶片平行處理，假如超級NES遠離RAM或ROM匯流排，當匯流排瑪利歐晶片使用時。假如超級NES給瑪利歐晶片，在遊樂器卡匣上RAM和ROM匯流排的控

（請先閱讀背面之注意事項再填寫本頁）

.....裝.....訂.....線.....

五、發明說明 (35)

制下，則超級NES能執行其工作RAM32範圍外程式，如圖2所示。因此，整個系統的輸出增加，利用從程式ROM中拷貝將被執行的一個超級NES程式，到它的工作RAM中。同時，利用瑪琍歐晶片執行一個程式。

圖5所示的流程圖，代表著主CPU(例如超級NES CPU)執行一個“執行瑪琍歐”程式之操作順序，係用於起始瑪琍歐晶片，從ROM所需位址抓取和執行程式碼。圖5所代表常式，將典型地被超級NES CPU所執行，在如圖2所示之從程式ROM10拷貝常式到其工作RAM32，在任何時間，當要求瑪琍歐晶片執行一個操作時，則主CPU執行這常式。

在方塊125所示，當執行“執行瑪琍歐”主CPU常式，執行起始操作，包括保存超級NES暫存器。於起始步驟期間，該常式從程式ROM10拷貝到主CPU工作RAM32。

如方塊127所示，ROM10碼組儲存將被執行的瑪琍歐程式碼，載入到一個瑪琍歐晶片暫存器。此外，在碼組內實際位址儲存在瑪琍歐晶片銀幕畫面基底暫存器，如方塊129所示。

此後，如方塊131所示，在瑪琍歐晶片設定I/O輸入/輸出模式，以資識別是否使用4，16，或256色彩模式。此模式相關於主CPU操作之色彩模式。此外，模式設定用以定義銀幕畫面顯示字元數目的高度。

模式位元係被設定，將ROM和RAM匯流排的控制給瑪琍歐晶片。分離地可選擇ROM和RAM匯流排之控制，因此設定瑪琍歐片到一個模式，即它存取於ROM匯流排，RAM匯流排

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

214588

五、發明說明 (36)

或二者。因此，若是“瑪利歐擁有者”模式之設定於ROM和RAM，則主CPU不能從或到ROM或RAM讀式或寫。須注意，當瑪利歐晶片正在使用程式ROM匯流排，假如主CPU企圖存取程式ROM，則有一個機構使得瑪利歐晶片傳回假位址給超級NES。跳到此位址而保持超級NES佔有，直到瑪利歐晶片不再需存取卡匣ROM匯流排。

如方塊133所示，在瑪利歐晶片程式計數器載入一個位址，此位址儲存著瑪利歐常式需執行的第一個指令後，瑪利歐晶片開始操作。

然後，主CPU等待從瑪利歐晶片(方塊135)來的一個中斷信號。當接收一個中斷信號時，通知超級NES瑪利歐晶片已完其操作，並且已停止(方塊137)。假如未接收到此中斷信號，則主CPU繼續等待一個中斷(方塊135)。超級NES也許在這週期間，平行地和瑪利歐晶片執行程式碼，利用執行它的工作RAM32範圍之外，如圖2所示。

接著，超級NES檢查狀態暫存器(例如瑪利歐晶片暫存器方塊76)，以決定是否已設定瑪利歐晶片“GO”旗標，此旗標指示瑪利歐晶片係在操作中(137)。此外，在瑪利歐晶片狀態暫存器設定一個中斷旗標，用以指示主CPU所接收之中斷信號，瑪利歐晶片是以中斷信號的來源。因此，在主CPU(135)接收一個中斷信號後，測試適當的瑪利歐狀態暫存器，以決定是否瑪利歐晶片為中斷的來源(相對於例如一個垂直熄滅間隔的中斷信號)。假如瑪利歐晶片已停止(137)，然後對ROM和RAM瑪利歐擁有者模式位元被

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (37)

清除。並且超級NES完全存取於ROM和RAM。超級NES離開常式(141)，返回到其程式進入執行瑪利歐常式之前的地方。

當CPU22遊樂器程式將瑪利歐晶片轉成ROM瑪利歐擁有者模式。它必需自願停止存取該ROM。不論何時，CPU22為相同理由需存取ROM時，它只有關閉ROM瑪利歐擁有者模式。瑪利歐晶片將自動停止，當其下一之次需存取此ROM，直到它再回到ROM瑪利歐擁有者模式時。假如從內部快速RAM執行，也許這一點也不需要。

如果瑪利歐晶片是在瑪利歐擁有者模式使用於ROM，CPU22遊樂器程式甚至不從ROM讀任何事甚為重要。例如，由於垂直熄滅，它引起一個NMI，然後CPU22自動地止企圖從ROM抓取其中斷向量。這是不期望的，因為CPU22已明顯告訴瑪利歐晶片，它將遠離ROM，並且發生一個中斷，，然後從ROM抓取。在此狀況下，從CPU22存取一個ROM，雖然是在瑪利歐擁有者模式下，將使得瑪利歐晶片假定這是一個中斷向量要求。

於ROM瑪利歐擁有者模式中下一個中斷向量抓取期間，瑪利歐晶片將再配置中斷向量，進入超級NES內部工作RAM32於堆疊地區的底部。例如，假如平常中斷向量是\$00:FFEC，然後它將產生一個跳躍到位置\$00:oloc。類似地，從\$00:ffex之所有中斷向量，將使得CPU22跳躍到\$00:olox之相關位置。這技術避免CPU22存取ROM10，當它們不應如此時，並且使其轉到板上超級NES RAM32。須注

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (38)

意，RAM基中斷向量需包含跳躍或跳開到中斷管理器，亦即，儲存著實際碼，而不只是向量位址。當瑪利歐晶片不在瑪利歐擁有者模式時，使用正常ROM斷向量，最好指到這些位置保持相同位址，當此RAM基中斷向量走到相同地方時。

指令集

瑪利歐晶片指令集，提供一個有效方式，俾用於程式化高速繪圖與其它處理演算法。以下將簡短說明某些指令，並說明不同指令所使用的某些暫存器。亦包括指令集的指令詳細列表。

指令是8位元指令，並且典型地在一個時鐘信號週期執行。當，指令利用8位元前置指令修改。瑪利歐晶片指令集，包括一個獨特暫存器強制(override)系統，允許程式者在任何指令前指定目的地和來源暫存器。若設有這種“強制”前置，指令將只在疊積器上操作。因此，指令集係可變長度指令集，並且具有無數個組合。有某些基本指令是一個位元組長，能在一個週期中執行。利用提供前置指令，一個程式者能延伸指令的功能。一個指令可以是8，16或24位元，依程式者的意願而定。

瑪利歐處理器使用指令起始高速度、在板上快速RAM程式執行，和延遲/緩衝I/O到記憶體。經單一週期像素畫圖指令的使用，能有效率地繪圖處理，此指令所引起始操作使用上述像素畫圖硬體。

在識別瑪利歐指令集之前，執行指令時，處理器設定

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 §9)

或存取各種不同記憶體映至暫存器。如下說明。首先識別狀態旗標；暫存器。狀態暫存器是一個16位元暫存器，和相關於每個暫存器16位元之旗標，如下所述：

狀態旗標暫存器16位元

位元	旗標	
0	-	保留
1	Z	零旗標
2	C	進位標旗標
3	S	正負號旗標
4	V	溢位旗標 ([位元 14 進入 15] XOR [15 到 進位])
5	g	GO 旗標：1 瑪利歐晶片執行中 ：0 停止
6	r	(R14) 正在進行 ROM 位元組抓取
7	-	保留

“GO”旗標(位元5)被設定為“1”狀態，以指示瑪利歐晶片正在執行，並且一個“0”狀態指示瑪利歐晶片停止，(其引起一個中斷信號的產生，此信號耦合到超級NES)。利用超級NES處理器檢查這旗標。設定位元6指示目前正在進行一個ROM位元組抓取。如下所列的GET位元組指令不能執行，直到此旗標清除，以指示已完成資料抓取。這些狀態暫存器的較高位元，可以利用瑪利歐晶片處理或主CPU單獨地讀出，或組合其餘8位元。設定狀態旗標暫存器的較高位元，利用預先決定前置指令，與定義不同模式

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (40)

的指令編譯。

位元	模式	
8	alt1	變換 (ADD->ADC, SUB->SBC等)
9	alt2	變換 (ADD->ADD#, SUB->SUB#等)
10	il	立即位元組低 (在 ih 之前做)
11	ih	立即位元組高 (低位元組緩衝直到 hi 準備)
12	b	SReg & DReg 二者設定, 利用 WITH 設定。
13	-	保留
14	-	保留
15	irg	中斷旗標

在如上所識別 ACT1 模式中, 一個 ADD 指令將被編譯成一個 ADD 具有進位。並且一個 SUBTRACT 指令將編譯成 SUBTRACT 具有進位。一個指令 ACT1 起始該模式。

一個 ACT2 指令修改 ADD 指令的編譯, 成為 ADD 具有立即資料, 並修改 SUBTRACT 成為 SUBTRACT 具有立即資料。這“立即”資料在指令後立即位元組。須注意, 指令 ACT3 將設定位元 8 和 9 成為邏輯“1”的準位。位元 10 和 11 之設定依是否此立即資料為立即高位元組或立即低位元組而定。狀態暫存器的位元 12 定義一個“6”模式, 其中利用一個前置指令“WITH”, 以設定來源和目的地暫存器。狀態暫存器的位元 15 儲存瑪利歐中斷信號, 其在瑪利歐停止執行後被設定。

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (41)

瑪利歐晶片包括許多暫存器，除上述狀態暫存器。如上所述，瑪利歐晶片包括16個16位元寬之暫存器，如圖4A與4B在暫存器方塊76所說明的。大部份這些暫存器係一般目的地暫存器，並且能於資料或位址儲存。但，暫存器R15都是一直使用成程式計數器。典型地，暫存器有二個目的，使用於和主CPU通信，以及使用於控制程式的執行。此外，在瑪利歐晶片所使用的其它暫存器，其功能如下表所述。

暫存器	特別功能
r0	預設 DReg 與 SReg
r1	PLOT指令的 X 座標
r2	PLOT指令的 Y 座標
r3	無
r4	低位元組的 LMULT指令結果
r5	無
r6	字組乘法使用於 FRMOLT與 LMOLT指令
r7	使用於 MEREG指令之來源 1
r8	使用於 MERGE指令來源 2
r9	無
r10	無
r11	使用於副常式呼叫之連接暫存器
r12	LOOP指令的計數
r13	LOOP指令跳躍的位址
r14	ROM位址，當修正從ROM讀出的一個

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (42)

	起 始 位 元 組
r 15	程 式 計 數 器
其 它 暫 存 器	
8 位 元 PCBAK	程 式 碼 組 暫 存 器
8 位 元 ROMBAK	程 式 資 料 ROM組 暫 存 器 , 64K組
8 位 元 RAMBAK	程 式 資 料 ROM組 暫 存 器 , 64K組
16位 元 SCB	銀 幕 畫 面 基 底
8 位 元 NBP	位 元 平 面 的 碼 號
8 位 元 SCS	銀 幕 畫 面 列 大 小 選 擇 :
	256, 320, 512, 1024, 1280
	(銀 幕 畫 面 16 & 20字 元 高 , 在 2, 4
	& 8位 元 平 面)

瑪 珞 歐 晶 片 亦 包 括 一 個 色 彩 模 式 CMODE暫 存 器 。 在 這 暫 存 器 的 4 個 位 元 , 係 用 於 本 實 施 例 中 , 創 造 如 下 所 述 特 殊 效 應 。 利 用 設 定 一 個 CMODE暫 存 器 位 元 變 化 , 創 造 該 等 效 應 , 係 基 於 是 否 設 定 16或 256色 彩 解 析 度 模 式 , 其 例 如 下 說 明 之 。

以 下 是 CMODE暫 存 器 位 元

CMODE位 元 0

畫 圖 色 彩 0 位 元 (NOT透 明 性 位 元)

於 16色 彩 模 式 :

假 如 位 元 0=1, 並 且 所 選 擇 色 彩 半 位 元 組 =0, 則 不 畫 圖 。

於 256色 彩 模 式 和 位 元 3=0:

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (3)

假如位元 0=1，並且色彩位元組=0，則不畫圖。

於 258 色彩模式和位元 3=1：

假如位元 0=1，並且色彩低半位元組=0，則不畫圖。

N.B. 透通 ON=0

透通 OFF=1

只有使用透明性 OFF，利用 0 充滿一個地區（使用於清除銀幕畫面）

CMODE 位元 1

抖動位元

抖動在 16 色彩模式。（高/低半位元組給 2 個色彩）

假如 $(xpos \text{ XOR } ypos \text{ AND } 1) = 0$ ，則選擇低半位元組

假如 $(xpos \text{ XOR } ypos \text{ AND } 1) = 1$ ，則選擇高半位元組。

假如透明性是開，並且所選擇色彩半位元組是零，則不畫圖。

於 258 色彩模式，抖動無效應。

CMODE 位元 2

高半位元色彩位元

於 16 色彩模式或 256 色彩模式，並且 CMODE 位元 3 設定。

當此位元設定，COLOUR 命令設定色彩暫存器的低半位元組，至來源位元組的高半位元組。（使用未緊縮 16 色彩儲存幽靈，當高半位元組為另一個幽靈）。

假如色彩暫存器的低半位元組是零，則不畫面，且透明性打開。

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明 (44)

CMODE位元3

複雜位元

只有256色彩模式。當設定此位元，則鎖定色彩的高半位元組。並且COLOR命令只改變低半位元組。只從低半位元組計算其透明性。

於正常256色彩模式，假如是開，則從所有位元計算透明性。

；16色彩模式例子

ibtr ro, \$C0

colour ; 設定色彩 \$C0

ibtr ro, %0000 ; 設定0

cmode

ibtr ro, \$97

colour

plot ; 畫圖色彩 \$7

ibtr ro, \$30

colour

plot ; 無畫圖，當色彩是 \$0

；透明性開並且低半位

元組 = 0)

ibtr ro, %0001 ; 設定位元1

cmode

ibtr ro, \$40

colour

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (45)

plot ; 畫圖色彩 \$ 0

; (透明性關)

STOP

; 16色彩模式，位元2設定例子

ibt ro, \$ C0

colour ; 設定色彩 \$ C0

; 256色彩模式，位元3設定例子

ibt ro, \$ C0

colour ; 設定色彩 \$ C0

ibt ro, % 1000 ; 設定位元3

cmode

ibt ro, \$ 47

colour

plot ; 畫圖色彩 \$ C7

ibt ro, \$ 50

colour

plot ; 當色彩是 \$ C0，不畫圖

; (透明性開並且低半位元組 = 0)

ibt ro, % 1001 ; 設定位元3與位元1

cmode

ibt ro, \$ 60

colour

polt ; 畫圖色彩 \$ C0

; (透明性開)

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (46)

STOP

; 256色彩模式，設定位元3與位元2例子

ibt ro, \$ C0

colour ; 設定色彩 \$ C0

ibt ro, % 1100 ; 設定位元3與位元2

cmode

ibt ro, \$ 74

colour

plot ; 畫圖色彩 \$ C7

ibt ro, \$ 03

colour

plot ; 當色彩是 \$ C0，不畫圖

; (透明性開與低半位

; 元組 = 0)

ibt ro, % 1101 ; 設定位元3，位元

; 2與位元1

cmode

ibt ro, \$ 08

colour

plot ; 畫圖色彩 \$ C0

; (透明性開)

stop

許多瑪琍歐晶片暫存器有關特別功能如上表所示，假

如沒有其它指定，系統須設到暫存器R0，當成目的地暫存器

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 47)

或來源暫存器依一個特定指令之所需者。R0暫存器亦使用成一個ALU疊積器。乘法指令，如上所述，返回一個32位元結果。最低16位元儲存在暫存器R4中。暫存器R6使用於相關一個分數正負號乘法指令(FRMULT)，與一個長乘法指令(LMULT)。

暫存器R7與R8使用於執行MERGE指令。這指令使二個預先決定暫存器(亦即，暫存器R7，R8)，相互結合在一起，以形成幽靈座標資料。以座標資料用於定址一個ROM表，其映至一個預先決定幽靈到一個預先決定多邊形。該指令是幫助有效執行紋理映到操作，利用組合部份的二個暫存器，以定義下一個像素的色彩位址，這像素被包含在靈內映至到一個多邊形。

使用暫存器R11到R13以控制副常式執行。暫存器R11當成副常式呼叫的連結暫存器，並且儲存程式計數器加1的內容。暫存器R11的內容定義在一個循環完成後必需存取之位址。使用暫存器R12儲存一個計數，以定義將被執行循環的次數，循環之位址儲存在暫存器R13。

如上所述，不論何時修改暫存器R14的內容，一個位元組從ROM10在暫存器R14的所儲存位址讀出。於此方式，實施一個延遲或緩衝READ操作，相關於如下識別的GET位元組指令。

參照上表之“其它暫存器”。在程式ROM中將被執行程式位置，使用一個24位元址定址。此位置的最低16位元在程式計數器中發現。最高位元定義程式組儲存在一個程

(請先閱讀背面之注意事項再填寫本頁)

· · · · · · 裝 · · · · · · 訂 · · · · · · 線 · · · · · ·

五、發明說明 48)

式碼組(PC組)暫存器。

ROM組暫存器(ROMBANK)儲存最高位元，用於允許定址儲存在ROM10之程式資料，並且附加到儲存在暫存器R14之16位元ROM位址。類似地，RAM組暫存器(RAMBANK)儲存較高位址位元，使用於允許瑪琍歐晶片處理器，以存取在RAM中之程式資料。使用RAM和ROM組暫存器的內容，相關於瑪琍歐晶片ROM和RAM存取指令，係用於有效延伸瑪琍歐處理器的位址範圍。

銀幕畫面基底暫存器(SCB)用於儲存目標的虛位元映至的位址。此目標能被創造，與旋轉，放大及減少。當執行一個PLOT像素指令時，銀幕畫面基底暫存器SCB，儲存在RAM位址，其被存取並寫入資訊。

暫存器NBP係用於儲存位元平面的號碼。典型地，指示使用2，4或8平面。此外，使用一個銀幕畫面列大小暫存器SCS，以指定相關於虛位元映至的資訊，以資訊係以在列所包含字元數目表示。

以下所列的瑪琍歐晶片指令集，指令簡字符號(mnemonic)，並且相關於解碼有關指令所執行之功能。首先簡短註解說明一個相關指令的某些功能。

當瑪琍歐晶片完成它的操作時，執行stop指令，並且操作設定“GO”旗標為零，產生任何中斷信號到主CPU。

CACHE指令操作以定義一部份程式ROM，其被拷貝進入瑪琍晶片快速RAM，並且從那裡執行。當執行CACHE指令時，程式計數器的內容載入快速基底暫存器，和以下說明的

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (49)

快速標識被重置。

瑪喇歐晶片包括一連串延遲跳躍指令，執行在跳躍後指令當成下表之指示。跳躍發生的位址相對於程式計數器的內容。指令集基於如下表所說明條件，有各種不同廣泛的延遲跳躍。

瑪喇歐晶片包含多個“前置”指令，亦即 to/with/from。這些前置指令隱含後續指令一個資料分配。例如“TO”前置設定目的地暫存器(DReg)，使用於下一個指令。“FROM”前置設定來源暫存器(SReg)，使用於下一個指令。“WITH”前置設定二者。

大部份指令具有一個第二來源暫存器在此運算碼。假如不利用前置指令設定SReg和DReg，則它們將預設到R0。在每個指令後，SReg和DReg設定到R0。假如DReg設定到R15，程式計數器使得下一個指令儲存其內容在R15，然後起始一個週期延遲跳躍。

其它前置指令設定高位元組的狀態暫存器之旗標，以改變如下指令之操作。所有非前置指令清除高位元組的狀態字組。下列例子，係如何經前置指令修改後續指令。

```

1 sr                ; r0=r0向右移 1
to r4               ; r4=r0向右移 1
1sr
from r4             ; r0=r4向右移 1
1sr                ; r0=r4向右移 1
alt 1

```

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (50)

from r6

to r5

add r7 ; r5=r6+r7+進位

alt 1

with r3

add r3 ; r3=r3+r3+進位 (6502 rol)

如果在狀態暫存器設定“b”旗標，修正“T0”指令當成一個“MOVE”指令操作。T0指令指定資訊移到暫存器，及From指令資訊來源。

STW指令儲存一個特定字在一個緩衝區內，它不需等待，在執行下列指令之前直到完成一個儲存操作。依此方式，一個RAM的使用慢於處理器，不要不必需地使處理器慢下來。

LOOP指令的執行減少一般暫存器R12的內容。假如R12的內容不是零，則起始跳躍到R13所指定位址。

Alt1, Alt2和Alt3是前置指令，其設定在狀態暫存器的上述旗標，因此使得執行指令以不同方式編譯，如下表所示。

PLOT指令識別將畫圖像素X和Y銀幕畫面座標。並且畫出利用COLOR指令所指定色彩，相對於X和Y座標上銀幕畫面位（如暫存器R1和R2所示）。PLOT像素指令包括R1內容的自動增加，係協助於高速度畫出水平線，並且免除包含一個額外增加指令。

設定Alt1旗標，然後編譯PLOT指令，係一個READ指令

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (51)

PIXEL指令(RPIX)。利用執行讀像素指令RPIX，在特定銀幕畫面位置之像素的色彩讀出，亦可使用於從硬體清除不要的像素資訊。

讀像素指令RPIX，主要使用畫圖硬體從字元矩陣反向讀出，以決定指令所指定之一個特定像素的色彩。COLOR指令提供到色彩硬體，利用一個特定來源暫存器的內容而定義下一個像素的色彩。

"CMODE"指令設定色彩模式，與能使用於產生不同特別效應，如上例所證明者。例如，使用CMODE指令產生一個抖動效應，在其另一個像素變質不同色彩，以產生一個遂漸變化效應。CMODE指令能使用於控制透明性，因此一個幽靈先顯示然後將畫出背景顯示。透明性的決定係利用一個色彩模式相關旗標的設定，如上例所示。

指令集亦包括一個小數正負數乘法，係用於旋轉多邊形的計算，以決定所顯示目標的梯度或斜度。

增加指令，若使用於相關暫存器R14，將從ROM起始一個讀。GETC指令將從ROM存取位元組，並且載它進入色彩暫存器。

下表指定出依據本發明實施例之一個瑪琍歐晶片指令集，這些指令在上文已討論。

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 5(2)

