申請日期：	88.10.7	案號：	88117772
類別：	G06F 9/45		

（以上各欄由本局填註）

公告本

發明專利說明書 446914

一、發明名稱	中文	表格格式程式設計
	英文	Table Format Programming
二、發明人	姓名（中文）	1.林亞夫
	姓名（英文）	1.Peter Ar-Fu Lam
	國籍	1.香港
	住、居所	1.美國加州陶倫斯市威那路20104號
三、申請人	姓名（名稱）（中文）	1.與我成長有限公司
	姓名（名稱）（英文）	1.GROW WITH ME, INC.
	國籍	1.美國
	住、居所（事務所）	1.美國加州陶倫斯市C-6區第208街2401號
	代表人姓名（中文）	1.林亞夫
	代表人姓名（英文）	1.Peter Ar-Fu Lam

本案已向

國(地區)申請專利	申請日期	案號	主張優先權
美國 US	1998/10/09	09/169,462	有

有關微生物已寄存於　　寄存日期　　寄存號碼

無

五、發明說明（1）

本申請案是1995年10月2日提交的申請號為08/538,426且專利號為US 5,867,818的美國專利的追加申請案，在此引用此申請作為參考。

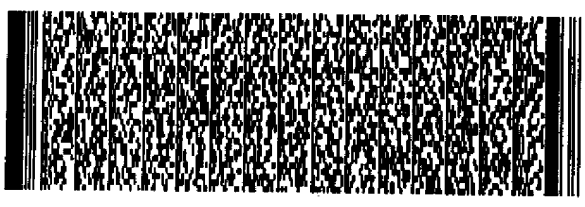
發明之範疇：

本發明有關一種利用填寫不同種類的表格來實現程式設計功能和便於控制程式流程的對計算設備進行程式設計的方法。

發明之背景：

本發明有關一種程式設計工具，該程式設計工具被設計成程式設計師與電腦之間的介面。這種程式設計方法工具廣泛使用表格來表達程式設計師的邏輯思維過程並使程式設計過程更易於為他人所理解。因此，這些改進提高了程式設計效率，減少了出現程式故障和結構錯誤的機會。另外，程式設計師學習這種程式設計方法所需的培訓成本是最低的。將來任何程式設計師都能夠很容易地讀懂和維護依據本發明而編寫的程式。

傳統的程式設計語言定義一組程式設計指令和程式設計規則。諸如BASIC、C和JAVA之類的通用程式設計語言是按照從上到下逐行順序列表的形式編寫程式。在很多應用程式中，需要編寫幾百頁代碼來描述要求的作業功能。要求程式設計師從很長的程式列表中指出一個程式的邏輯流程是非常困難的。因此，程式的長短為將來的維護工作帶




五、發明說明 (2)

來了困難。除此以外，多頁充滿了代碼行序列的程式很難被以前沒有涉及此作業的其他程式設計師所理解。雖然已經利用了大量的壓縮指令或程式設計符號，但以不同程式設計語言編寫而成的程式的長短仍超出了一個專業程式設計師能夠容易理解的範圍。在很多情況下，一個程式的原作者會發現經過了一個較長的時期之後，他也很難理解他本人所編寫的程式。壓縮指令集的要求與可讀性是彼此矛盾的。因此非常需要一種程式設計方法，該方法能夠提供壓縮的程式長度並易於被其他程式設計師及非專業人員所理解。

每次發明一種新的程式設計語言時，都定義編碼符號和指令集來描述一個程式功能。還定義規則來限制如何使用程式設計指令以便計算設備能夠翻譯和執行利用該方法編寫而成的程式。在很多情況下，為了確證所有這些指令集並向程式設計師講述程式設計規則，用以描述一種程式設計語言的指令手冊超過了兩英寸厚。利用幾個月來培訓一個程式設計師學會所有的指令集並理解一種新程式設計語言的所有程式設計規則是非同一般的。通常是利用數年的程式設計經驗而學到更多的程式設計技巧。

雖然很多程式設計語言允許程式按照該作業的結構要求提供條件跳轉、調用功能或分支到程式的其他分段，但是專業的程式設計師在花費了大量時間閱讀該程式之後仍很難解釋這些互動式分支操作在程式結構中的邏輯流程。

市場上提供了各種可用的程式設計語言，每種專用於

五、發明說明 (3)

一種特定應用或按照特殊的程式設計環境而設計。許多其他的程式設計語言被設計成程式指定硬體配置或系統。在很多情況下，不同的程式設計語言使用不同的符號或運算式來表示一個描述相似功能的指令。程式設計師在學習了多種分別具有不同格式和指令運算式的程式設計語言之後，對於應當使用什麼符號或運算式經常會感到混淆不清。因此非常需要一種能夠使程式設計師克服這些困難的新的程式設計方法。

由於大多數程式設計語言被設計成在其指定的應用環境中能夠最佳地解釋程式設計功能，所以很多涉及多種功能狀態的程式設計作業最好被分解成多個模組，每個模組是按照其指定的環境以不同語言編寫而成的。因此，非常需要一種適用於管理以不同語言編寫的程式段的集成的專用程式設計方法。隨著因網際網路的普及，非常需要一種通用的程式設計方法，該方法適用於協調不同格式的程式設計模組以形成一個組合程式，並適用於將程式下載到與主電腦距離遙遠的本地電腦上。對於這種類型的應用，程式必須是壓縮的並且能夠根據不同本地電腦的種類進行自重構。

專利號為US 5,867,818的美國專利最先引入了一種用以對具有多個輸入和輸出端子的硬體控制器晶片進行程式設計的原始表格格式方法。本申請指向在發明這種原始結構之後進行的進一步研究中所實現的各種改進。

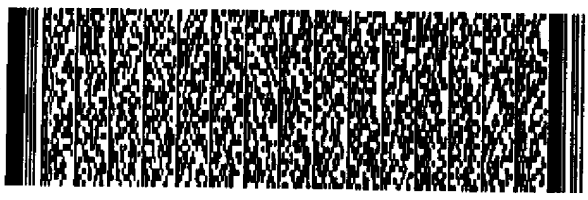
五、發明說明（4）

發明之概要：

本發明指向學習一個通用模組結構的方法以便利用表格格式程式設計的概念來對計算設備進行程式設計。本發明的一個目的是建立用以指導計算設備操作的程式設計方法的一個通用模組，並組建該模組以使普通人員無須接受專業培訓即可使用該模組。本發明的另一個目的是建立一個程式設