指 令 集

十六進位	簡字符號	功 能
\$ 00	STOP	停止瑪琍歐晶片，並且產生 65816 IRQ, $g=0$
\$ 01	NOP	1 週期無操作
\$ 02	CACHE	設定快速基底到 PC 與重置快 速旗標。(只有如果 PC 不等 於現在快速基底) 假如快速基底 $\neq r15$, 則快速基底 $= r15$, 重置快速旗標。
\$ 03	LSR	邏輯向右移, $DReg = DReg$ LSR1。
\$ 04	ROL	進位向左從轉 $DReg = SReg \text{ ROL } 1$
\$ 05nn	BRA sbyte	必定相對地延遲跳躍 $r15 = r15 + \text{正負號位元組偏移}$
\$ 06nn	BGE sbyte	假如大於或等於, 則相對的 延遲跳躍, 假如 $(s \text{ XOR } v) = 1$, 則 $r15 =$ $r15 + \text{正負號位元組編移}$ 。
\$ 07nn	BLT sbyte	假如少於, 則相對的延遲跳 躍, 假如 $(s \text{ XOR } v) = 0$, 則 $r5 = r15$ + 正負號位元組偏移。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

(請先閱讀背面之注意事項再填寫本頁)

假如相等，則相對的延遲跳躍，

假如 $z=1$ ，則 $r15=r15+$ 正負
號位元組偏移。

假如不相等，則相對的延遲跳躍

假如 $z = 0$ ，則 $r15 = r15 +$ 正負號位元組偏移

假如是正的，則相對的延遲跳躍，

假如 $s=0$ ，則 $r15=r15+$ 正負號
位元組偏移

假如是負的，則相對的延遲跳躍

假如 $s=1$ ，則 $r15=r15+$ 正負
號位元組偏移

假如進位旗標清除，則相對的延遲跳躍

假如 $c = 0$ ，則 $r15 = r15 + \text{正負號位元組偏移}$

假如進位旗標設定，則相對的延遲跳躍

假如 $c=1$ ，則 $r15=r15+正負$
號位元組偏移

五、發明說明 (54)

\$ 0e nn	BVC sbyte	假如溢位清除，則相對的延遲跳躍
		假如 v=0，則 r15=r15+正負號位元組偏移
\$ 0f nn	BVS sbyte	假如溢位設定，則相對的延遲跳躍
		假如 v=1，則 r15=r15+正負號位元組偏移
\$ 10-\$ 1f	TO r0... r15	(前置)設定 DReg to rn (目的地暫存器使用於下一個 op) DReg=rn
ifb:	MOVE	rn=SReg(無旗標設定)
\$ 20-\$ 2f	WITH r0-r15	(前置)設定 DReg與 SReg到 rn DReg=rn SReg=rn b=1
\$ 30-\$ 3b	STW(rn)	在 rn位址儲存 SReg RAM[rn]=SReg[低字元組/高緩衝] [正常地字元組在偶數位址]
if altl:	STB(rn)	在 rn位址儲存低位元組的 SReg RAM[rn]=SReg.1[位元組緩衝]

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (55)

\$ 3c:	LOOP	減 r12, 並且假如 r12<>0, 則延遲跳躍到在 r13 的位址。 r12=r12-1 假如 r12<>0, 則 r15=r13 (TO/WITH/FROM 忽略)
\$ 3d	ALT1	(前置) 設定 alt1 旗標 alt1=1
\$ 3e	ACT2	(前置) 設定 alt2 旗標 alt2=1
\$ 3f	ACT3	(前置) 設定 alt1 與 alt2 旗標 alt1=1 alt2=1
\$ 40 ~ \$ 4b	LDW(rn)	從在 rn 位址載入到 DReg, DReg=RAM[rn] (字元組高/低等待) (正常地字元組偶位址)
if alt1:	LDB(rn)	從在 rn 位址載入到 DReg, (無正負號位元組) DReg.h=0 DReg.l=RAM[rn][位元組等待]
\$ 4c	PLOT	在 r1, r2(x.g) 畫圖像素, 並且增加 r1 (將不檢查 N.B. r1 與 r2 在銀幕畫面上, 並且將畫在 RAM 任何地方)

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (56)

		<code>plot(r1,r2),r1=r1+1</code>	
<code>if alt1:</code>	<code>RPIX</code>		在 <code>r1,r2(x,y)</code> 讀出像素的色彩
		<code>DReg=point(r1,r2)</code>	
<code>\$ 4d</code>	<code>SWAP</code>		互換位元組
		<code>DReg.h=SReg.l</code>	
		<code>DReg.l=SReg.h</code>	
<code>\$ 4e</code>	<code>COLOUR</code>		設定 PLOT 色彩，畫圖色彩 =
		<code>SReg</code>	
<code>if alt1:</code>	<code>CMODE</code>	<code>DReg=NOT SReg</code>	
			設定 PLOT 色彩，畫圖色彩 =
		<code>SReg</code>	
<code>\$ 50 ~ \$ 5f</code>	<code>ADD r0.r15</code>	<code>DReg=SReg+rn</code>	
<code>if alt1:</code>	<code>ADC</code>	<code>DReg=SReg+rn+c</code>	
<code>if alt2:</code>	<code>ADD</code>	<code>DReg=SReg+#n</code>	
<code>if alt1+alt2</code>	<code>ADC</code>	<code>DReg=SReg+#n+c</code>	
<code>\$ 60 ~ \$ 6F</code>	<code>SUB r0.r15</code>	<code>DReg=SReg-rn</code>	
<code>if alt1:</code>	<code>SBC</code>	<code>DReg=SReg-rn-c</code>	
<code>if alt2:</code>	<code>SUB</code>	<code>DReg=SReg-#n</code>	
<code>if alt1+alt2</code>	<code>CMP</code>	<code>SReg-rn</code>	(零旗標、正負號
			旗標、進位旗標、溢位旗
			標)
<code>\$ 70</code>	<code>MERGE</code>		結合 <code>r7</code> 與 <code>r8</code> 的高位元組成
			為 <code>DReg</code>

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (57)

DReg.h=r7.h, DReg.l=r8.

h

旗標有如下結果設定：

s=b15 OR b7

v=b14 OR b6 OR s

c=b13 OR b5 OR v

z=b12 OR b4 OR c

\$ 71~ \$ 7f AND r1.r15

DReg=SReg AND rn

if alt1: BIC

DReg=SReg AND NOT rn

if alt2: AND

DReg=SReg AND #n

if alt1+alt2 BIC

DReg=SReg AND NOT #n

\$ 80~ \$ 8f MULT r0.r15

DReg=SReg*Rn(有正負號 8
× 8 位 元)

if alt1: UMULT

DReg=SReg*Rn(無正負號 8
× 8 位 元)

if alt2: MULT

DReg=SReg*#n(有正負號 8
× 8 位 元)

if alt1+alt2 UMULT

DReg=SReg*#n(無正負號 8
× 8 位 元)

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

十六進位	簡字符號	功能
\$ 90	SBK	儲存 SReg 回到最後使用 RAM 位址
\$ 91 - \$ 94	LINK1-4	連結返回位址到 r11 $r11 = r15 + 1 \cdots 4$
\$ 95	SEX	正負號延伸低位元組到字元組 $DReg.[b15-b17] = SReg.[b7]$ $DReg.1 = SReg.1$
\$ 96	ASR	算術向右移 $DReg = SReg \text{ ASR } 1$
if alt1:	DIV2	除2捨低位元 $DReg = SReg \text{ ASR } 1$ 假如 $DReg = 1$, 則 $DReg = 0$
\$ 97	ROR	進位旗標向右旋轉 $DReg = DReg \text{ ROR } 1$
\$ 98 - \$ 9d	JMP r8-r13	跳躍到 r13 位址, $r15 = rn$ (延遲跳躍)
if alt1:	LJMP	長跳躍到 rn 位址, (ROM 組從 SReg) 並且重置快速記憶體 $r15 = rn$ (延遲跳躍) 程式 ROM 組暫存器 = SReg

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (59)

\$ 9e	LOB	低 位 元 組
		DReg.h=0
		DReg.l=SReg.l
\$ 9f	FMULT	分 數 有 正 負 號 乘 法 ,
		DReg=(SReg*r6).hw
		(正 負 號 16*16 位 元 乘 法) ,
		c=(SReg*r6).b15
if alt1:	LMULT	長 正 負 號 乘 法 , DReg=(
		SReg*r6).hw
		(正 負 號 16*16 位 元 乘 法)
		r4=(SReg*r6).lw
		c=(SReg*r6).b15
\$ a0-\$ af nn	IBT r0-r15,	載 入 rn 具 有 信 號 延 伸 位 元
sbyte		組
		rn=立 即 位 元 組 (正 負 號 延
		伸
if alt1:	LMS r0-r15,	從 絕 對 移 位 位 元 組 位 址 載
byte		入 rn
		rn=RAM[位 元 組 <C1] (字 組
		資 料)
if alt2:	SMS r0-r15.	儲 存 rn 到 絕 對 移 位 位 元 組
byte		位 址
		RAM[位 元 組 <C1]=rn (字 組
		資 料)

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (60)

\$ b0 - \$ bf	FROM r0-r15	(重置) 設定 SReg=rn SReg=rn
ifb:	MOVES	SReg=rn (零旗標、正負號旗標溢 位旗標(正負號低位元組) 高位元組, DReg.h=0 DReg.l=SReg.l
\$ c0	HIB	DReg=SReg OR Rn
\$ c1 - \$ cf	OR r1-r15	DReg=SReg XOR Rn
if alt1:	XOR	DReg=SReg OR #n
if alt2:	OR	DReg=SReg Xor #n
if alt1+alt2	XOR	增加 rn rn=rnt1 (忽略 TO/WITH/FROM)
\$ d0 - \$ de	INC r0-r14	從 ROM 緩衝器獲得位元組 到 PCOT 色彩
\$ df	GETC	RAM 資料組暫存器 = SReg
if alt2:	RAMB	RAM 資料組暫存器 = SReg
if alt1+alt2	ROMB	減 rn, rn=rn-1 (忽略 TO/ WITH/FROM)
\$ e0 - \$ ee	DEC r0-r14	從 ROM 緩衝區獲得無正負 號位元組到 DReg DReg=ROM 緩衝位元組, 零 延伸
\$ ef	GETB	

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (61)

if alt1:	GETBH	從 ROM 緩衝區獲得到 DReg 的高位元組 DReg=ROM 緩衝區位元組， 及低混合 DReg=(SReg & \$ FF)+(位 元組 <C8)(使用 WITH)
if alt2:	GETBL	從 ROM 緩衝區取得送到 DReg 的低位元組 DReg=ROM 緩衝位元組，及 高混合(使用 WITH)
if alt1+alt2	GETBS	從 ROM 緩衝區取得正負號 位元組送到 DReg DReg=ROM 緩衝位元組，正 負號延伸
\$ f0-\$ ffnnnn	IWT r0-r15,	載入立即字組到 rn rn=立即字組(緩衝)
字組		
if alt1:	LM r0-r15,	從絕對字組位址載入 rn rn=RAM[字組位址](字組 資料)
字組		
if alt2:	SM r0-r15,	儲存 rn 到絕對字組位址 字組

圖 6 到圖 17 顯示圖 4A 與 4B 之進一步詳細組件的方塊圖。欲更清楚表現本發明獨特之特色，對熟悉該行業專業人士所習知電路，若有混淆此獨特特色者，於以下圖中將不

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (62)

予顯示。

如圖 6 所示，在 ALU 單元 50 之範例算術和邏輯單元。ALU 50 如圖 4A 及圖 6 所示，耦合到 X，Y 和 Z 匯流排。因此，瑪利歐晶片一般暫存器 R0 到 R15 耦合到 ALU。

ALU 50 利用一個 16 位元加法器/減法器 152，以執行加法與減法功能。ALU 50 亦包括傳統“AND”邏輯電路 154，“OR”邏輯電路 156，以及“互斥 OR”邏輯電路 158。

ALU 亦包括傳統移位功能電路，其中任何進位旗標位元移到最高位元位置。並且其結果經由線 160 耦合到多工器 164 的一個輸入端。此外，ALU 50 執行傳統位元組交換操作，因此，在匯流排上最低位元組和最位元組也可以交換。並且其結果在線 162 上到多工器 164。X 和 Y 匯流排耦合到電路 152、154、156 和 158，如圖 6 所示。

從加法器/減法器 152，電路 154、156、158，移位暫存器，和交換功能等輸出，耦合到 16 位元，6 個輸入一到一個“結果”多工器 164。依所解碼結果而定，其適當結果係輸出到目的地匯流排 Z。

加法器/減法器 152，除由 X 匯流排接收 16 位元外，亦接收在 X 匯流排上向前地資訊，或指令其本身之資訊，依指令解碼器輸入到多工器 150 而定。

此外，ALU 50 包括一個 CPU 旗標產生電路 166。CPU 旗標電路 168 產生零溢位、正負號、與進位信號，使用於載入到電路 166 內至少一個旗標暫存器。從指令解碼電路 60 設定 CPU 旗標，此電路解碼指令所產生的進位旗標致能，零

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (63)

旗標致能，正負號旗標致能與溢位旗標致信信號。使得旗標設定依利用加法器/減法器152所決定之相關條件而定。旗標亦被基於輸入到旗標電路166之目的匯流排Z的內容（或結果）而設定。使用旗標，例如，基於一個廣範圍的條件，而觸發條件跳躍之操作。

圖7、8A與8B係圖4A的圖素畫電路（52、54、56和58）之進一步說明。該電路執行PLOT命令，此命令取得一個特定X和Y座標，並且在這銀幕畫面座標上畫一個像素，此像素色彩是利用COLOR命令載入色彩暫存器54的內容所指定的。

須注意，超級NES使用一個字元映至顯示畫面。畫圖硬體操作轉換像素座標位址資料，成為字元映至位址資料。

超級NES字元定義在位元平面。字元可以是2，4或8位元平面，用於定義4，16或256色彩。字元定義的每位元組，包括一個位元平面的一列像素的字元，定義像素從左到右，高位元到低位元。對一個256色彩模式的操作，需更新8個RAM位置。

像素畫圖硬體包括一個當地緩衝機構，該機構包括一個色彩矩陣206，其儲存將被顯示為一個特定位元組之所有位元，因為所有這些位元最終均需更新。一個位元平面計數器28耦合到色彩矩陣電路208。像素座標從X和Y匯流排，載入到畫圖X與畫圖Y暫存器202，204。於目前實施例中，一般暫存器R1與R2，使用成如圖7所示畫圖X暫

（請先閱讀背面之注意事項再填寫本頁）

· · · · · 裝 · · · · · 訂 · · · · · 線 · · · · ·

五、發明說明 (64)

存器 202 與畫圖 Y 暫存器 204。這些暫存器像素的 X 和 Y 座標，如利用 PLOT 命令所指定者。

畫圖 X 與 Y 暫存器 202，204 耦合到全及半加法器，係基於字元位址計算電路，輸出於位址上到一個位置筒移位電路 214，其依序耦合到一個畫圖位址暫存器 216 與一個位址比較器 218。畫圖 X 暫存器的三最低位元耦合到一個解多器 212，其依序耦合到一個位元未定暫存器 210。

如圖 8A 所示畫圖控制器 200，接收信號指示已解碼一個 PLOT 像畫 (PLOT) 或 READ 像素命令，與如下所述之其它控制信號。畫圖控制器 200，產生畫圖電路控制信號，以如下所述方式而使用。

如上所示，畫圖控制電路 200，產生控制信號用於像素畫圖硬體 52 內。如圖 8A 所示，像素控制電路 200，從位元未定暫存器 210 接收輸出，此輸出經 AND 閘 201，耦合到像素控制電路 200。如果位元未決暫存器 210 的所有 8 位元被設定，則通知此像素控制邏輯 200，跳過一個讀週期，並且在色彩矩陣 206 之資訊寫出至 RAM。

像素控制電路 200 亦響應於 PLOT 命令，以起始其操作。像素控制邏輯 200 亦響應於 READ 像素命令 PR1X，以起始完全相同之操作，除新資訊未寫到色彩矩陣 206，使用於輸出到 RAM 外。如上所述，假如需知道銀幕畫面上一個特定像素的色彩可執行此 READ 像素命令，並且亦可用於清除色彩矩陣上之現在資訊。

控制器 200 並接收一個 RAM 完成控制信號 RAMDONE，用

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (65)

於指示該RAM存取已完成。RAM完成信號，如上所述，亦於增加以位元平面設計數器208，以識別在色彩矩陣206的一個位元平面。畫圖控制器200從位址比較器218接收PLEQ信號，其指示已有一個位址相符合，並且不需寫出色彩矩陣206之內容到RAM，因此指示對現在色彩矩陣內容更新應繼續。畫圖控制器200，則接收銀幕畫面模式SCR、MD控制信號，通知畫圖控制器200需讀和寫多少位元組。

畫圖控制電路200產生一個倒出控制信號DUMP，參考圖7和8B，其使得色彩矩陣206的內容被緩衝在它的第二緩衝段。此外，控制器200產生一個清除位元未定暫存器信號CLRPND，與一個載入位元未定暫存器控制信號LDPND，並耦合此信號到此位元未決暫存器210。此外，控制器200相關於色彩矩陣元素之LDPZX與BPR控制信號，係於圖8B說明。

利用指令解碼器對PLOT命令解碼，與PLOT信號輸入到畫圖控制器200，起始載入未定信號LDPND之產生，其先決條件是像畫圖硬體不忙。LDPND信號耦合到位元未定暫存器210，致能資料的載入從解多工器212進入位元未定暫存器210。產生清除未定信號CLRPND，響應於RAM。RAM完成信號RANDONE，指示未定資料已寫出到RAM。之後，空出此位元未定暫存器，俾用於下一個像素畫圖資訊。

時序圖顯示利用畫圖控制器200，所接收之各種信號間的關係。如圖8C所示之各種不同位址與資料信號，其它相關控制信號，與畫圖控制器產生輸出控制信號。如範例

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (66)

位址位值、資料值等，只為說明的目的。

如下是畫圖硬體 52 之操作。當畫圖控制器 200 決定畫圖硬體 52 不忙。如圖 4A 所示，色彩暫存器 54 的內容載入到 8×8 色彩矩陣電路 206 的水平行。利用列載入色彩矩陣 200，並利用行讀出。利用一個 COLOR 命令更新色彩暫存器 54 的內容。色彩暫存器 54 是一個暫存器，經由此任何後續 PLOT 命令將載入色彩資料到色彩矩陣。

於色彩矩陣 206 的垂直位置，其色彩暫存器位元所載入者，係利用在畫圖 X 暫存器 202 儲存三最低位元所決定。因此，畫圖位址的此三最低位元定義一行的位元，該等位元色彩矩陣 206 被更新。

使用位元未定暫存器 210，以記錄所欲更新銀幕畫面字元段的特定位元。暫存器 210 包括 16 個暫存器旗標，該旗標指示將被寫入銀幕畫面的部份之位元。響應於一個信號 LDPND 載入位元未定暫存器 210，並且利用畫圖控制器 210 產生一個信號 CLRPND 所清除。

如果將執行一個後續畫圖命令，俾在相同地區更新銀幕映至。對一個已知位元重覆操作，同時相關於一個像素之增加色彩資料，被載入到 8×8 色彩矩陣 206。然後，經儲存在畫圖 X 暫存器 202 之最低位元的畫圖位址，另一個位元設定到位元未定暫存器 210。一個特定位元經耦合到畫圖 X 暫存器 202 的一個 3 到 8 解多工器 212，載入到位元未定暫存器 210。如果更新像素多於 8 個像素，或者如果它佔在不同垂直位置，則已寫到矩陣 206 之資料需讀出到

(請先閱讀背面之注意事項再填寫本頁)

· · · · · 裝 · · · · · 訂 · · · · · 線 · · · · ·

五、發明說明 (67)

RAM6(或8)。此後，空間色彩矩陣206，接收新色彩資料，直到接收一個後續畫圖命令，需寫入到RAM9，色彩矩陣206的目前內容緩衝於像素畫圖器硬體內，亦即，在色彩矩陣206內。

當從色彩矩陣206的資料寫到RAM6或8時，做位址轉換計算，使用如圖7所示邏輯閘，全及半加位法器電路，以轉換X、Y座標成為一個RAM位址。本範例的實際位址計算說明如下。此計算將變化，依所使用者是4，16或256色彩模式而定。以256色彩模式為範例之計算。

該256色彩字元具有4方塊的16個位元組，每個定義位元平面對，用於總共64個位元組。

結構一個位元映至，係用在所需銀幕畫面地區的每個位置上，放置一個唯一字元。當在相關於超級NES畫圖時，最好以列安排字元。

例如：(128像素高銀幕畫面)

字元號碼

0	16	32...	...
	1	17	33...
	2	18	24...
	.	.	.
	.	.	.
	15	31	47...

超級NES不受限於256字元，因此位元映至大小主要受限於記憶體與DMA傳送時間。瑪琍歐晶片能畫圖在，例如，

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (68)

128與160像素高銀幕畫面上。最大銀幕畫面是32字元或256像素。

以下演算法範例說明如何使用列結構安排位元映至，以控制像素畫圖。

首先，從X座標的最低三位元，計算一個像素掩蔽(mask)使用於所有位元平面。

像素號碼	掩蔽
0	% 10000000
1	% 01000000
:	:
7	% 00000001

其次，使用移除低3位元之Y座標，計算下列之偏移，給字元於下列，然後利用字元的大小相乘。

銀幕畫面色彩	字元大小以位元組計
4	16
16	32
256	64

再其次，從移除低3位元的X座標，計算字元列的頂部之偏移。列大小是列字元的數目乘以字元大小。

正常列大小

字元高度

	16	20	
4	256位元組	320位元組	
色彩	16	512位元組	640

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (69)

256

1024位元組

1280位元組

Y座標的低3位元給在字元下之位元組偏移。全部偏移加上現在位元映至的指標，獲得像素保有一位元平面之位元組的位址。後續位元平面係可選擇之1位元組，然後15個位元組最後使用像素掩蔽而設定或清除像素位元。在每位元平面之位元設定或清除，使得成為色彩號碼相關位元的狀態時。這號碼依像素之所需儲存在色彩暫存器54。

程式碼的範例

；以65816程式碼在4位元平面上畫圖，使用成遊樂器展示

；此常式大部份都是表驅動

；暫存器A,X與Y是16位元

Setcolour

；取得色彩及二倍

```
lda      colour
```

```
asl      a
```

```
tax
```

；設定色彩掩蔽，使用於位元平面與1

```
lda      mask1tab, x
```

```
sta      mask1
```

；設定色彩掩蔽，使用於平面2與3

```
lda      mask2tab, x
```

```
sta      mask2
```

```
rts
```

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (70)

Plot

; 取得水平和垂直座標

; 二倍二者並且移到 Y 和 X 暫存器

lda plotxl

asl a

tay ; Y 是 X 座標 *2

lda plotyl

asl a

tax ; X 是 Y 座標 *2

; 取得偏移

lda pyoftab, x

; 加上列偏移的開始

clc

ade poxftab, y

; 加上二倍緩衝指標 (選擇位元映至)

clc

adc drawmap

tax

; x 是從位元映至基底保持所需像素字組的偏移

; y 是像素 *2 的 x 座標

; 做位元平面 0 與 1

lda.l bitmapbase, x; 取得字組所保持像素

and pbittabn, y; 掩蔽舊像素色彩

sta pmask

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (71)

```

lda      mask1      ; 掩蔽色彩以及
and      pbittab, y ; 像素掩蔽在一起
ora      pmask       ; 與其它像素相連
sta.1    bitmapbase, x ; 儲存到位元映至

```

; 做位元平面 2 與 3,

```

lda.1    bitmapbase+16, x
and      pbittabn, y
sta      pmask
lda      mask2
and      pbittab, y
ora      pmask
sta.1    bitmapbase+16, x
rts

```

; 像素位元掩蔽對的 256 字組表

pbittab

```

rept      ; num-col
dw $ 8080, $ 4040, $ 2020, $ 1010, $ 0808,
    $ 0404, $ 0202, $ 0101
endr

```

; 上述表具有反向字組

pbittabn

```

rept 32 ; num-col
dw $ 7f77f, -$ 4040, -$ 2020, -$ 1010, -$
    808, $ 404, -$ 202, -$ 101

```

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (72)

endr

; 使用於平面 0 & 1 之列掩蔽 (0 到 15 列)

mask1tab

dw \$ 0000, \$ 00ff, \$ ff00, \$ ffff, \$ 0000,

\$ 00ff, \$ ff00, \$ ffff

dw \$ 0000, \$ 00ff, \$ ffff, \$ 00ff, \$ 00ff,

\$ ff00, \$ ffff

; 使用於平面 2 & 3 之列掩蔽 (0 到 15 列)

mask2tab

dw \$ 0000, \$ 0000, \$ 0000, \$ 0000, \$ 00ff,

\$ 00ff, \$ 00ff, \$ 00ff

dw \$ ff00, \$ ff00, \$ ff00, \$ ffff, \$ ffff,

\$ ffff, \$ ffff, \$ ffff

col-size equ Number-char-rows*&*Number-bit-planes

; (16) (4)

; 到字元列表的開始之偏移

pyoftab

temp=0

rept 32 ; 字元列的數目

dw temp, temp, temp, temp, temp, temp, temp, temp,

temp

temp=temp+col-size

endr

; 偏移下列表

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (73)

p toftab

temp=0

rep 16; 字元行的數目

dw temp

dw temp+2

dw temp+4

dw temp+6

dw temp+8

dw temp+10

dw temp+12

dw temp+14

temp=temp+32

endr

回到圖7更詳細之內容，在銀幕畫面上X與Y座標定義將被畫像的位置，其被載入PLOT X與Y暫存器202及204，（其暫存器也許在暫存器方塊76中R1及R2暫存器）。畫圖位址的最低三位元載入到PLOT X暫存器202，定義在一個位元平面位元組的該位元，被X與Y座標寫入或指定。疊積器R0的內容載入到色彩矩陣206的列，此矩陣利用X暫存器202的最低位元選擇。

如果畫圖X暫存器202是0，則將更新8位元所定義像素之最高位元。若畫圖X暫存器202是0，則3對8解多工器212將設定最低位元，並且在位元未定暫存器210成為邏輯“1”。

(請先閱讀背面之注意事項再填寫本頁)

. 裝 訂 線

五、發明說明 (74)

RAM控制器88使用位元未定暫存器210，以指示不需從RAM寫出之空隙，因為位元未定暫存器210之相關位元指示不需修正。

位元未定暫存器210當成一個像素掩蔽(mask)緩衝器，以防止從RAM覆蓋新資料，假如此新資料不是所期望需要者。欲執行這方式，如圖7所示位元未定暫存器210的市容，則耦合成輸入到色彩矩陣電路206。

如果此BIT-PENDING暫存器210是零，計算出像素的銀幕畫面位址，同時載入到一個畫圖位址暫存器216，並在位元組第像素位置被設定相同於在BIT-PENDING暫存器210之位元。如果BIT-PENDING暫存器210是非零，則設定BUSY旗標。

如果新計算位址等於PLOT-ADDR暫存器216的內容，則在BIT-PENDING暫存器210內設定新像素位元位置，並且重置BUSY旗標。

如果新位址不同於PLOT-ADDR暫存器的內容，則採取如下步驟：

步驟1：假如BIT-PENDING暫存器210包含FFH，則直接到步驟3。

步驟2：從RAM在PLOT-ADDR+scr.基底讀位元組，進入一個暫時資料緩衝器，PLOT-BUFF。

步驟3：假如利用BIT-PEND暫存器210掩蔽資料緩衝位元後，則所有相等於PLOT-COLOR暫存器陳列的第0行，直接到步驟5。

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 (75)

步驟 4：寫 PLOT-COLOR 暫存器陳列的第 0 行，進入利用 BIT-PENDING 暫存器致能 PLOT-BUFF 之所有位元。寫 data-buff 回到 PLOT-ADDR 的 RAM。

步驟 5：做相同操作 (PLOT-ADDR+1)，與 PLOT-COLOR 暫存器陳列的第 0 行。

步驟 6：假如是 8 或 256 色彩模式，做相同操作於 (PLOT-ADDR+16)，與 PLOT-COLOR 暫存器陳列的第 2 行。

繼續直到更新所有色彩位元。

畫圖 X 與畫圖 Y 座標 202, 240 的內容，利用如圖 7 所代表全加法器與半加法器代表之。全和半加法器 FA 和 HA，及相關邏輯電路的構造，已被簡化如圖 7 之方塊圖。如下完成位址計算：

位址：
$$\text{scr-base} + 2 * y[0 \cdots 2] + (y[3 \cdots 7] * 16 + ((x[3 \cdots 7] * 4) \& \& \text{scr-ht}) * \text{char-size})$$

中間項是：

$$\begin{array}{ccccccccc}
 & & & & y7 & y6 & y5 & y4 & y3 & & \\
 & & & & x7 & x6 & x5 & x4 & x3 & 0 & 0 \\
 & & x7 & x6 & x5 & x4 & x3 & 0 & 0 & 0 & 0 \\
 p x 9 & p x 8 & p x 7 & p x 6 & p x 5 & p x 4 & p x 3 & p x 2 & p x 1 & p x 0
 \end{array}$$

因此產生一個 10 位元部份結果 $p x[0 \cdots 9]$ ；使用，例如，6 個全加法器與 4 個半加法器。

該結果給到一個 12×3 方式多工器，利用 char-size 值控制，欲移位部份的結果，則對所選擇銀幕畫面模式進入正確精度。此組合 y 較低位元 $y[0 \cdots 2]$ ，形成一個 16 位元銀

(請先閱讀背面之注意事項再填寫本頁)

· · · · · 裝 · · · · · 訂 · · · · · 線 · · · · ·

五、發明說明(76)

幕畫面位址。欲完成位址計算，則加到 screen-base 值 scr[9...22]，其允許銀幕畫面放置在 1k 的邊際。

然後，該位址耦合到一個 2 位置桶移位器 214，利用 1 或 2 或 4 乘以位址資訊輸入，相關於是否選擇 4，16，或 256 色彩解析度。

移位電路 214 的輸出耦合到一個畫圖位址暫存器 216，此暫存器當成 RAM 位址的緩衝區儲存。因執行畫圖命令後，需緩衝此位址。暫存器 R1 和 R2 的內容，亦即，畫圖 X 和畫圖 Y，也許會改變。

位址比較器 218，比較利用畫圖硬體當成從移位電路 214 輸出所決定之新位址，和儲存在畫圖位址暫存器 216 之舊位址。假如位址不同，則此位址需寫出到 RAM。位址比較器 218 產生一個控制信號 PLEQ [係耦合到畫圖控制器 200]，假如儲存在位址暫存器 216 的畫圖位址相等於移位電路 214 的輸出。

回到色彩矩陣 206，如上所述，色彩矩陣 206 以列方式讀出。一個位元平面計數器 208，耦合到色彩矩陣 206，並且定義該列讀出。位元平面計數器 208 耦合到 RAM 控制器，並且完成一個 RAM 操作。此 RAM 控制器 88 產生一個信號，增加位元平面計數器 208。

色彩矩陣 206，包括如圖 8B 所示之一個陣矩的元素。共 64 個此種元素在 8×8 矩陣 206 的矩陣元素。當解碼此畫圖命令，控制器 200 耦合指令控制信號 LDPIX 到門門 220，以致能從色彩暫存器 54 載入色彩資料 COL 到門門。利用控

(請先閱讀背面之注意事項再填寫本頁)

.....裝.....訂.....線.....

五、發明說明 7(7)

制器 200 產生控制信號 DUMP，指示在色彩矩陣 206 內第一層次緩衝，並且資料需輸出到銀幕畫面。一旦產生 DUMP 信號，儲存在門 200 的資料耦合到開電路 226，以及門 228。當 DUMP 信號活性地耦合到開電路 226，此開電路耦合資料到門 228。同時，除能開 224，依序防止保持先前儲存資料之門 228 的反向輸出，免於迴饋循環。

當從 RAM 所讀之資料，填充到資料空隙，控制信號 BPR 提供一個零輸入到開 222，與此 LDRAM 信號將在一個零狀態。於這些情況下，從 RAMP 輸入之資料輸入將經開電路 226，進入門 228。門 228 資料然後可使用讀出，經此 RAM 控制器 88 到 RAM 資料匯流排，如圖 7 所示。組合其它此種元素，轉換利用 X，Y 像素識別所指示像素資料，成為相容於超級 NES 字元格式的字元資料。

如圖 9 所示 RAM 控制器 88，產生相關於遊樂器卡匣 RAM 之不同控制信號。超級 NES、瑪利歐晶片內畫圖硬體 52，以及執行瑪利歐晶片程式所抓取資料，共同卡匣 RAM。RAM 控制器 88 確保在適當時間送適當位址到 RAM 位址匯流排。在適當仲裁邏輯 310 控制，如圖 10 所示。

RAM 控制器 88，包括多工器 304，俾以多工從 RAM 資料腳經 RAM D 資料匯流排之輸入，與指令匯流排二者之間。指令匯流排或 RAM 資料匯流排的選擇，係響應於從指令解碼器 60 所接信號，並且放置適當 RAM 輸出在目地 Z 匯流排上。

RAM 控制器 88 亦包括一個 16 位元資料暫存器 300，係保

(請先閱讀背面之注意事項再填寫本頁)

· · · · · 裝 · · · · · 訂 · · · · · 線 · · · · ·

存寫到 RAM 資料，其從 16 位元 X 匯流排，或 16 位元 Y 匯流排，均在從指令解碼器 60 所接收信號控制之下。載入到資料暫存器 300 的資料分成一個低位元組，與一個高位元組，並且經多工器 302 耦合到 RAM 資料腳，多工器響應於從指令解碼器 60 接收一個信號，輸出低或高位元組。

RAM控制器88亦包括一個20位元位址多工器308。多工器308響應於從仲裁電路310所接收之一個控制信號，以選擇一個位址輸入，其中仲裁電路出自碼確認PACK，資料確認DACK，或畫圖確認PACK信號，係在仲裁電路310所產生的。從超級NES位址匯流排HA的位址信號，利用多工器308所接收，並且經記憶體時序信號產生器312，耦合到RAM位址匯流排，無論何時瑪利歐“擁有者”狀態位元皆設定為零時。仲裁電路310通知瑪利歐晶片RAM擁有者的狀態，經耦合到仲裁電路310的信號RAN。此仲裁電路亦接收一個RAM更新控制信號RFSH。RAN和RFSH信號係“或”邏輯在一起，以形成如圖10所示之“暫停”信號。

位址多工器 308，亦從 16 位元多工器暫存器 306，接收一個位址輸入。多工器暫存器 306 接收 Y 匯流排的內容，或指令匯流排的內容，依指令解碼器 60 所產生一個選擇信號而定。多工器 308 亦接收資料組暫存器 314 的輸出，當成一個位址輸入與程式計數器 PC 的內容一起，如圖 9 所示。銀幕畫面組暫存器 316 輸出，使用於形成輸入到多工器 308 畫圖位址的最高位元，從圖 7 的畫圖電路輸入成為較低位元。銀幕畫面組暫存器 316 與資料組暫存器 314，從主資料

五、發明說明 (79)

匯流排 HD 載入資料，並且可利用主 CPU 存取。這些暫存器，如圖 9 所示，不需在 RAM 控制器 88 其本身實施，但寧可其內容耦合到 RAM 控制器，資料組暫存器 314 也許，例如，如下所述在 ROM 控制器 104，並且銀幕畫面暫存器也許，例如，在畫圖硬體 52 實施。

多工器 308 輸入信號如下選擇。如果產生碼確認信號 CACK，則選擇碼組與程式計數器 PC。如果產生資料確認信號 DACK，則選擇資料組加多工器暫存器輸入。如果出現畫圖確認信號 PACK，則選擇畫圖位址。最後，如果沒有 CACK，DACK，或 PACK 出現，則選擇主（亦即，SNES）位址輸入。

多工器 308 的 20 位元位址輸出，耦合到記憶體時序信號產生器 312，俾在適當時間耦合這些位址信號到 RAM 6，8。記憶體時序信號產生器 312，從仲裁方塊 310 的格雷計數器接收輸出。記憶體時序信號產生器 312，解碼從格雷計數器之輸出，並且如圖 1 所示經 RAM 位址匯流排 RAMA，產生使用於位址 RAM 6，8 之輸出信號。另一種方式，時序信號產生器 312，將產生用於存取 RAM 6，8 之控制信號，包括行位址選通 RAS，列位址選通 CAS，與寫致能 WE 信號，如圖 1 所示。

記憶體時序信號產生器 312 產生一個 DONE 信號，係回到仲裁邏輯 310，以指示完成 RAM 週期。記憶體時序信號產生器 312，亦產生一個資料門門信號 DATLAT，其門門從外部 RAM 的資料，進入 RAM 控制器 88 的資料門門（未顯示）。從 RAM 資料然後，例如經 RAM 資料匯流排 RAMD-IN，耦合到瑪

（請先閱讀背面之注意事項再填寫本頁）

裝訂線

五、發明說明 (80)

瑪琍歐晶片電路。從時序信號產生器 312 的該 RAM 位址信號輸出，耦合到在遊樂器卡匣上任何靜態 RAM 上。如果遊樂器卡匣上使用動態 RAM，則產生控制信號 CES，RAS 和 WE。依瑪琍歐晶片的結構，適當地產生靜態或動態 RAM 信號，如上述可選擇的電阻器設定所指示。如圖 9A 所示，利用時序信號產生器 312 所產生範例時序信號，與其它相關信號。如所述資料值及範例位址只是基於說明的目地。如圖 8C 所示為 RAM DONE 信號。

部份地利用仲裁邏輯 310，在適當時間控制 RAM 存取信號的產生。如圖 10 所示，仲裁邏輯 310 接收記憶體存取輸入相關於信號 CACHE 要求 CACHERQ，資料要求 DATRQ，及畫圖要求 PLTRQ。每個這些輸入信號暫時儲存在門 325，327，329。如果一個瑪琍歐指令是在 RAM 或 ROM 範圍外執行，起始此程序利用一個快速要求信號 CACHERQ 之接收，此信號在圖 10 的前後中使用，以確認正在執行指令不離開快速 RAM 範圍，並且需被執行在 RAM 或 ROM 範圍外。因此，快速要求 CACHERQ 信號，指示不能執行指令在快速記憶體 94 範圍外。產生資料要求信號 DATARQ，係當成一個解碼一個指令所需 RAM 存取的結果（例如，載入位元組，載入字組指令）。此外，仲裁邏輯 310 接收一個畫圖要求信號 PLTRQ，利用畫圖控制器 200 響應於一個畫圖命令的解碼所產生者。

仲裁邏輯 310 只能致能（如所示，利用一個狀態暫存器暫停模式位元是在“0”狀態），係當瑪琍歐晶片正在執行

（請先閱讀背面之注意事項再填寫本頁）

裝訂線

五、發明說明 (81)

時，與當瑪琍歐擁有者位元設定時。在快速要求，資料要求，及畫圖要求信號，門門325，327與329後，產生CRQ、DRQ與PRQ信號。開331、333與335，從個別的門門非反向輸出，接收這些信號。並且對這些信號建立優先順序。在這方面，快速要求信號具有最高優先，資料要求第二最高優先，畫圖要求信號具有最低優先。快速要求信號指定成最高優先，因為它指示企圖執行快速記憶體範圍外指令，並且它需從RAM存取此指令。操作開電路333與335，確保一個較低優先要求不設定門門339與341，如果一個較高優先要求已設定它的個別的門門。只有當系統不在暫停模式，才能設定門門337，339，341，因為暫停模式信號輸入到每個開331，333，335。暫停模式信號將設定在一個低邏輯準位狀態，當瑪琍歐擁有者，亦即自由存取於RAM時。當任何確認門門337，339及341已經在“1”(亦即，一個週期正在進行中)，假如暫停設定為“1”，則不能設定門門337，339及341。開331，333和335建立RAM存取的優先。如果快速要求門門337設定，則將不設定資料確認門門339。如果快速或資料要求門門設定，則也不設定畫圖確認門門341。

門門337利用快速要求信號設定，立即產生快速確認信號CACK，並且立即利用如圖10所示邏輯電路建立可用的快速94(或RAM)。產生資料確認信號DACK與畫圖要求確認信號PACK，以確認資料要求與畫圖要求信號，如果是圖所示邏輯電路決定RAM是不忙。

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (82)

門 337, 339 和 341 的非反相輸出, 耦合到開電路 343, 係依序經 NOR 閘 344 重置格雷計數器 345, 產生用於 RAM 存取之時序信號。該行業專業人士均知道, 格雷計數器是在一個時間只有一個輸出位元改變之計數器, 可方便地使用於控制 RAM 存取時間。

利用時序信號產生器 312 產生的完成信號, 利用 NOR 閘 344, 與門 337, 339, 341 而接收。完成信號指示完成一個 RAM 週期。完成信號的產生, 觸發仲裁邏輯 310 清除適當門, 以清除已門之要求。完成信號亦耦合到原始電路, 亦即, 快速控制器 68 或畫圖控制器 52, 俾指示已完成 RAM 存取。

根據本發明另一個可選擇的實施例。瑪利歐晶片使用一個雙時鐘信號系統。因此, 瑪利歐晶片處理器, 與上述 RAM 控制器電路不需利用相同時鐘信號驅動。RAM 控制器 88 也許, 例如, 利用從超級 NES 所接收 21MHz 時鐘信號驅動, 並且利用其它可變頻率時鐘信號驅動瑪利歐晶片處理器。於此方式, 瑪利歐晶片處理器將不受限制於 21MHz 時鐘信號速率之操作。

根據本實施例之瑪利歐晶片, 也許使用一個非同步狀態機械控制電路, 如圖 11 所示, 使用於執行一個再同步雙時鐘信號界面功能。圖 11 電路用於和瑪利歐晶片處理器界面, 如果是用一個不同於記憶體控制器操作之時鐘信號系統實施時。

如圖 11 所示之再同步電路, 接收一個輸入時鐘信號

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明(3)

DIN, 此信號不同步於一個時鐘信號CK。再同步電路從DIN產生一個信號, 此IDN同步於CK, 不管是否DIN頻率高於或低於時鐘信號速率CK。

如圖12所示實施例, 響應於信號DIN, 如圖11所電路。從狀態010, 110, 100, 101, 111轉換並回到起始狀態010。圖11再同步電路用於任何界面接收雙時鐘信號, 例如ROM控制器104和RAM控制器88。

如圖11所示電路響應於輸入信號DIN, 因此交換從空間或重置狀態“010”以形成狀態“110”, 係由於利用開F設定門門A。只是再同步時鐘信號CK走慢, 也許已經是真, 則利用開E重置門門B形成狀態“100”。當時鐘信號再走高, 則設定門門C利用開A形成狀態“101”。

門門C從電路產生輸出, 如圖11在Q所示者。當輸入信號再走低, 門門B再次利用開C設定形成狀態“111”。當到達狀態“111”後, 時鐘信號CK再走低, 然後利用開G重置門門A, 以形成狀態011。此後, 時鐘信號CK再走高, 並且利用開重置門門C, 返回到狀態機械的空間狀態。

圖13所示係圖4B ROM控制器104之進一步詳細圖。ROM控制器104包括一個快速載入器400, 其控制載入儲存在ROM10與卡匣RAM內, 目前正執行程式指令到瑪琍歐晶片快速RAM94中。載入到快速RAM94指令以16-位元組為一群。當遇上一個跳躍指令, 於16位元組段的中間, 一個完全16位元組段需繼續填入, 在跳躍執行之前。快速載入電路

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (84)

400，包括一個2位元狀態機，係響應於跳躍指令的解碼，因此確保16位元組快速段的其餘位元組載入快速RAM94中。快速載入邏輯狀態機的第一狀態是空閒狀態，其將是真，如果程式執行在快速記憶體範圍外，或如果程式資料已載入到快速記憶體中時。第二狀態指示，於同時發生從卡匣ROM或RAM中，載入快速記憶體和執行程式。利用跳躍指令的解碼觸發第三狀態，其狀態仍然有效，直到16位元組快速段的所有位元組已載入為止。第四狀態遭遇，當執行一個跳躍時，此跳躍到不精確此相關於一個快速16位元組邊界的位址上。於此狀況中，此快速記憶體填入，從邊界的開始，到程式跳躍的部份16位元段相關位址。

如圖4B所示快速控制器68，其產生一個快速信號，輸入到快速載入器400。並且指示所要求指令不出現在快速RAM94中。因此，指令需從ROM中抓取。碼組信號識別將被存取位址的最高三位元。並且指示是否將被存取程式ROM或RAM。快速載入器400亦包括一個計數器(未顯示)，其在程式執行期間，維持相關於程式計數器PC的最低位元之計數。此計數器載入經快速載入器400的PC輸入。

在ROC控制器104的快速載路電路400，亦接收等待與執行控制信號。其指示瑪利歐處理器不論任何理由保持在等待狀態，與瑪利歐晶片是在“執行中”模式。於此環境下，快速載入電路400產生一個程式碼抓取控制信號，如圖13所示耦合到NOR閘408，依耦合到ROM時序計數器406的清除輸入。當快速載入電路400產生一個程式碼抓取信號，

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 8(5)

在 ROM 控制器 104 內的邏輯電路，起始一個高於資料抓取的優先之程式碼抓取，此時和程式碼抓取起始先於資料抓取。仲裁電路配合優先邏輯，如圖 10 所示，也許使用於致能此產生信號，此信號所給的優先高於資料抓取者。

當從 ROM 時序計數器 406 移除清除信號時起始一個計數週期。ROM 時序計數器 406 用於產生 ROMRDY 時序信號，係指示 ROM 資料在 ROM 資料腳可用者，其信號是從開電路 410 輸出。

此 ROM 資料準備信號 ROMRDY 開，耦合到再同步電路 402，其也許，例如，包含如圖 11 所示再同步電路。於處理器時鐘信號獲得同步後，產生信號 ROM DCK 重置門 404，並且產生一個資料抓取信號，指示一個資料抓取，其利用暫存器 R14 的存取所觸發者，因此獲得 EN-R14 信號。當 ROM 時序計數器 406 到達一個預先決定之計數，產生資料抓取信號，以確保資料在 ROM 資料腳是可用的。

圖 13 所示 ROM 控制器，產生一個 ROM 位址從多工器 414 輸出，並從下列輸入之一選擇位址資訊。程式碼暫存器 412 從超級 NES 資料匯流排載入，以定義將被執行瑪利歐程式碼之 ROM 程式組。程式組暫存器 42，提供一個 23 位元 ROM 位址的 8 位元到多工器 414。ROM 位址的較低位元，係從程式計數器 PC 的內容獲得。當資料寫入到快速 RAM 中，利用快速載入 400 產生從快速載入信號之較低 4 位元。無論何時存取暫存器 R14，從瑪利歐一般暫存器 R14 的內容，產生增加多工器 414 位址輸入。

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

五、發明說明 (86)

暫存器 R14 的存取所得的資料抓取門 404，產生一個資料抓取信號，使用於一個控制輸入使得多工器 414 選擇它的 R14 輸入。(並且從超級資料匯流排 HD，載入資料組暫存器 416 的內容)。資料組暫存器 416，包含資料組的較高位元，係相關於一個 R14 抓取操作。

此外，資料抓取信號耦合到門 408，將起始 ROM 時序計數器 406 的計數，依序經閘 410 產生一個 ROM 準備信號 ROMRDY。當產生 ROMRDY 信號時，來自 ROM 資料匯流排 ROM D[7... 0]之資料是可用的。

位址多工器 414，亦從超級 NES 位址匯流排 HA，接收一個 ROM 位址。將選擇超級 NES 位址匯流排，依信號“ROM”的狀態而定，並耦合到多工器 414 控制輸入。“ROM”控制信號，指示瑪利歐 ROM 控制器，超級 NES 具有 ROM 位址匯流排的控制。

在解碼一個跳躍指令後，位址多工器 414 進給程式計數器，加上快速載入器 400 內計數器所產生的 4 個數低位元。而允許快速段載入其餘 16 位元組，此載入在解碼跳躍之前。

多工器 422，提供 ROM 控制器 104 內資料路徑，從 ROM 資料 ROMD 到瑪利歐的目的匯流排 Z。資料抓取信號利用門 404 所產生，與利用 ROM 時計數 406 產生之 ROMRDY 信號，耦合到閘 418 以致能 ROM 緩衝區 420 的載入。從 ROM 資料匯流排 ROMD[7... 0]之 ROM 資料，載入到 ROM 緩衝區 420。

多工器 422 響應於一個指令碼的解碼，以選擇一個輸

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 ()
87

入。(例如，GET B利用暫存器R14的存取，觸發自動資料抓取)。如果解碼程式碼抓取操作，ROM控制器104將耦合指令，到如圖15A所示瑪琍歐晶片的指令匯流排。如果解碼一個GET B指令，則儲存在暫存器402的位元組放置在Z匯流排上。某些GET B指令操作包含在X匯流排上資料，係經相關輸入到多工器422，如圖13所示。然後耦合到目的地Z匯流排的資料，被載入到瑪琍歐一般暫存器76之一。

如圖14所示之快速控制器進一步詳細說明。快速控制器68包括一個標識門門506。標識506，包括，例如，64個門門以指示是否指令儲存在快速RAM94(當實施在快速控制器時，只為說明之目的而已)。

在標識門門506之每個64旗標，相對於儲存在快速RAM94儲存之16位元資訊。指令載入快速RAM94，同時指令從ROM或RAM執行。當執行一個跳躍指令，如上所述，RAM94經快速載入器400，載入16位元組段的其餘位元組，相關於圖13所示ROM控制器104。直到載入這些其餘位元組為止，當載入經旗標識門門506，其不能旗標整個16位元組段。

敘及開電路510，當程式計數器從0到15的計數，14位元減法器502輸出一個範圍外的信號(其係反相)，並且當ROM控制器輸出其ROM資料準備信號ROMRDY(指示一個位元組準備好輸出)時，開電路510設定標識門門506，其所在位置的定址則利用解多工器504。

(請先閱讀背面之注意事項再填寫本頁)

經濟部中央標準局員工消費合作社印製

五、發明說明⁸⁸)

當解碼一個快速指令，在匯流排 501 產生一個控制信號，以指示從快速 RAM 記憶體 94 有後續指令執行。在匯流排 501 上之控制信號，耦合到快速基底暫存器 500 載入輸入，並且載入程式計數器 PC 的 13 較高位元於快速基底暫存器 500。同時，如圖 14 所示，清除識門門 506。

快速基底暫存器 500 的輸出與程式計數器的較高位元（例如位元 3-15），耦合到減法器 502，決定是否從程式計數器 PC 的位址輸入，於快速 RAM 94 範圍內。減法器 502 之輸出，例如，其 6 個較低位元，如快速 RAM 位址的較高位元，此三較低位址位元從程式計數器 PC 耦合。

自減法器 502 由一個進位輸出信號，產生一個範圍外信號 O/RANGE，並且反相。當反相範圍外信號高時，以起始門陣列 506 之一個門的設定。門設定將依經解多工器 504 從減法器 502 之快速位址輸出而定，並且相關於在快速 RAM 94 的一個 16 位元組段，以指示一個指示儲存於快速記憶體，其係相關於輸出快速 RAM 位址。標識門 506 輸出耦合到一個多工器 512，以耦合 64 標識門信號之一，到 NOR 閘 514，基於多工器選擇輸入以將輸出的一個門信號，相對於從 DEMUX 504 之 64 個選擇輸出之一。其它需要一個外部抓取，因為快速 RAM 94 無法發現到所期望指令。

圖 15A 係圖 4A 中 ALU 控制器 / 指令解碼器 606 的方塊圖。如圖 15 所示，ALU 控制器 / 指令解碼器 60，從快速 RAM 94，ROM 控制器 104 與 RAM 控制器 88 接收指令。此瑪利歐晶片組件不是 ALU / 指令解碼器 60 的一部份，但圖 15 所示亦只是說

（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()

明之目的。

多工器 525 選擇一個指令輸出，經由快速 RAM 94，ROM 控制器 104，或 RAM 控制器 88。並且輸入此選擇指令到平行管道門 527。利用多工器 525 在 RAM 或 ROM 基指令間選擇，依程式碼組暫存器，亦即位元 4，的預定位元狀態而定。因此，依所載入到程式碼組暫存器之位址資訊，將解碼 ROM 或 RAM 之指令。另一可選擇之多工器 525 從快速 RAM 94 選擇一個指令，依快速控制器 68 之一個控制信號 CACHE CTC 的狀態而定，指示在快速 RAM 94 範圍內之一個指令將被執行，並且設定一個適當標識位元，相關連於快速控制器 68。

平行管道門 527，自多工器 525 接收一個 8 位元指令，當利用一個程式計數器致能信號 PCEN.IL.IH 時，如果從 ROM (或 RAM) 正在抓取一個指令，則利用 ROM 控制器 104 (或 RAM 控制器 88) 產生此致能信號。因為從 RAM 或 ROM 抓取一個指令，所費時間超過一個處理週期，因此，利用 ROM 或 RAM 控制 104，88 所產生之程式計數器致能信號 PCEN，以解觸發指令解碼操作。

另一方面，假如所執行指令在快速 RAM 94 範圍外，則所有時間程式計數器致能信號 PCEN 是活性的，並且在全處理器時鐘信號速率執行指令。因為 ROM 存取時間慢於快速 RAM 或卡匣 RAM 存取時間。必需所產生 PCEN 信號對 ROM 存取的頻率間隔，低於相關快速 RAM 或動態或靜態 RAM 解碼致能信號。

於平行管道門 527 所暫存儲時儲存之指令，輸出到

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明⁹⁰)

傳統指令解碼電路，利用開電路 537，539 和 541 開電路所代表，以產生操作程式碼 1, 2... N 的信號指示。

載入到平行管道門 527 指令，亦耦合於向前看邏輯 551。向前看 (Look-ahead) 邏輯 551 提供操作程式碼的一個預先解碼指示，係當成瑪利歐晶片暫存器方塊 76，選擇適當暫存器。因此，欲在解碼此運算碼之前最佳化執行速度，快速決定所需存取的暫存器，以致能指令所需資料的高速存取。

向前看邏輯 551，響應於指令運算碼位元，與各種不同程式解碼控制旗標。指令解碼電路 60 包括程式控制旗標檢測器邏輯 543，其響應於先前所被解碼操作碼，以產生 ACT1 和 ACT2 信號，指示如上述相關前置指令已被解碼。如下所述相關 ACT1 PRE 信號，亦利用旗標檢測器邏輯 543 產生。此外，產生 IL 與 IH 信號，以指示已解碼指令所需立即資料。(其中 L 和 H 指示低位元組和高位元組)。IH 與 IL 旗標預先排除此立即資料相關指令，被解碼成操作碼。因此，亦不需 IL (IL) 和 IH (IH) 致能平行管道門 527。如前所述，ACT1 與 ACT2 修改一個後續產生運算碼，並且輸入到解碼邏輯 537，539，541 等。例如，在開電路 541 根據先前討論這些信號，修改輸出運算碼。

向前看邏輯 551 產生暫存器選擇，此信號基於預先解碼的運算碼，與當先前操作解碼 (例如，前置碼 ACT1 或 ACT2) 所產生信號。例如，在程式控制旗標檢測邏輯 543 內所示，如果利用解碼邏輯 545 解碼 ACT1 信號，產生一個

(請先閱讀背面之注意事項再填寫本頁)

五、發明說明 (91)

ACT1 PRE信號，其中利用程式控制旗標檢測器邏輯 543 輸出信號，並且信號依序經 OR 閘 549 耦合到向前看邏輯 531。ACT1 PRE信號亦設定 ACT1 門 547。OR 閘 549 亦從門 547 輸出 ACT1 信號，並且耦合 ACT1 信號，並且耦合 ACT1 信號到解碼邏輯 537，539，541 等。

圖 15 示意地代表向前看邏輯，說明如何產生 4 個暫存器選擇控制位元 XSEL0、XSEL1、XSEL2、及 XSEL3。然後這 4 個控制位元，耦合到所述多工器 620 與 622，相關連於圖 17 暫存器控制邏輯 76，選擇 16 個暫存器之一的內容，並輸出到 X 匯流排被一個執行中指令所使用。

因此，一個指令在載入到平行管道 527 之前，耦合到向前看解碼邏輯元素 529，產生一個暫存器選擇位元 XSEL-0，依序地，門 535 中，然後當成信號 XSEL0 輸出。利用程式計數器信號 PCEN 致能門 535。類似地，邏輯電路 531 產生 XSEL-U1，被門 533 中，當成信號 XSEL1 輸出。ALT1 PRE 信號耦合到各種不同解碼邏輯電路 529，531 等在此向前看邏輯 551。並且用於定義利用暫存器控制邏輯 76 所選擇之適當暫存器。例如，如向前看電路 551 所示，ACT1 PRE 信號耦合到邏輯電路 531 信號之一，產生 XSEL-U1，被門 553，依序輸出信號 XSEL1。

圖 15B 所示範例時序信號，係用於證明向前看邏輯 551 的操作。圖 15B 中，一個時鐘信號 CK，和一個範例指令運算碼相關於快速 RAM 資料存取。所示時序信號亦指示當載入平行管道門 527 時，當執行指令解碼操作時。當產

五、發明說明 (92)

生暫存器選擇信號時，以及當從暫存器資訊載入在於目的地 Z 匯流排上。

如圖 15B 所示，快速 RAM 資料運算碼運算碼 1) 成為有效，係在某些點上於時鐘信號脈衝 CK 的上昇邊緣後時間。儲存運算碼在平行管道門 527，直到，例如，第二時鐘信號脈衝的下降邊緣，此時運算碼 2 載入到門 527。指令解碼器 60 開始解碼相關於運算碼 1 之指令，於剛接收從門 227 的輸出，其時間如圖 18 所示意。指令解碼的結果將如上述，適當地耦合控制信號到瑪喇歐組件，例如 ALU 50，快速控制器 68，與畫圖硬體 52 等。

如圖 15 所示向前看電路 551，在運算碼 52 的解碼之前時間點，利用產生一個信號 XSEL-U 開始此暫存器選擇解碼程序。在被門到門 535 之前，XSEL-U0 信號代表解碼邏輯 529 的輸出。利用門 535 輸出 XSEL-0 信號，在此時間點，對於令所需資料於指令執行週期內，儘可能及早存取，更快的耦合到適當匯流排。

如圖 16 所示一部份的暫存器控制邏輯 78，係用於產生 Y 和 Z 匯流排相關暫存器選擇信號。多工器 604 選擇從 Z 匯流排寫到 16 個暫存器的那一個。多工器 606 選擇那一個暫存器進給 Y 匯流排。

多工器 604 與 606 從 4 位元暫存器 600 與 602 接收輸入。使用暫存器 600 與 602 以實施上述 "TO" 及 "FROM" 前置指令。利用 "TO" 及 "FROM" 前置的解碼致能暫存器 600 與 602，以操作耦合指令匯流排的較低位元，到暫存器 600 和

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (03)

602。清除暫存器 600 與 602，以響應於一個指令，如上述之重置控制旗標。

此外，多工器 604 與 606 從在暫存器方塊 76 之不同暫存器接收輸入。多工器 604，606 從指令匯流排上較低位元接收輸入，以實施指令的較低 4 位元，定義指令目的地或來源暫存器。此外，從超級 NES 位址匯流排之預先決定較低位元，耦合到多工器 604 與 606，俾提供超級 NES 存取到此暫存器組。多工器 604 與 606 選擇暫存器進給 Z 和 Y 匯流排。

圖 17 顯示暫存器方塊 76，和實施在圖 4B 內暫存器控制邏輯 78 的增加暫存器選擇控制邏輯。設定 FROMX 暫存器 618，利用解碼一個 FROM 指令所產生的一個 FROMSET 信號。在接收 FROMSET 信號後，Y 匯流排的內容載入到暫存器 618。載入在暫存器 618 之資料，則成為用於後續指令執行的資料。暫存器 618 的內容，耦合成輸入之一到多工器 622。多工器 622 亦接收 R0 暫存器的內容，（其係用一個預設暫存器），當成它的輸入之一。

多工器 622 另一個輸入是多工器 620 的輸入。多工器 620 接收程式計數器的內容輸入（亦即暫存器 R15），從使用在執行 MERGE 指令暫存器輸入，與暫存器 R1（例如，用於執行此畫圖指令）。多工器 620 選擇此輸入之一，基於利用如圖 15A 所示向前看邏輯 551，而產生 XSEL2 與 XSEL3 位元的狀態。

多工器 622 一個增加輸入，耦合到 Y 匯流排的內容，以放置相同資料於 X 匯流排上，如在 Y 匯流排上。如前所

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明 (4)

述，到多工器 622 另一個輸入，係如上述 FROM X 暫存器 618 的輸出。選擇多工器 622 的輸出，係基於在圖 15A 所產生 XSEL0 與 XSEL1 位元的狀態，並且耦合到 X 匯流排。

在上述說明相關許多暫存器 R0~R15 之特別目的地功能，於此將不再重覆。暫存器 R0~R3 的輸出耦合到多工器 608，暫存器 R4~R7 的輸出耦合到多工器 610，暫存器 R8，R11 的輸出耦合到多工器 612，暫存器 R12~R15 的輸出耦合到多工器 614。到多工器 608，610，612 和 614 之 4 個個別輸入之一，利用 YSEL1 與 YSEL0 位元選擇，如圖 16 所示從多工器 606 輸出。從多工器 608，610，612 和 614 輸出，依序輸入到多工器 616。到多工器 616 之 4 個輸入之一的選擇，係基於如圖 16 從多工器 606 的 YSEL2 與 YSEL3 位元輸出的狀態。多工器 616 的輸出耦合到緩衝區暫存器 617，其輸出依序耦合到 Y 匯流排。

參考暫存器 R0 到 R15 輸入，每個暫存器具有致能輸入，利用如圖 16 所示所產生的 ZSEL 位元 0 到 3 而選擇。每個暫存器亦有一個時鐘信號輸入 CK，與一個資料輸入 Data-in，於適當緩衝後，經此從 Z 匯流排接收資料。

暫存器 124，係用於相關於各種不同乘法操作，亦包括除能低與除能高位元輸入，以及致能低與致能高位元輸入。暫存器 R15，程式計數器 PC，從在圖 13 的 ROM 控制器之快速載入器 400，接收一個信號 CCHCD，禁止一個跳躍操作，直到目前 16 位元組快速段載入到快速 RAM 為止。此外，程式計數器從指令解碼器，接收一個程式循環未定信號

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

經濟部中央標準局員工消費合作社印製

五、發明說明⁹⁵)

LOOPEN，其指令應產生一個跳躍操作，並且致能PC載入暫存器R13的內容。此外，暫存器R15接收一個開機重置信號RESET，並且當執行一個循環指令時，一個PN輸入使得程式計數器載入暫存器R13的內容。

如上所述，本發明之繪圖共處理器組合於主電視遊樂器系統，利於使用產生不同的特殊效應，其中包括，例如，多邊形基目標的旋轉，放大，與/或縮小。圖18的範例瑪利歐晶片程式的一個流程圖，係用於畫出一個梯形，描述如何程式化瑪利歐晶片，以產生被顯示的一部份多邊形基目標。說明產生此多邊形目標的瑪利歐程式，將一起詳細解釋瑪利硬體如何執行此程式如下。

首先參照圖18所示流程圖的高層次，起初，某些在暫存器方塊R1到R15的暫存器，係相關於產生梯形之變數。（亦即，暫存器R1儲存像素X位置，暫存器R2儲存像素Y位置線，暫存器R7儲存梯形高度等）。此後，如方塊650所示，設定一個循環計數器，並且計算起始像素值。

如方塊652所示，做一個檢查，以決定梯形水平線之一的高度。假如線的開始點減去線的末端點的結果，是一個負值（-VE），則此常式跳躍到方塊660。假如線開始點減去線的末端點，是一個正值，係指示線的長度不超過，則減低循環計數器（654），並且執行一個畫圖像素指令，以獲得適當像素的畫圖。

如方塊658所示，做一個檢查，以決定是否循環計數器的內容為零。如果循環計數器不是零，則執行一個跳躍

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

214588

五、發明說明 96)

回到方塊 654，以減低循環計數器 (654)，並且畫出另一個像素 (656)。

如果循環計數器等於零，則更新左多邊形邊 X 座標，與右多邊形邊 X 座標 (660)。以後，減少梯形的 Y 高度 (662)。並且若結果不是零，則將跳回到方塊 650 (664) 再執行常式，並且增加 Y 座標，而移動到下一個掃描線 (665)。如果高度等於零，則將完全執行此常式，並且完成此梯形 (666)。

欲說明瑪利歐晶片指令的使用產生縮圖，以用於畫出一個梯形的範例程式，係圖 18 流程圖之實施例以次說明如下。

； 畫 梯 形 循 環

```

rx          = 1          ; 畫 x 位置
rx          = 2          ; 畫 y 位置
rx1         = 3          ; 最上左 x 位置
rx1inc      = 4          ; 最上左 x 位置增加
rx2         = 5          ; 最上右 x 位置
rx2inc      = 6          ; 最上右 x 位置增加
rdy         = 7          ; 梯形 y 高度
rlen        = 12         ; 循環計數, hline 高度
rloop       = 13         ; 循環標識
hlines

          miwt rloop, hline2 ; 設定 hline 循環的開始
hline1

```

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明(7)

```

mfrom rx1          ; x=(rx1)>>8
mto   rx
mhib
mfrom xr2
mhib
mto   rlen
msub  rx            ; 長度,rlen=(rx2>>8)-(rx1>>8)
mbmi  hlines3       ; 結果rlen<0則跳hline
mnop
minc  rlen          ; 經常畫一個像素
hlines2
mloop
mplot          ; 畫hline
hlines3
mwith rx1       ; rx1+=rxline
madd rxline
mwith rx2
madd rx2inc     ; rx2+=rx2inc
mdex rdy        ; rdy-=1
mbne hlines1    ; 重覆rdy次
mine ry         ; 並且下一個y完成

```

欲證明瑪琍歐硬體操作如何執行一個程式，如下說明
上述梯形產生之程式。在執行梯形產生程式之前，主電腦

(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(8)

系統，亦即，超級NES，直接寫到程式碼組暫存器，並且進入銀幕畫面基底暫存器，如上所說明相關於圖5流程圖之說明。此外，超級NES寫低位元組的XEQ位址，到ROM控制器104的一個當地暫存器，從超級NES位址匯流排HA解碼。然後超級NES寫高位組到ROM控制器104，結合當地暫存器的內容，並耦合到Z匯流排。此後，致能當成瑪利歐晶片程式計數器的暫存器R15。

在檢測上述超級NES寫操作到ROM控制器104的後邊緣，設定瑪利歐“GO”旗標。假如程式計數器減掉快速基底暫存器，大於快速記憶體大小，或假如快速旗標倍於程式計數器，減掉快速基底暫存器除以16，等於零，則程式計數器內容傳送到ROM10，並且開始ROM時序計數器（圖13方塊406）。

起始在執行畫梯形副程式之前，用於梯形循環程式的變數，其相關於超級瑪利歐暫存器，如所示在梯形程式列表的起始部份，例如，“rx”是“畫圖X位置”相關於暫存器R1，以及變數“rloop”相關於暫存器R13。

在指定這些暫存器後，如下執行梯形程式。當在ROM控制器104之ROM時序計數器406，到達5的計數（大約200微微秒/時，將執行第一個指令“IWT rloop, hline, 2”，從ROM資料匯流排門閥進入圖4A所示平行管道暫存器62。同時，資料寫入快速RAM94。在執行指令“IWT rloop, hlines”時，增加程式計數器。“IC”和“IM”旗標設定，以標明在指令流的後續二個位元組是立即資料。當ROM

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明(90)

時序計數器 406 到達 5，立即資料（低位元組）寫到快速 RAM 94 中，並且保持在 ROM 控制器 104 之暫存器。重覆 ROM 抓取機構，並且立即資料的高位元組組合於低位元組，並沿路進到 Z 匯流排。致能暫存器 R13，在以儲存 Z 匯流排的內容，欲設定循環計數器。從該常式的此點，自記憶體抓取每個指令，直到遇到循環指令。

在執行 FROM RX1 指令時，指令碼的最低 4 位元，載入到暫存器控制器（參考圖 16）的 4 位元“FROM Y”暫存器。此外，致能從 RX1（暫存器 R3）資料進入到 Y 匯流排，並且儲存在此 16 位元“FROM X”暫存器 618。在執行“TO RX”指令時，指令碼的最低 4 位元，載入到暫存器控制器（參考圖 16）的 4 位元“致能 Z”暫存器 600。

執行“HIB”指示，利用設置“FROM X”暫存器的 16 位元內容，進入 X 匯流排。ALU 放置 X 匯流排的最低位元組，進入 Z 匯流排的低位組，並且設定 Z 匯流排的最低位元組於零。此移動部份的 X 位置，並且留下開始點，用於在暫存器（暫存器 R1）之第一水平線。

在執行“FROM RX2”指令時，係類似於上述執行“FROM RX1”指令之操作。“HIB”指令所引起之操作，（類似於上述者），其相對於梯形的最上右 X 座標，留下第一水平線末端在暫存器 R0（此預設暫存器之操作如疊積器）。

執行“RLEN”指令與“SUB RX”指令時，因此線的開始減掉的末端 $RLEN(R12) = R_0 - R_x$ 。假如有一個負結果，則設定正負號旗標，指示出有一個錯誤條件。

（請先閱讀背面之注意事項再填寫本頁）

裝
訂
線

五、發明說明(00)

"BMI HLLNES3"指令是一個2位元組指令，假如正負號旗標設定，則其中第一位元組設定一個旗標。假如條件旗標設定，則第二位元組為跳躍偏移。(其中R15等於R15加上此指令)。假如不是，R15仍然未改變，並且繼續正常程式執行。

執行"INC RLEN"指令，線長度暫存器加1，以確保至少畫一個像素。"LOOP"指令操作使得 $R12 = R12 - 1$ 計算。假如R12不是零，則R15(程式計數器)載入R13的內容，因此有一個跳躍。

假如在此點程式於快速RAM94範圍內，則快速載入電路400將檢測到此跳躍，並且將如它做的暫停執行繼續載入快速RAM94。當它完成，程式計數器載入新值，並且從快速RAM94抓取下一個指令。

欲執行"PLOT"指令，該循環/畫圖指令將形成一個水平線畫圖演算法。"PLOT"指令將利用R1, R2(當成X和Y座標)設定銀幕畫面像素位址，到如圖4A所示"色彩暫存器"54之色彩設定。利用畫圖硬體52計算包含像素之字元位址。新像素資料保持在字元線緩衝區(色彩矩陣)，直到瑪琍歐晶片移入畫在不同字元位置。當所有色彩資訊拷貝到色彩矩陣內雙緩衝區機構的第二層時，則資訊寫到外部RAM。

執行"WITH RX1"與"ADD RX1 LNC"指令，係更新梯形的左邊X座標。類似地，"WITH RX2"與"ADD RX2 INC"係操作更新梯形的右邊。"DEC RDY", "BEN,

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

經濟部中央標準局員工消費合作社印製

五、發明說明 1(01)

Hlines1", 與 " INC RY " 指令操作移入下一個 Y 位置 (下一個掃描線), 直到完成梯形。

下列程式列表, 示範如何程式瑪瑙歐晶片旋轉陣列的 8 位元 X, Y 和 Z 點。此常式係根據本發明實施例程式繪圖共處理器, 如下是本常式之列表。

從轉列表:

; 旋轉一個 8 位元 X, Y, Z 點的陣列

; 利用暫存器之一個旋轉矩陣

; rmat1211, rmat2113, rmat2322, rmat3231,
rmat0033

; 矩陣元素係 8 位元有正負號分數

; 例如 $127 = 127/128 = \text{大約 } 1$

; $-128 = -128/128 = -1$

; 彼等緊密地儲存, 當成每個暫存器之 28 位元元素

rx = 1 ; x

ry = 2 ; y

rz = 3 ; z

rt = 4 ; temp

rmat1211 = 5 ; 矩陣元素 11 和 12

rmat2113 = 6 ; 矩陣元素 13 和 21

rmat2322 = 7 ; 矩陣元素 22 和 23

rmat3231 = 8 ; 矩陣元素 31 和 32

rmat0033 = 9 ; 矩陣元素 33

routyptr = 10 ; ptr 到旋轉點緩衝區

五、發明說明102)

msh-rotpoints8

```

miwt  r14,pointsaddr    ; ROM ptr到將被轉旋點
miwt  r12,numpoints      ; 將被旋轉點的數目
miwt  routptr,m-rotpnts; RAM ptr到旋轉點緩衝區
mcache                                ; 設定快速記憶體位址
mmove r13,pc              ; 起始循環位址

```

mmatrotplloop

```

mto    rx                ; 取得 x
mgetb
minc   r14
mfrom  rmat1211          ; 11
mto    rt
mmult  rx                ; m11*x

mto    ry                ; 取得 y
mgetb
minc   r14
mtrom  rmat2113          ; 21
mhib
mmult  ry                ; m21*y
mto    rt
madd   rt

```

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 1(03)

mto rz ; 取得 z

mgetb

minc r14

mtrom rmat3231 ; 31

mmult rz ; m31*z

madd rt

madd ro

mhib

mstb (routptr) ; 儲存旋轉 x

mine routptr

mtrom rmat1211 ; 12

mhib

mto rt

mmult rx ; m12*x

mfrom rmat2322 ; 22

mmult ry ; m22*y

mto rt

madd rt

mtrom rmat3231 ; 32

mhib

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

(請先閱讀背而之注意事項再填寫本頁)

m a d d r t

m h i b

```

m i n c      r o u t p t r

```

nto rt

```
mtrom rmat2322 ; 23
```

m h i b

m t o r t

m a d d r t

```
mmult rz ; m33*z
```

m a d d r t

m a d d r 0

m h i b

裝
•
•
•
•
訂
•
•
•
•
線

五、發明說明105)

mstb (routptr) ; 儲存旋轉 z

minc routptr

當使用本發明之可程式繪圖共處理器，結合主電腦系統，例如超級NES，可產生如圖19，20與21範例之某些特別效應。如圖19所示，目標亦即直升機的側視圖被畫出。此圖不精確地反射出使用瑪利歐晶片所產生的高品質顯示。圖20與21係前述圖19的放大與旋轉圖。使用本發明的繪圖共處理器，產生3D型(和其它)特別效應，包括在高速旋轉比例的變化多邊形基目標。而只有對主電視遊樂器處理系統有最小的負擔。

本發明已詳細說明，本發明之詳細揭示只是說明與範例之目的。當前述實施例做為最佳實施例，對熟悉該行業之專業人士而言，仍可有許多變化與修改。如下申請專利範圍將該等變化與修改，包含在本發明之精神與範圍之內。

(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: 資訊處理系統之外部記憶系統)

一種可完全程式繪圖微處理器，該微處理器係設計成實施在一個可移動之外部記憶體單元內，俾連接於一個主資訊處理系統。於一示範實施例中，係說明一個電視遊樂器(video game)系統，其中包括一個主電視遊樂器系統，與一個內含有繪圖微處理器之可插入式電視遊樂器卡匣。該遊樂器卡匣包括一個儘讀程式記憶體(ROM)，與一個隨機存取記憶體(RAM)。繪圖處理器係與遊樂器卡匣的一個三匯流排結構連接運作。繪圖處理器使用這種匯流排結構，可從程式ROM，外部RAM或其內部外速RAM中執行程式。全使用者可程式繪圖處理器具有一個指集，俾有效地實施有關3-D繪圖之算術操作，例如，包括專硬體執行特殊指令，使用於主電視遊樂器系統的字元映至(map)顯示上，畫出每個像素，即從程式者的觀點，經由允許每個像素的定址而產生一個“虛”位元映至(bit map)，即使該主系統為字元基(character based)。繪圖共處理器與主共處理器互相作用，因此主處理器可在任何時間存取繪圖共處理器的16個一般暫存器。

英文發明摘要(發明之名稱: EXTERNAL MEMORY SYSTEM FOR AN INFORMATION PROCESSING SYSTEM)

A fully programmable, graphics microprocessor is disclosed which is designed to be embodied in a removable external memory unit for connection with a host information processing system. In an exemplary embodiment, a video game system is described including a host video game system and a pluggable video game cartridge housing the graphics microprocessor. The game cartridge also includes a read-only program memory (ROM) and a random-access memory (RAM). The graphics coprocessor operates in conjunction with a three bus architecture embodied on the game cartridge. The graphics processor using this bus architecture may execute programs from

附註：本案已向 美 國(地區)申請專利、申請日期：1992.1.30 案號：07/728,201

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

四、中文發明摘要(發明之名稱：)

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

英文發明摘要(發明之名稱：)

either the program ROM, external RAM or its own internal cache RAM. The fully user programmable graphics coprocessor has an instruction set which is designed to efficiently implement arithmetic operations associated with 3-D graphics and, for example, includes special instructions executed by dedicated hardware for plotting individual pixels in the host video game system's character mapped display which, from the programmer's point of view, creates a "virtual" bit map by permitting the addressing of individual pixels -- even though the host system is character based. The graphics coprocessor interacts with the host coprocessor such that the graphics coprocessor's 16 general registers are accessible to the host processor at all times.

附註：本案已向

國(地區) 申請專利，申請日期：

案號：

第 81104859 號 專 利 申 請 案

申 請 專 利 範 圍 修 正 本

(82年 6 月 19 日)

1. 一個資訊處理系統(20,19)之外部記憶系統(19), 該資訊處理系統包括一個顯示銀幕畫面(36), 並具有一個微處理器, 用於執行一個影像繪圖程式, 以及一個影像記憶體(30), 用於儲存表示多個字元的字元資料, 其當被組合時定義一個顯示圖框; 該外部記憶體系統(19)之特徵在於:

一個程式記憶體(10), 用於儲存至少某些該影像繪圖程式的指令; 以及

一個轉換電路(2,52), 其耦合到該程式記憶體(10), 用於接收以一個像素規格表示的顯示資料, 處理該像素規格與轉換該像素規格資料, 成為被該影像記憶體(30)所使用的字元資料形式。

2. 如申請專利範圍第1項之外部記憶體系統, 所述之該像素規格, 包括座標資料定義在顯示銀幕畫面(36)上像素的位置, 與該轉換電路(2,52), 從該程式記憶體接收該像素規格。
3. 如申請專利範圍第2項之外部記憶體系統, 所述該轉換電路(52), 包括位址轉換電路(202,204,Ha,Fa,214,216), 用於接收像素座標資料, 並於產生一個字元指定位址。
4. 如申請專利範圍第1項之外部記憶體系統, 其中復包括一個隨機存取記憶體(RAM)(6,8), 與緩衝記憶體裝置(206), 其用於暫時儲存產生自該轉換電路之字元資

料，以及用於耦合儲存在該緩衝記憶體的字元資料該RAM(206,228,88)之裝置。

5. 如申請專利範圍第2項之外部記憶體系統，其中復包括記錄裝置，俾用於暫時儲存來自該程式記憶體(202,204)之像素座標資料。
6. 如申請專利範圍第1項之外部記憶體系統，其中復包括一個可程式繪圖處理器(2)，耦合到該程式記憶體，並且該轉換電路(52)在該繪圖處理器實施。
7. 如申請專利範圍第6項之外部記憶體系統，所述之該可程式繪圖處理器，包括一個第一來源共同匯流排(X)，一個第二來源共同匯流排(Y)，與一個目的共同匯流排(Z)，該轉換電路從該第一來源匯流排(X)，與該第二來源匯流排(Y)接收資料，並且用於向前資料到該目的匯流排(Z)。
8. 如申請專利範圍第1項之外部記憶體系統，所述該像素規格，包括一個像素的顯示座標，與相關於該顯示座標之色彩資訊，並進一步包括色彩暫存器裝置(54)，用於接收和暫時儲存該色彩資訊。
9. 如申請專利範圍第8項之外部記憶體系統，其中復包括一個暫存器矩陣(206)，用於從該色彩暫存器裝置接收像素色彩資訊。
10. 如申請專利範圍第1項之外部記憶體系統，其中復包括一個隨機存取記憶體(RAM)(6,8)，與一個RAM控制器(88)，用於控制存取該RAM，與該轉換電路，包括位址轉換裝置，用於產生一個字元位址(202,204,HA,FA,

214,216)與字元資料產生裝置，用於產生字元資料，其包括資料係相對於該像素規格(202,206)，以及用於傳送(206,216)，利用該轉換電路所產生的字元位址與字元資訊，到該RAM控制器(88)之裝置。

11.如申請專利範圍第1項之外部記憶體系統，其中復包括一個隨機存取記憶體(RAM)(6,8)，與用於耦合字元資訊到該轉換電路之裝置，俾從相關於該像素規格之該RAM，利用該轉換電路(82,206)處理該像素規格。

12.如申請專利範圍第1項之外部記憶體系統，其中復包括儲存字元資料之裝置，其中該轉換電路(52)包括用於接收之裝置，從該用於儲存之裝置，被顯示於像素附近之有關其它像素之字元資料資訊目前被處理中。

13.如申請專利範圍第1項之外部記憶體系統，所述之程式記憶體(10)與該轉換電路(2,52)，係在電視遊樂器卡匣(19)內實施。

14.如申請專利範圍第1項之外部記憶體系統，所述之該轉換電路，包括一個色彩矩陣(206)，用於儲存資料，此資料相關於指定像素與在此字元的其它像素，其中包括此指定像素。

15.如申請專利範圍第14項之外部記憶體系統，所述之該色彩矩陣(206)包括多數個行和列，此色彩矩陣利用列載入並且利用行讀出。

16.如申請專利範圍第15項之外部記憶體系統，其中復包括暫存器裝置(202)，用於暫時儲存該像素規格，包括像素座標資料，與定址該色彩矩陣(206)，一部份則利

用至少部份該暫存器裝置的內容。

17. 如申請專利範圍第1項之外部記憶體系統，其中復包括位元未定裝置，用於記錄是否將被處理像素係為部份將被處理的目前字元(210)。
18. 如申請專利範圍第17項之外部記憶體系統，其中復包括隨機存取記憶裝置(RAM)(6,8)，與用於傳送利用該轉換電路(52)所產生之字元資料，到該RAM之裝置，其係依該位元未定裝置(210)的狀態而定。
19. 如申請專利範圍第1項之外部記憶體系統，其中復包括指示不需要更新(210)的處理有關一個像素規格之一個字元的位元之裝置。
20. 如申請專利範圍第19項之外部記憶體系統，所述之該轉換電路(52)，包括產生一個字元位址之裝置，以響應於用在指示該裝置的一個預先決定狀態。
21. 如申請專利範圍第1項之外部記憶體系統，其中復包括位址暫存器裝置，用於儲存利用該轉換電路(216)所產生的一個字元位址，與一個位址比較器(218)，耦合到位址暫存器裝置，用於比較利用該轉換電路所產生的目前字元位址與一個先前產生的位址。
22. 如申請專利範圍第21項之外部記憶體系統，其中復包括字元記憶體裝置，用於儲存字元資料(6,8)，與控制裝置(200)，響應於該位址比較器，用於寫出儲存在該位址暫存器裝置之位址，到該字元記憶體裝置。
23. 一種用於顯示銀幕畫面的繪圖處理器(2)，其中包括：
接收一個像素規格(56,58)的裝置；以及

一個轉換電路(25)，用於處理該像素規格，並且產生一個字元規格指定包括指定像素之一個字元。

24.如申請專利範圍第23項之繪圖處理器，所述之該像素規格，包括一個像素的顯示座標，與相關於該顯示座標之色彩資訊，其中復包括色彩暫存器裝置，用於接收與暫時儲存該色彩資訊。

25.如申請專利範圍第23項之繪圖處理器，其中復包括一個暫存器矩陣(206)，用於從該色彩暫存器裝置(54)，接收像素色彩資訊。

26.如申請專利範圍第23項之繪圖處理器，其中復包括一個隨機存取記憶體(RAM)(6,8)，以及用於從該RAM耦合字元資訊到該轉換電路之裝置，其利用該轉換電路(82,206)處理相關於該像素之規格。

27.一種用於一個資訊處理系統(20,19)的繪圖處理器(2)，此系統具有一個主處理單元(20)，用於執行儲存在至少一個記憶體裝置(10)的一個影像繪圖程式，該繪圖處理器(2)其特徵在於：

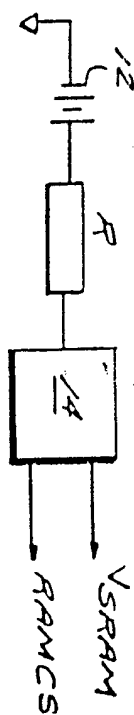
用於從該至少一個記憶體(INSTR,62,60)接收程式指令之裝置；以及

響應於至少一個預先決定程式指令之裝置，用於轉換相關於該至少一個預先決定指令之像素基格式，成為一個字元基資料格式(52)。

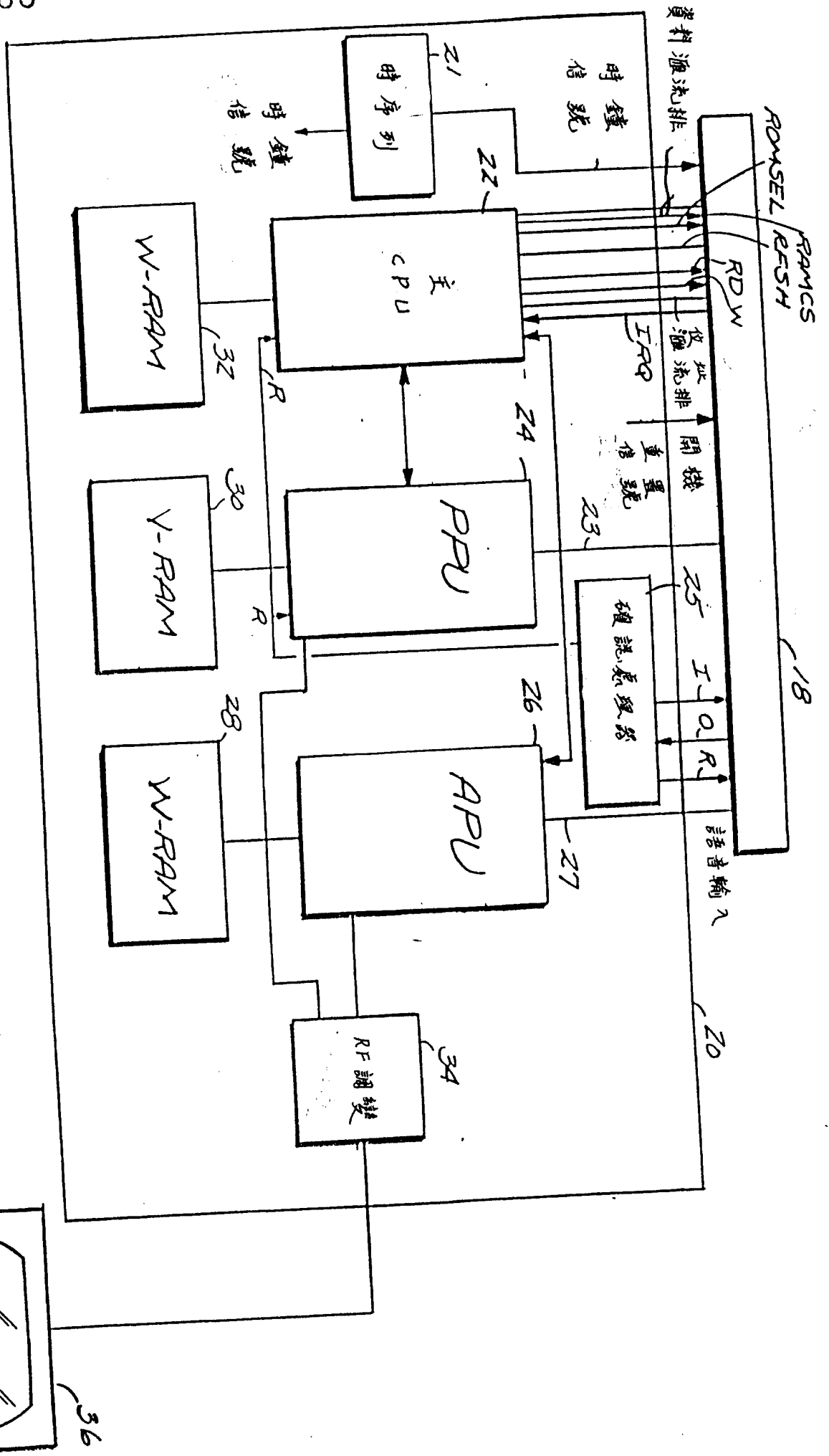
28.如申請專利範圍第27項之繪圖處理器，其中復包括一個第一來源共同匯流排(X)，一個第二來源共同匯流排(Y)，與一個目的共同匯流排(Z)以及於轉換從該第

一來源匯流排與該第二來源匯流排所接收資料，與向前資料到該目的匯流排之裝置。

29. 如申請專利範圍第27項之繪圖處理器，所述之該繪圖處理器用於耦合到一個隨機存取記憶體(RAM)(6,8)，其中復包括一個RAM控制器(88)，用於控制到該RAM之存取，該用於轉換(52)之裝置，其包括位址轉換裝置用於產生一個字元位址(202,204,HA,FA,214,216)，與字元資料產生裝置，用於產生字元資料，包括相關於該像素基資料(202,206)，以及用於傳送(206,216)之裝置，於利用轉換電路的該裝置所產生字元位址與字元資料，到該RAM控制器(88)。

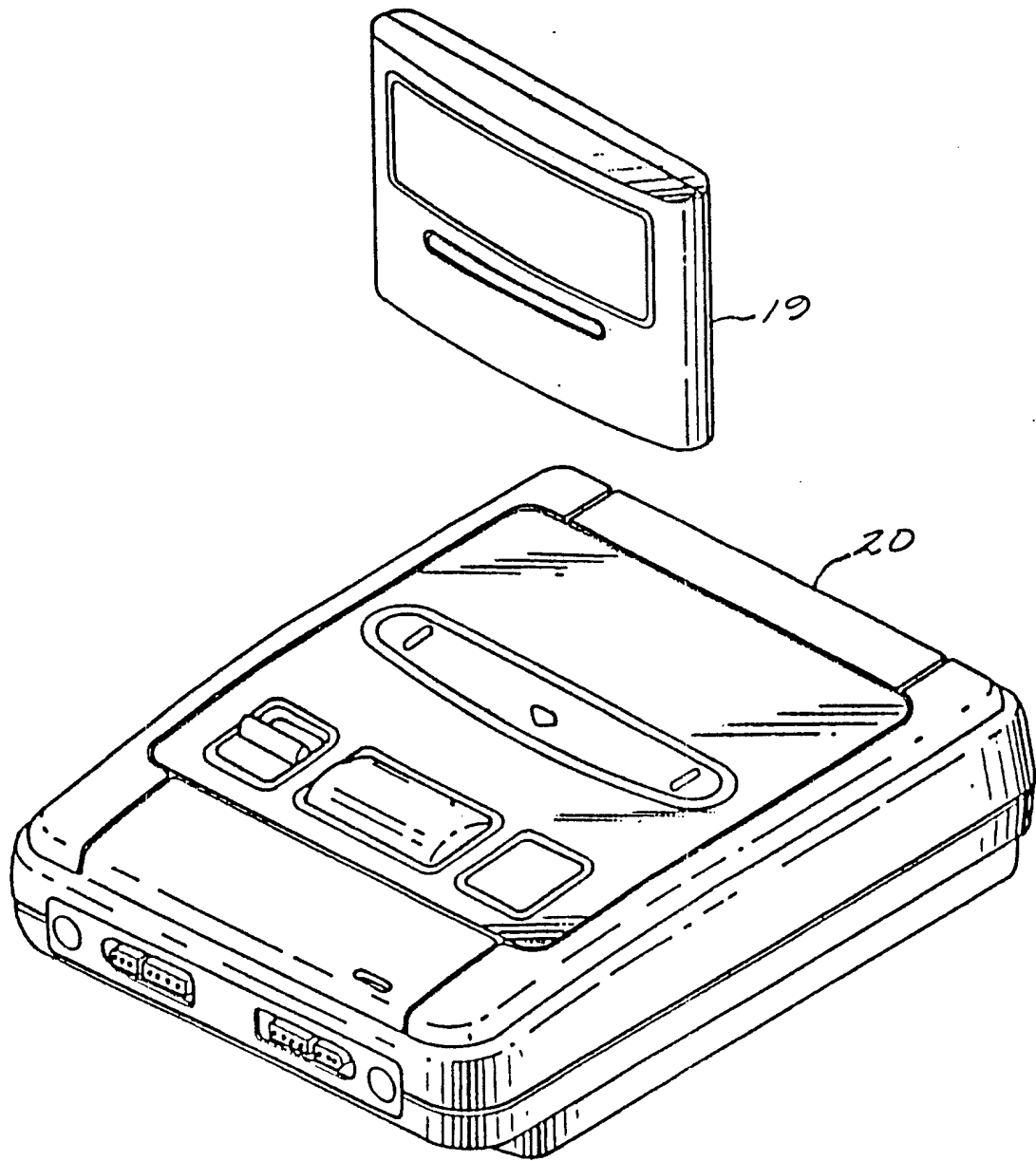


第一圖



第2圖

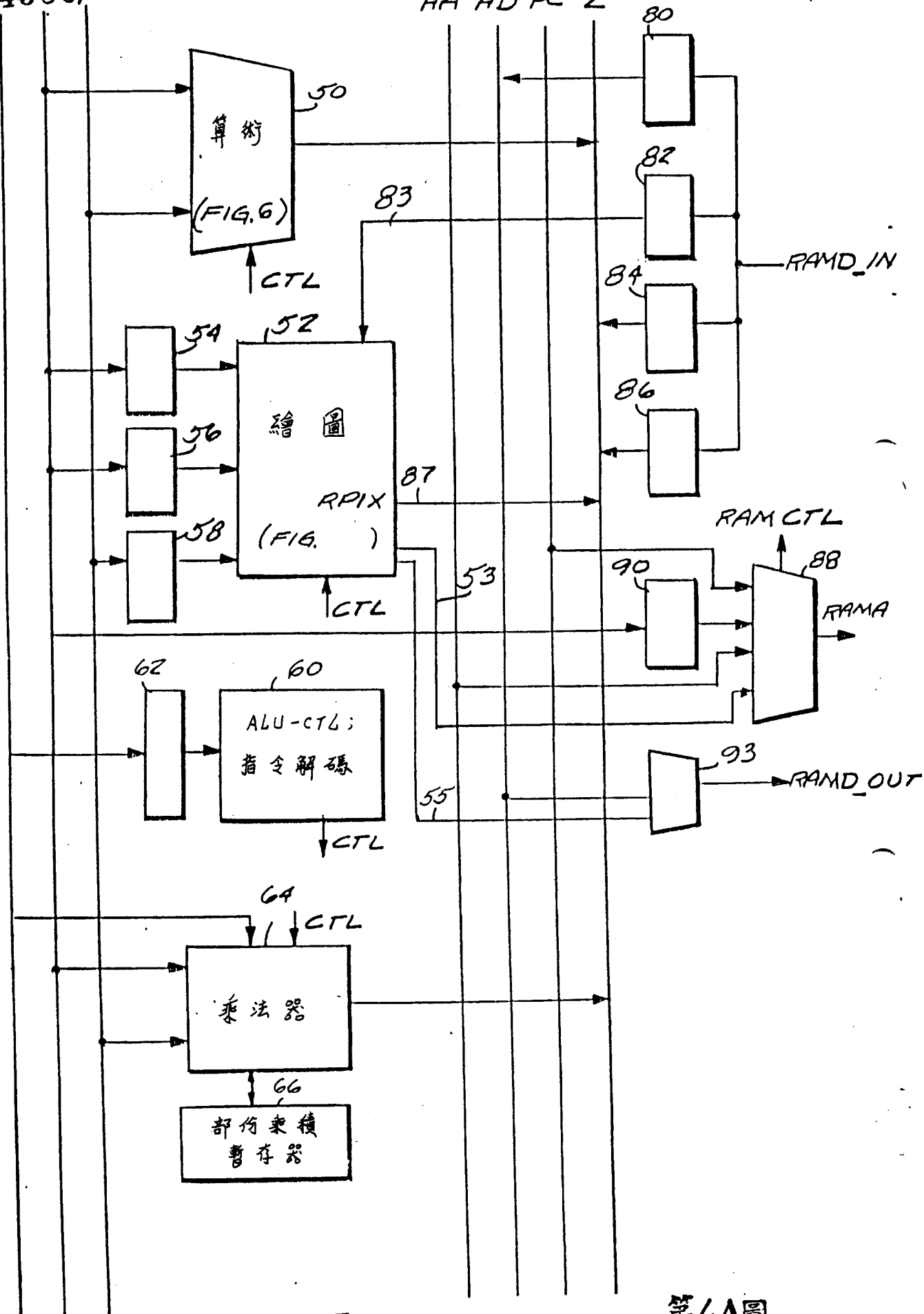
214588



第3圖

12514588Y

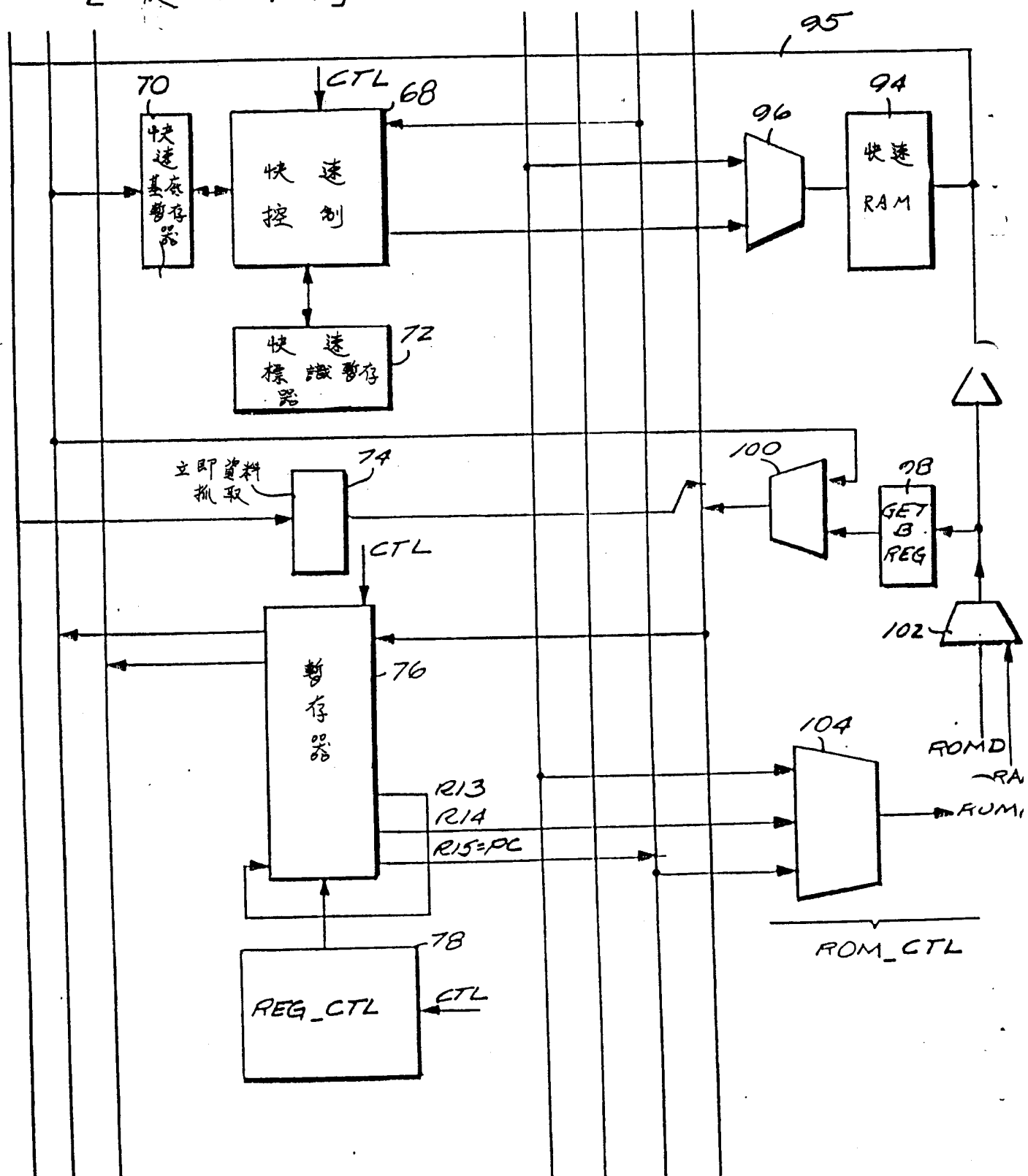
HA HD PC Z



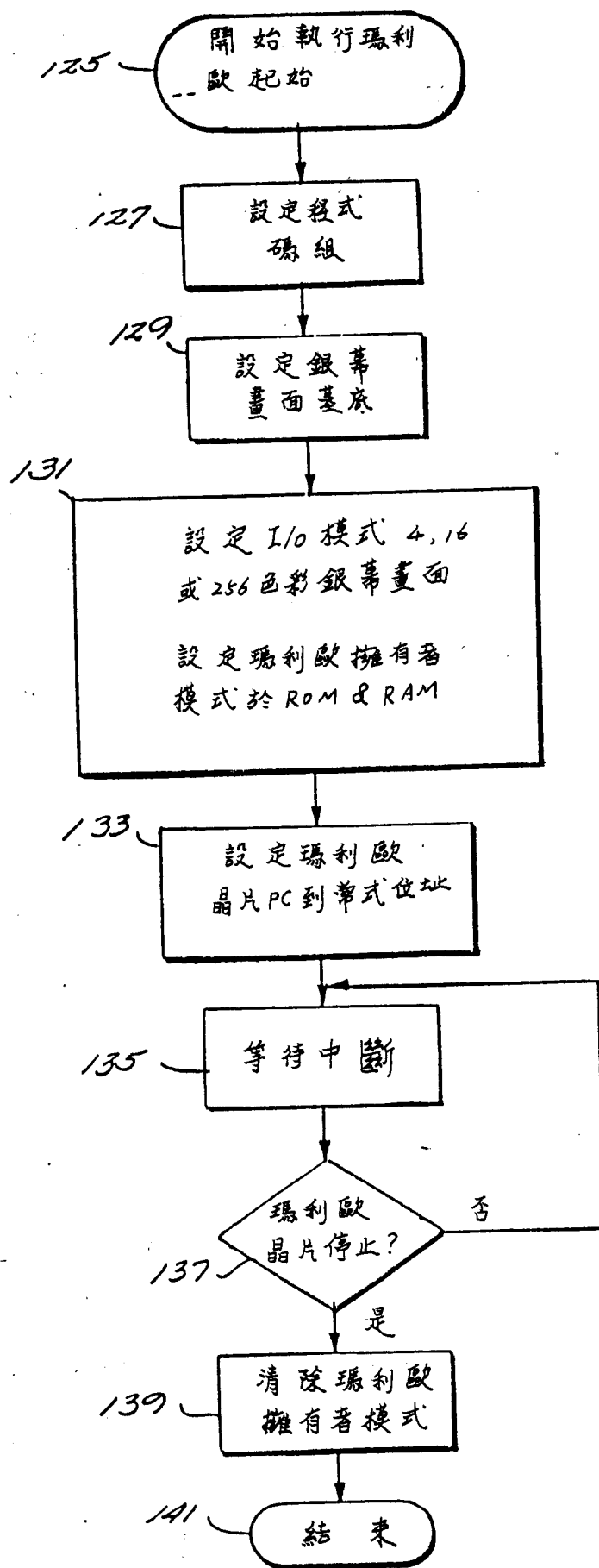
[到 FIG. 4B]

第4A圖

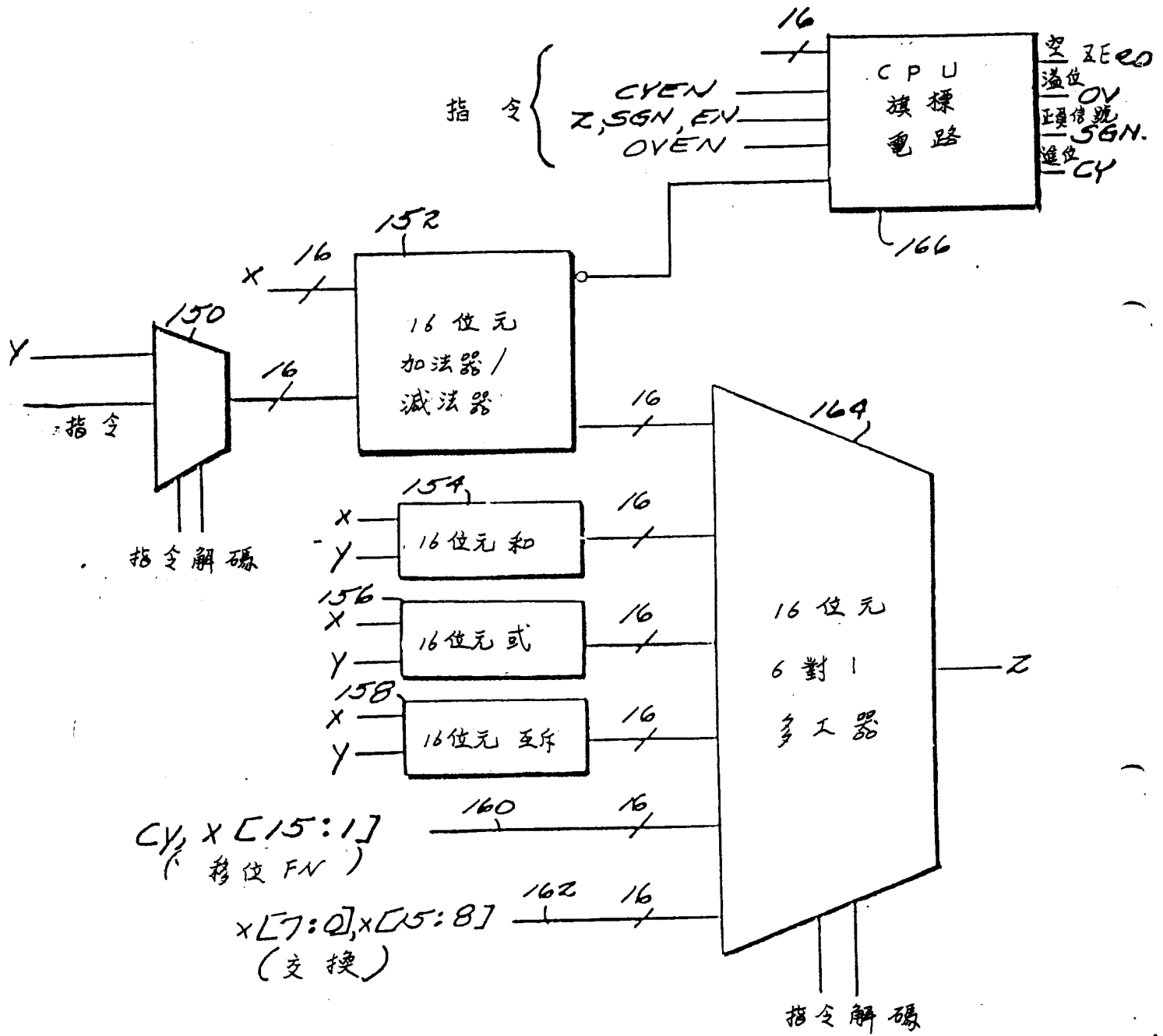
[從 FIG. 4A]



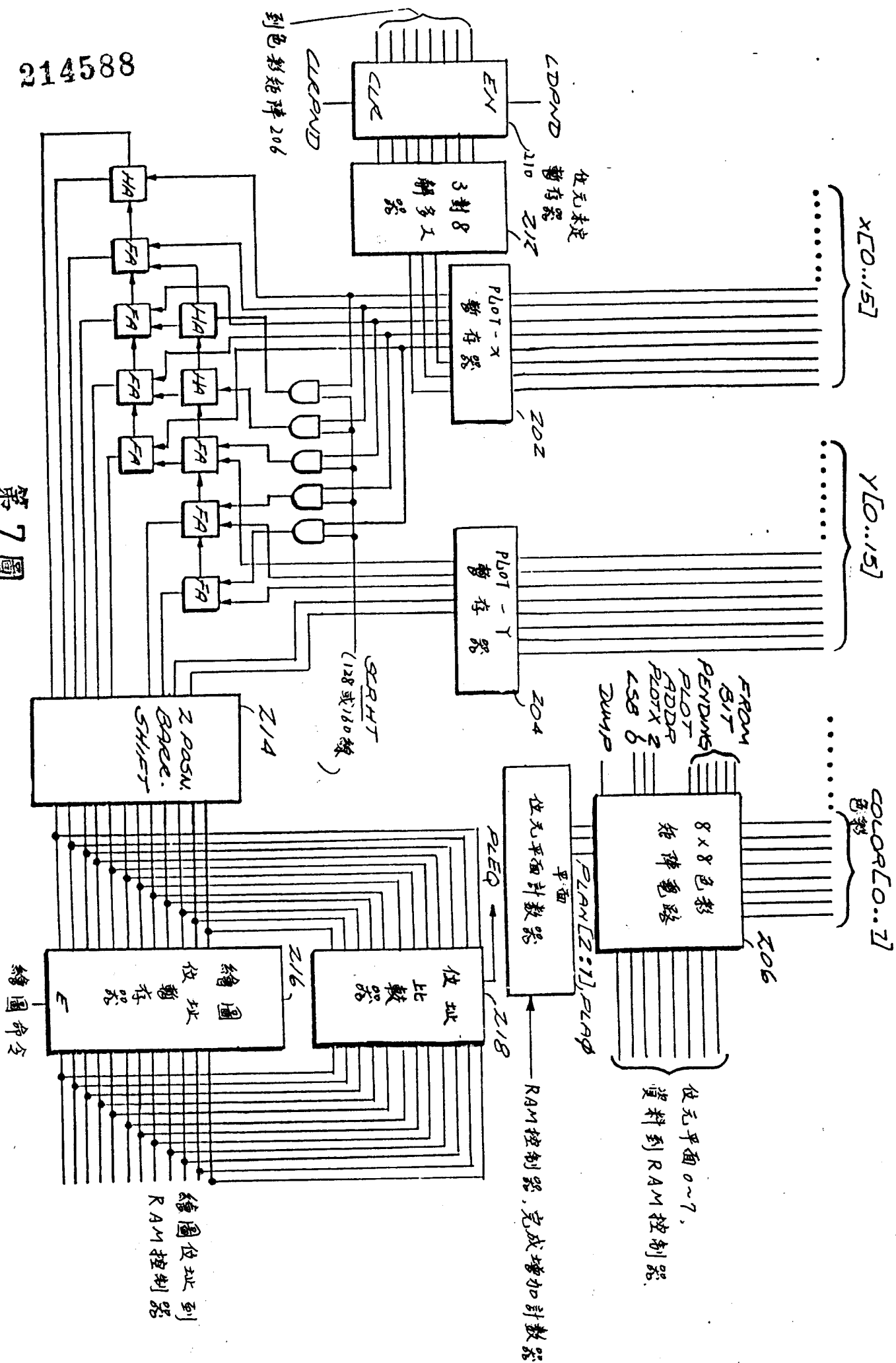
第4B圖



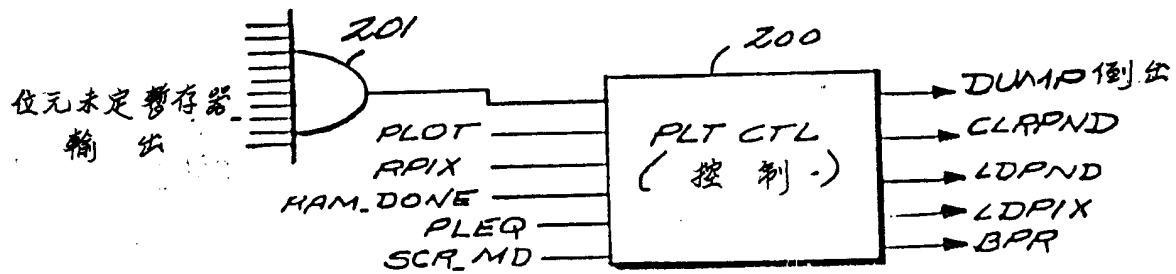
第 5 圖



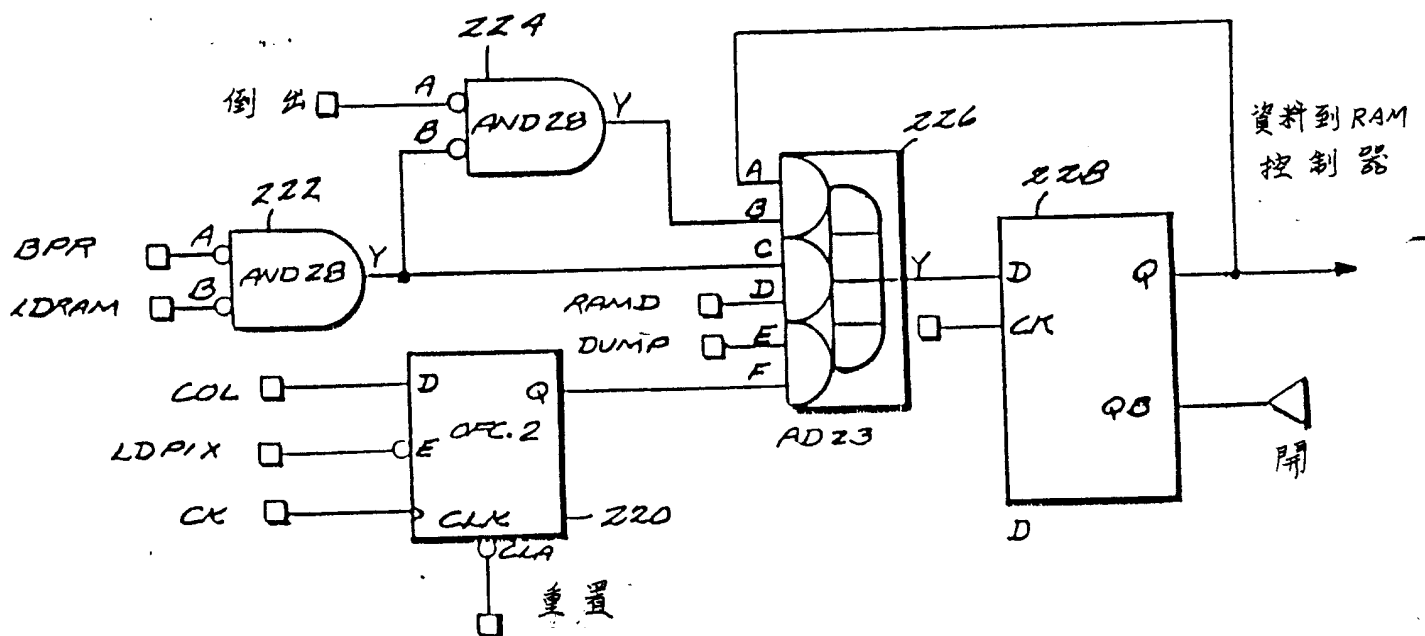
第6圖



214588



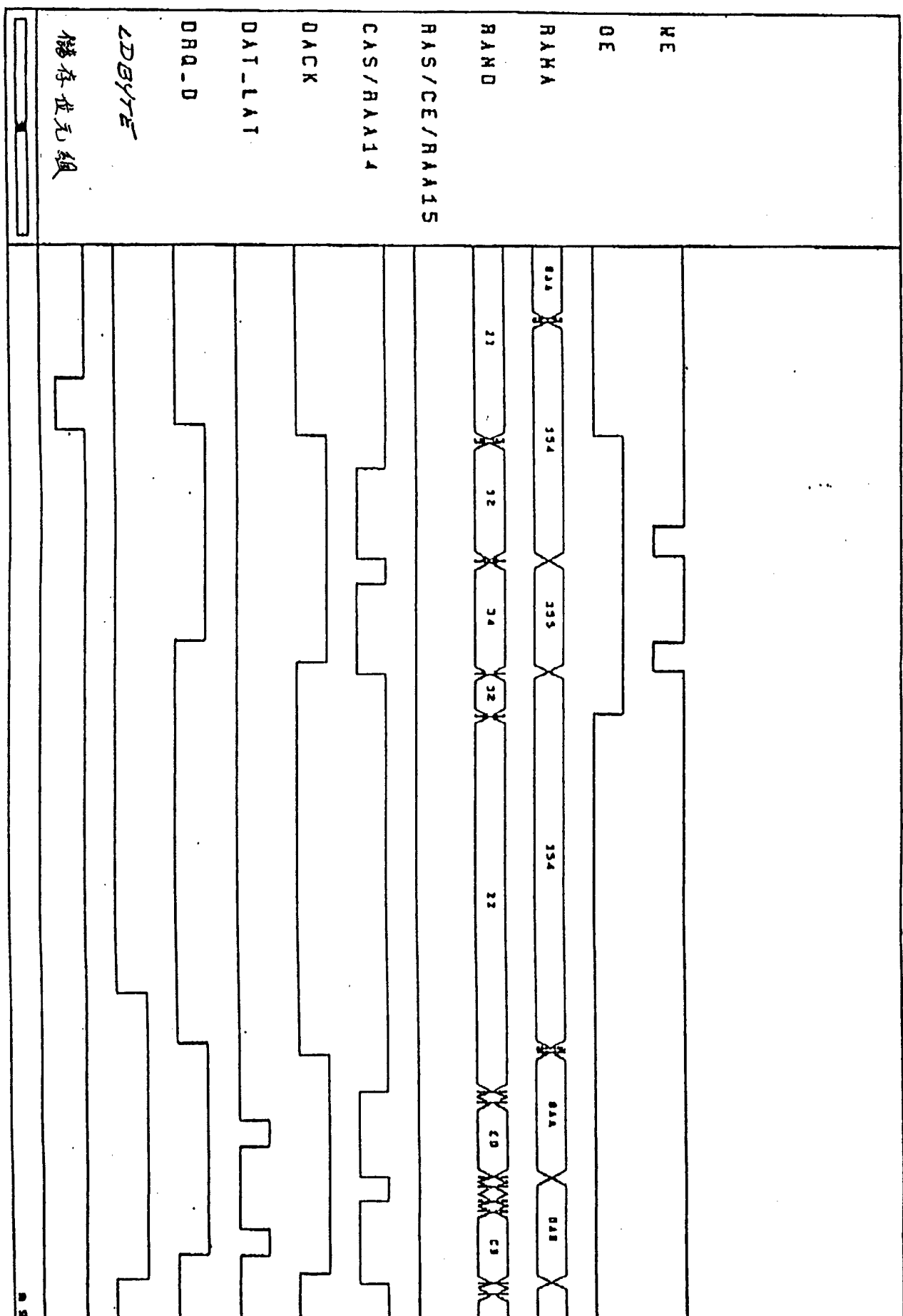
第8A圖

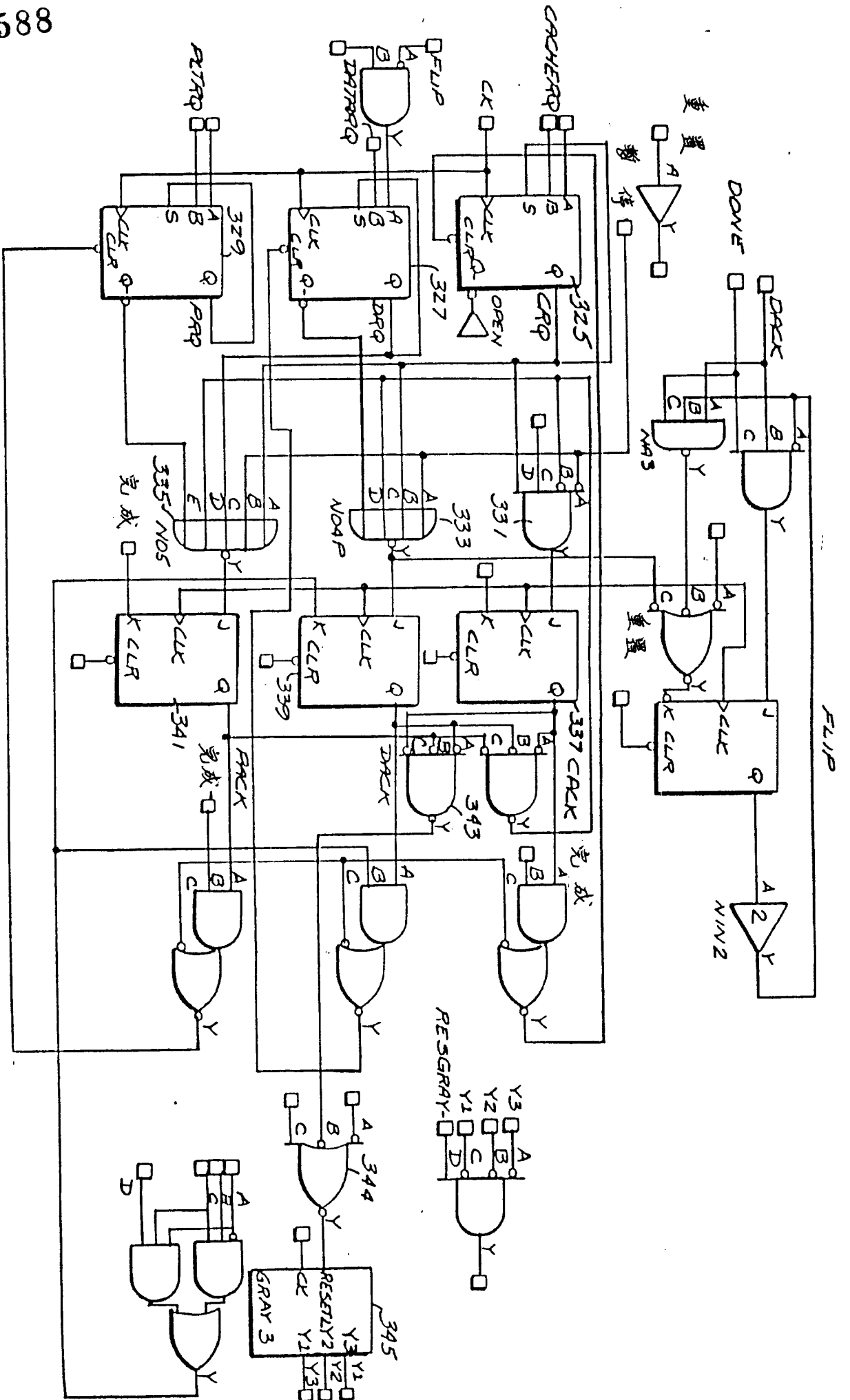


第8B圖

WE	OE	RAMA	RAMDI	RAMDO	BPR	MATEIX (BUEFLOHENTS)	AMTX (BUEFLOHENTS)	色 彩	DUMP	終 端 圖	APIX	LDPND	CLRPND	RAMDNE	PLED	LDRAH	R1	R2	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99																																																																																																																																																																																																																																																																																																																																																																																																																											
		2AB	2AA	2A8	2A6	2AC	2AD	2AE	2AF	2AG	2AH	2AI	2AJ	2AK	2AL	2AM	2AN	2AO	2AP	2AQ	2AR	2AS	2AT	2AU	2AV	2AW	2AX	2AY	2AZ	2BA	2BB	2BC	2BD	2BE	2BF	2BG	2BH	2BI	2BJ	2BK	2BL	2BM	2BN	2BO	2BP	2BQ	2BR	2BS	2BT	2BU	2BV	2BW	2BX	2BY	2BZ	2CA	2CB	2CC	2CD	2CE	2CF	2CG	2CH	2CI	2CJ	2CK	2CL	2CM	2CN	2CO	2CP	2CQ	2CR	2CS	2CT	2CU	2CV	2CW	2CX	2CY	2CZ	2DA	2DB	2DC	2DD	2DE	2DF	2DG	2DH	2DI	2DJ	2DK	2DL	2DM	2DN	2DO	2DP	2DQ	2DR	2DS	2DT	2DU	2DV	2DW	2DX	2DY	2DZ	2EA	2EB	2EC	2ED	2EE	2EF	2EG	2EH	2EI	2EJ	2EK	2EL	2EM	2EN	2EO	2EP	2EQ	2ER	2ES	2ET	2EU	2EV	2EW	2EX	2EY	2EZ	2FA	2FB	2FC	2FD	2FE	2FF	2FG	2FH	2FI	2FJ	2FK	2FL	2FM	2FN	2FO	2FP	2FQ	2FR	2FS	2FT	2FU	2FV	2FW	2FX	2FY	2FZ	2GA	2GB	2GC	2GD	2GE	2GF	2GG	2GH	2GI	2GJ	2GK	2GL	2GM	2GN	2GO	2GP	2GQ	2GR	2GS	2GT	2GU	2GV	2GW	2GX	2GY	2GZ	2HA	2HB	2HC	2HD	2HE	2HF	2HG	2HH	2HI	2HJ	2HK	2HL	2HM	2HN	2HO	2HP	2HQ	2HR	2HS	2HT	2HU	2HV	2HW	2HX	2HY	2HZ	2IA	2IB	2IC	2ID	2IE	2IF	2IG	2IH	2II	2IJ	2IK	2IL	2IM	2IN	2IO	2IP	2IQ	2IR	2IS	2IT	2IU	2IV	2IW	2IX	2IY	2IZ	2JA	2JB	2JC	2JD	2JE	2JF	2JG	2JH	2JI	2JJ	2JK	2JL	2JM	2JN	2JO	2JP	2JQ	2JR	2JS	2JT	2JU	2JV	2JW	2JX	2JY	2JZ	2KA	2KB	2KC	2KD	2KE	2KF	2KG	2KH	2KI	2KJ	2KK	2KL	2KM	2KN	2KO	2KP	2KQ	2KR	2KS	2KT	2KU	2KV	2KW	2KX	2KY	2KZ	2LA	2LB	2LC	2LD	2LE	2LF	2LG	2LH	2LI	2LJ	2LK	2LL	2LM	2LN	2LO	2LP	2LQ	2LR	2LS	2LT	2LU	2LV	2LW	2LX	2LY	2LZ	2MA	2MB	2MC	2MD	2ME	2MF	2MG	2MH	2MI	2MJ	2MK	2ML	2MN	2MO	2MP	2MQ	2MR	2MS	2MT	2MU	2MV	2MW	2MX	2MY	2MZ	2NA	2NB	2NC	2ND	2NE	2NF	2NG	2NH	2NI	2NJ	2NK	2NL	2NM	2NN	2NO	2NP	2NQ	2NR	2NS	2NT	2NU	2NV	2NW	2NX	2NY	2NZ	2OA	2OB	2OC	2OD	2OE	2OF	2OG	2OH	2OI	2OJ	2OK	2OL	2OM	2ON	2OO	2OP	2OQ	2OR	2OS	2OT	2OU	2OV	2OW	2OX	2OY	2OZ	2PA	2PB	2PC	2PD	2PE	2PF	2PG	2PH	2PI	2PJ	2PK	2PL	2PM	2PN	2PO	2PP	2PQ	2PR	2PS	2PT	2PU	2PV	2PW	2PX	2PY	2PZ	2QA	2QB	2QC	2QD	2QE	2QF	2QG	2QH	2QI	2QJ	2QK	2QL

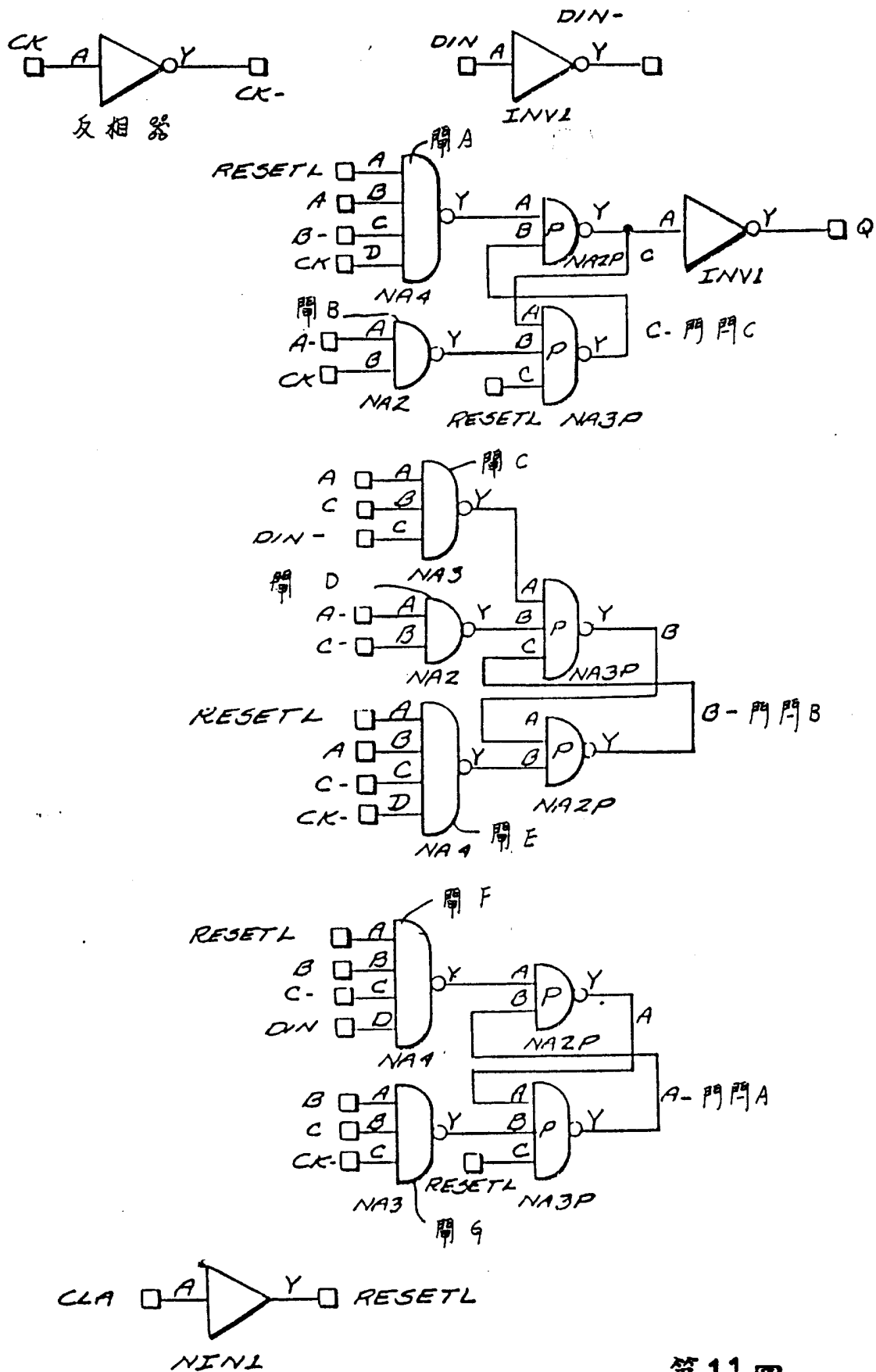
第9A圖



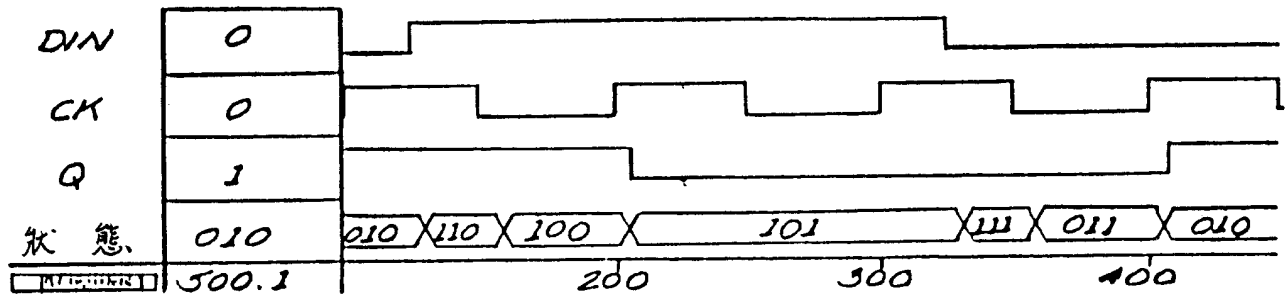


第10圖

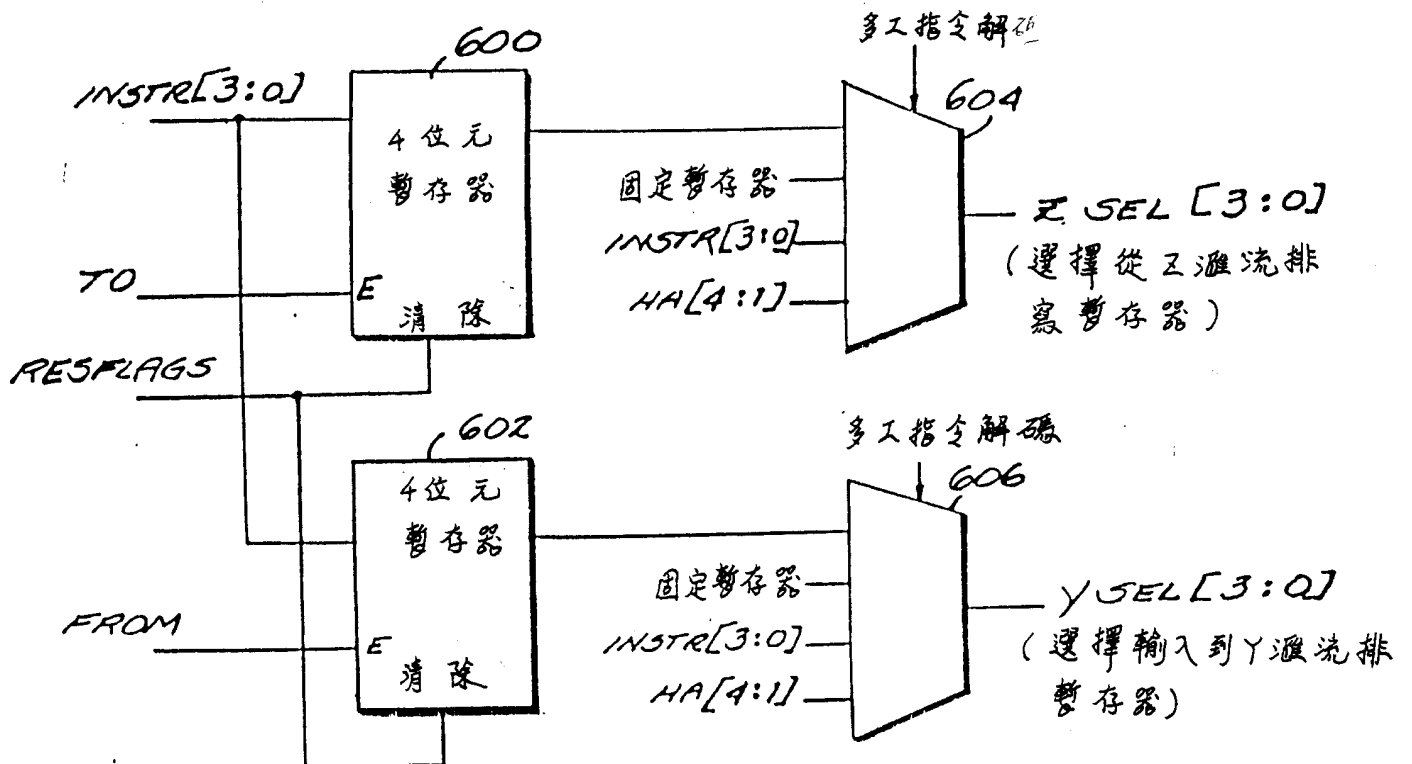
214588



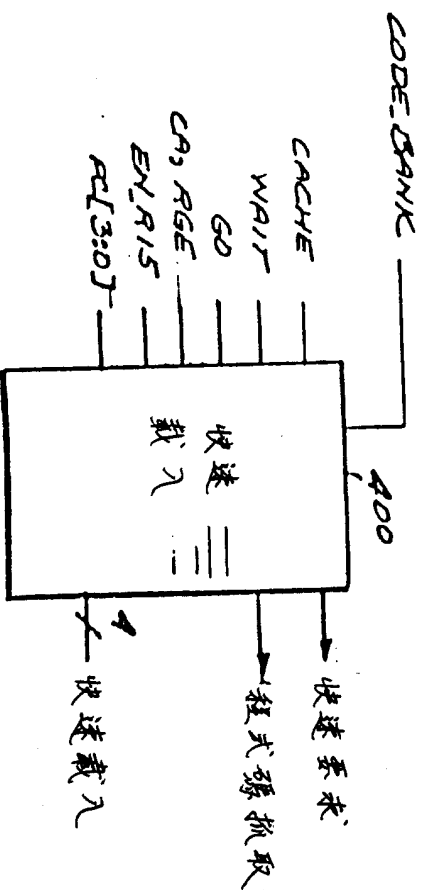
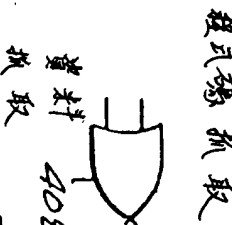
第11圖



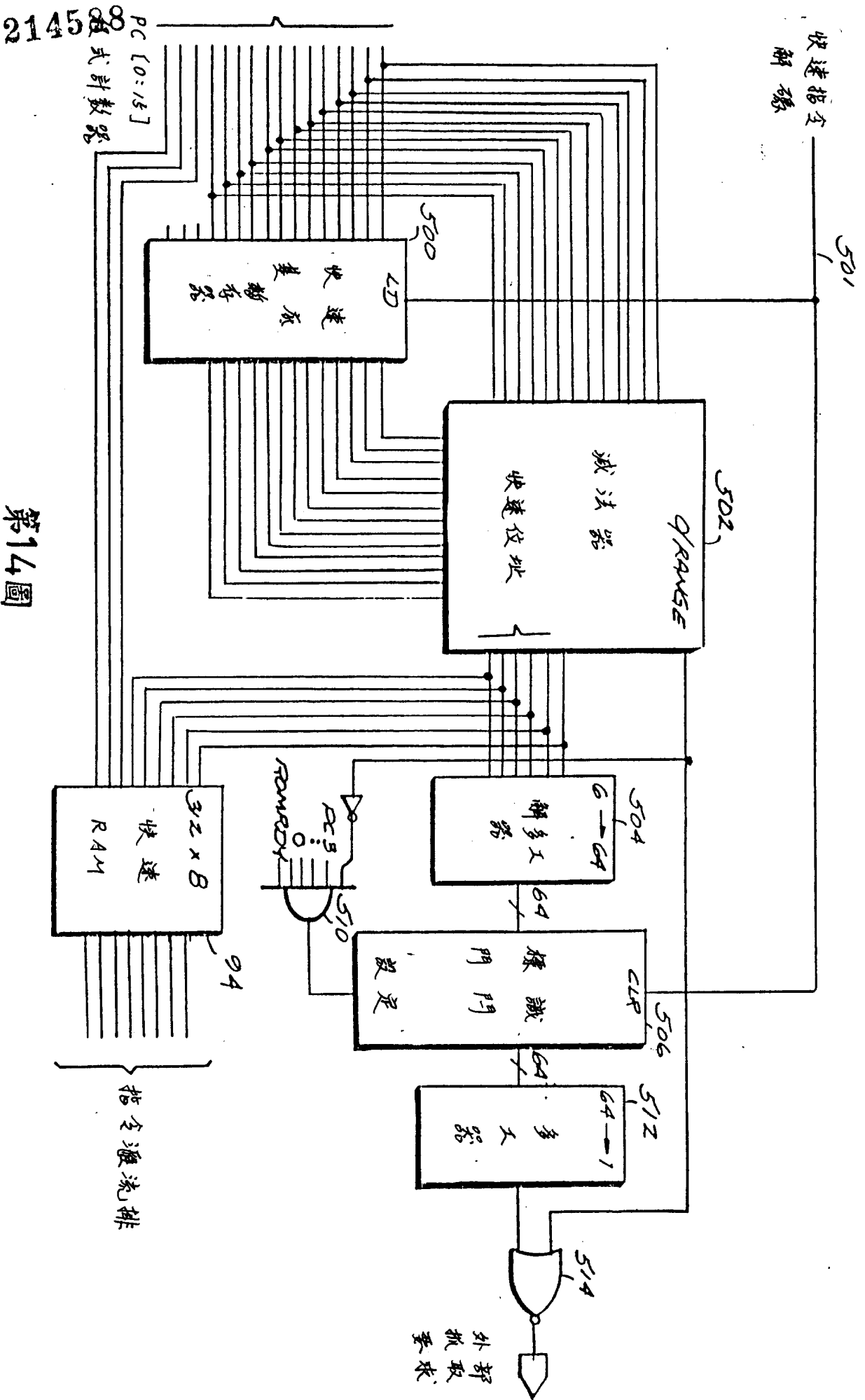
第12圖



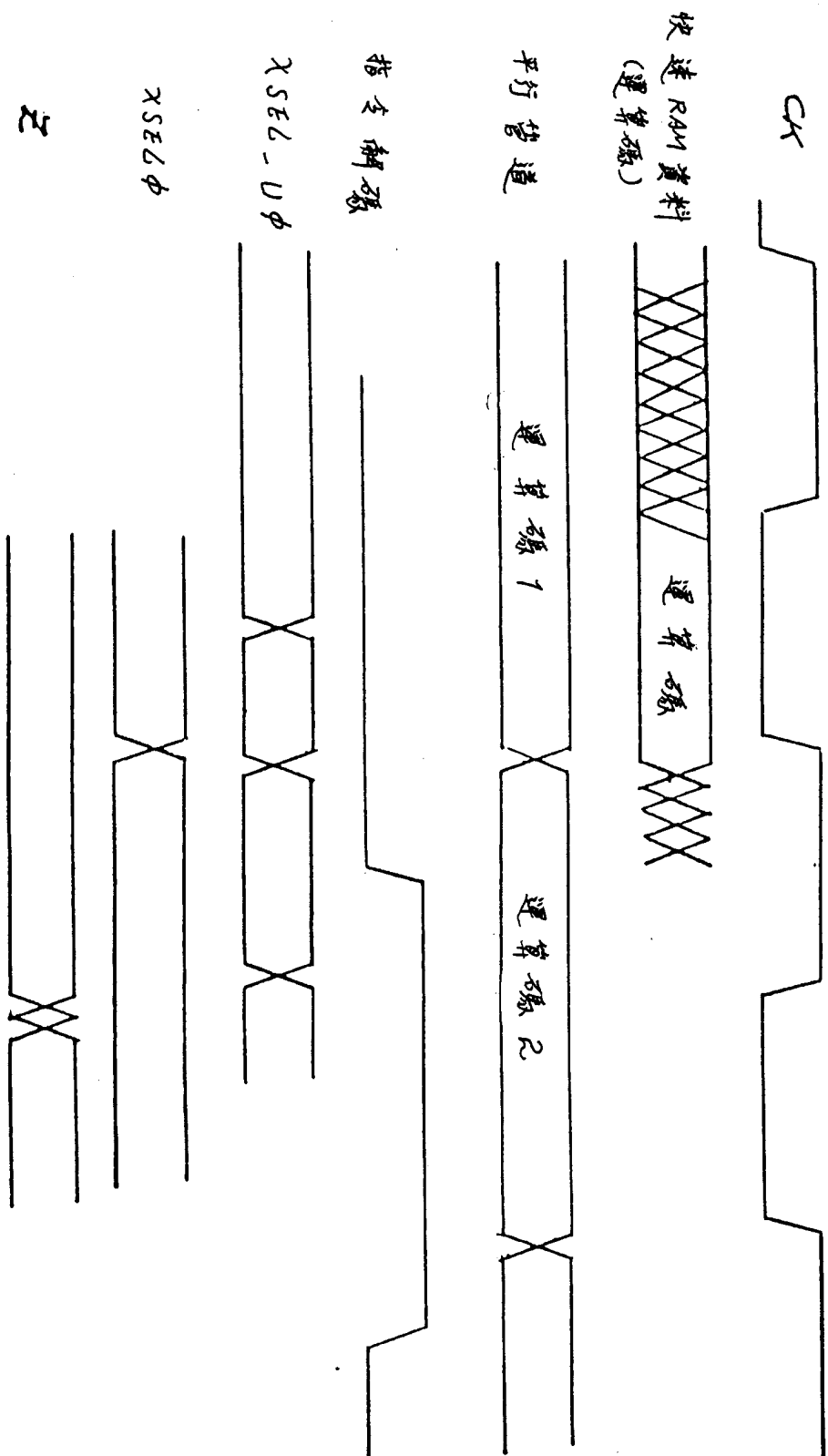
第16圖



214588

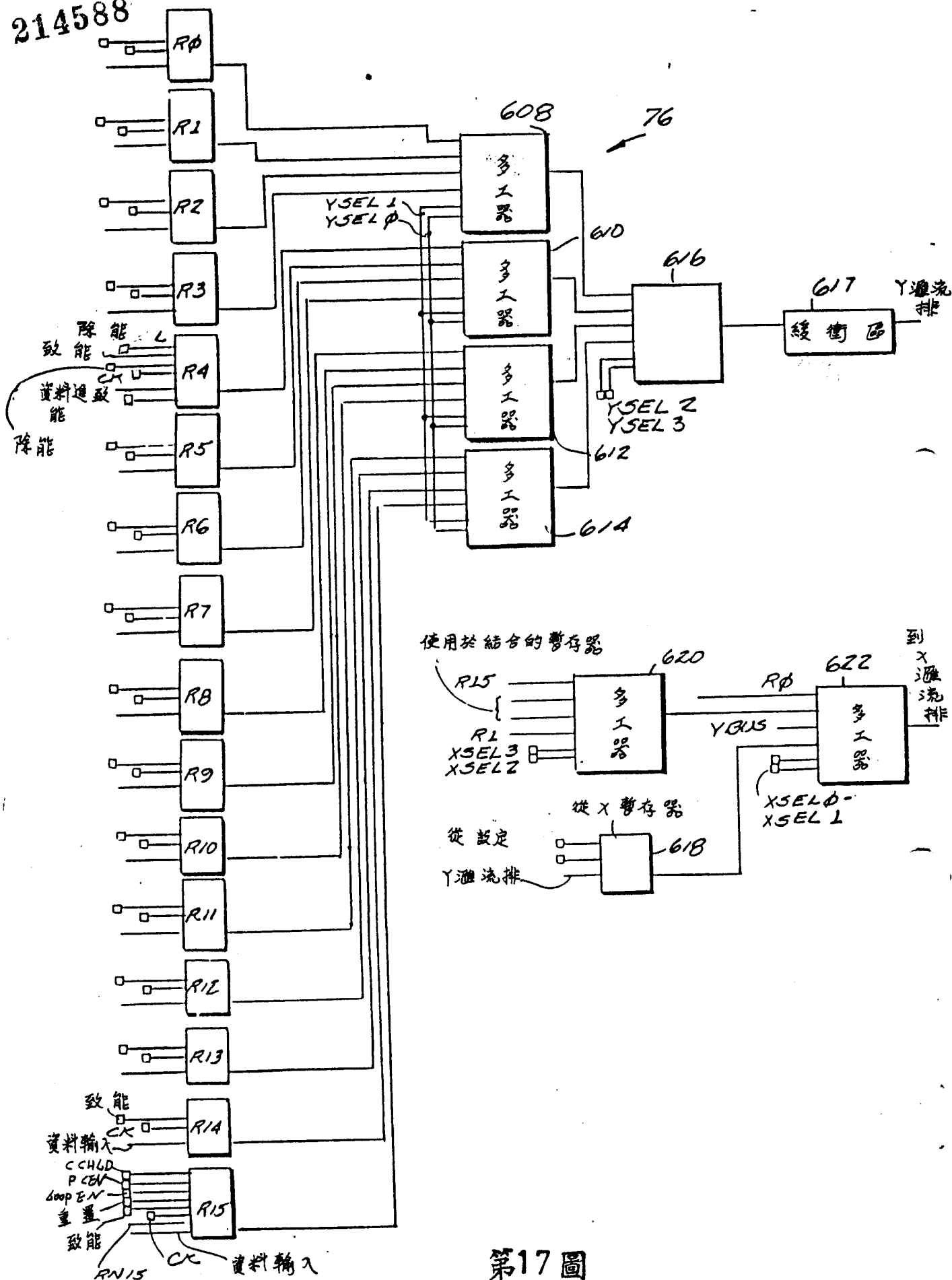


第14圖

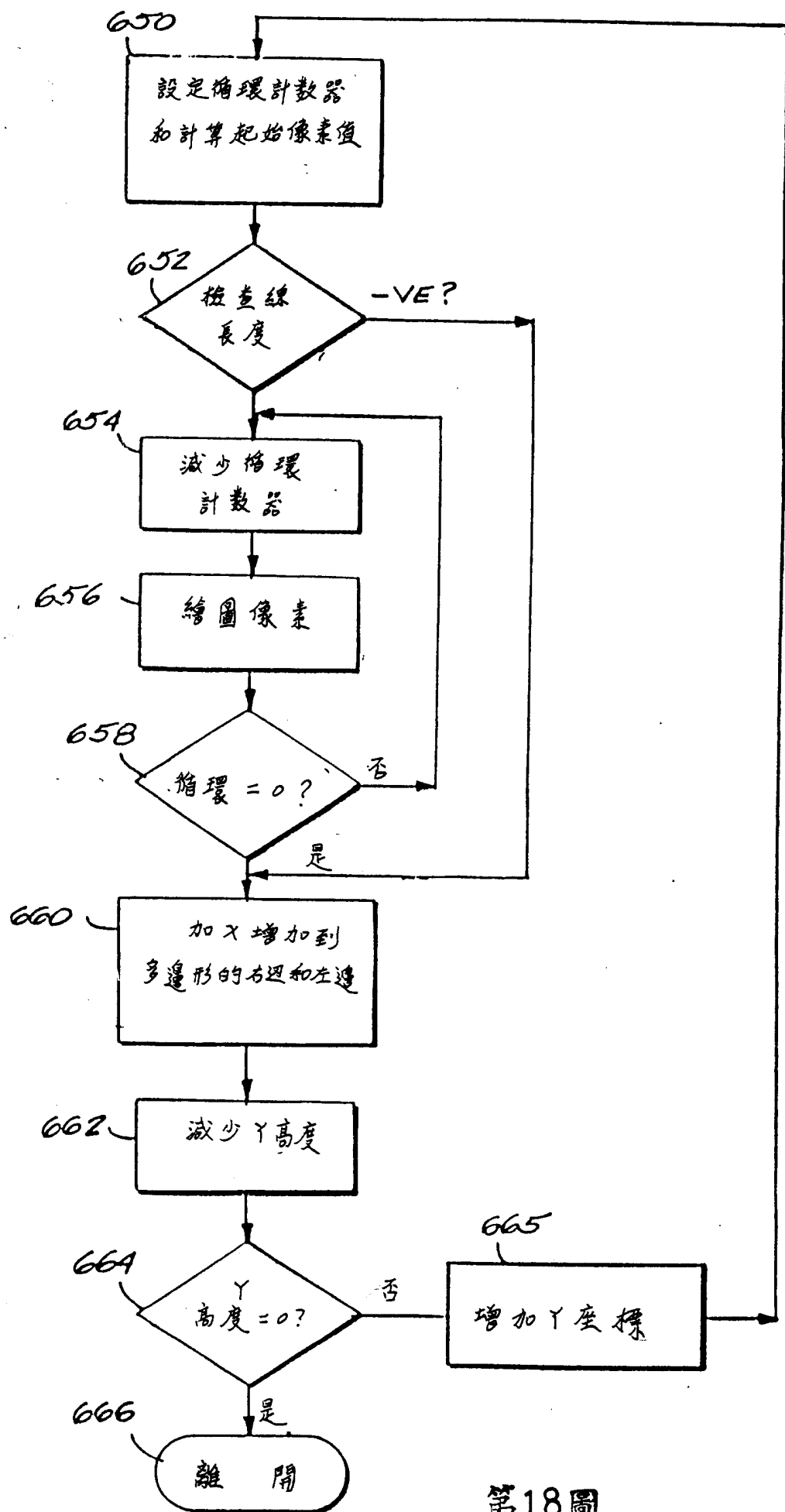


第15B圖

214588

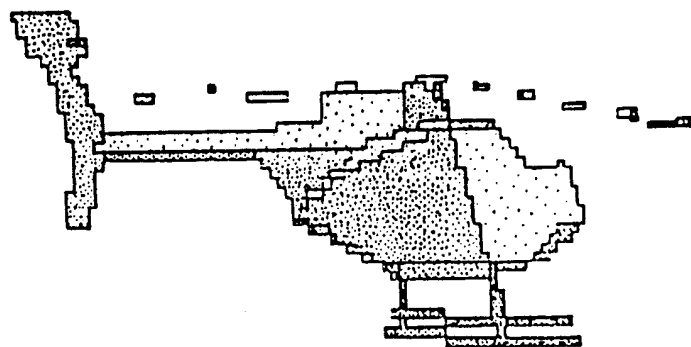


214588

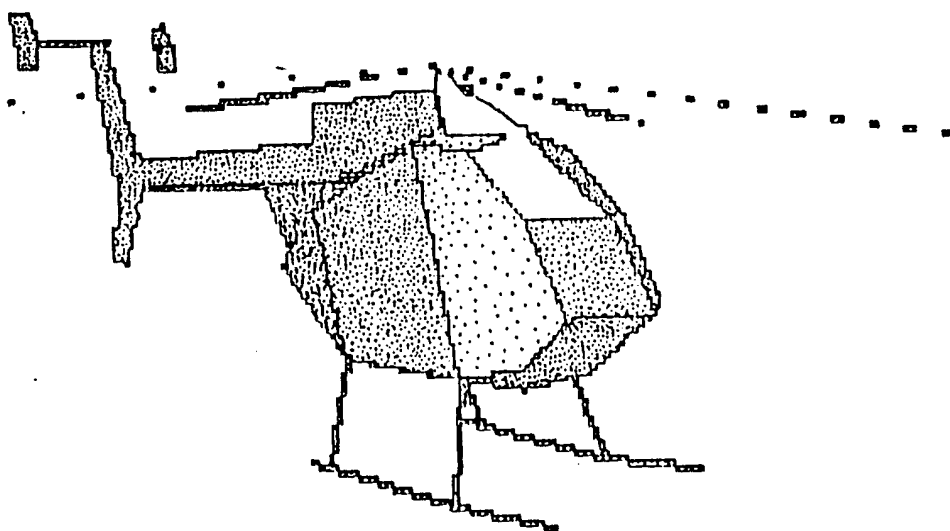


第18圖

214588

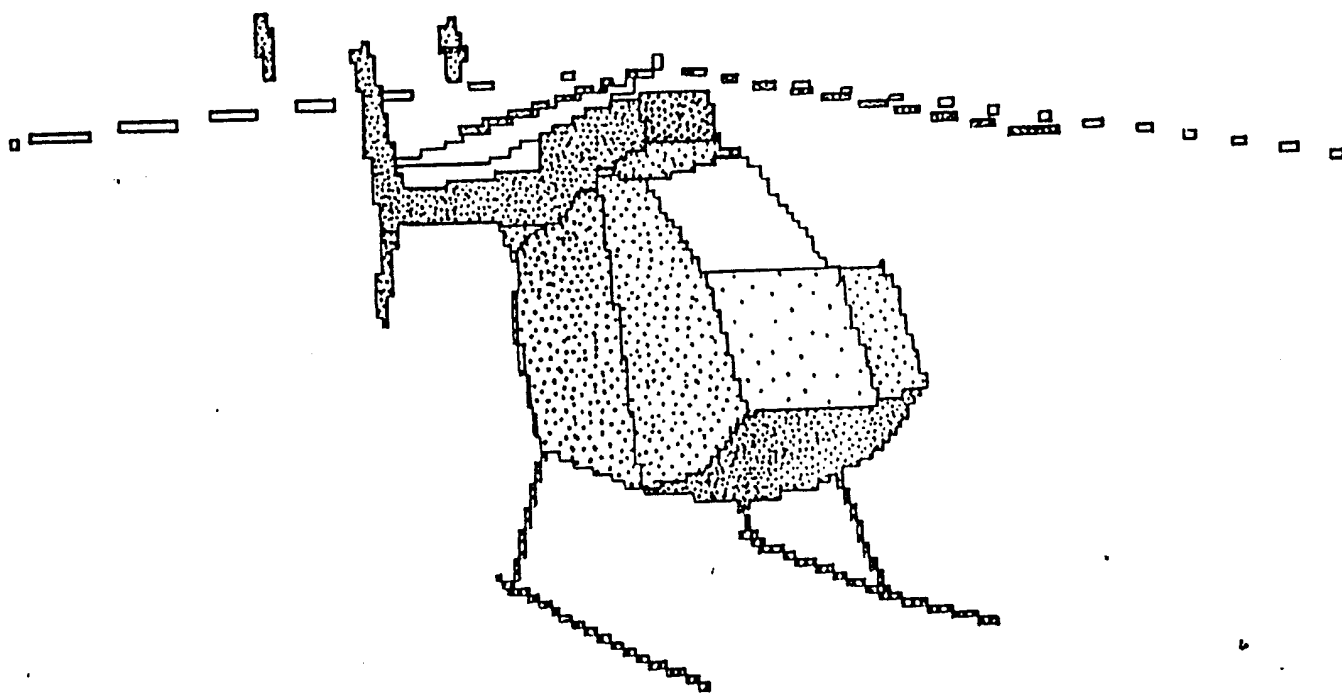


第19圖



第20圖

214588



第21圖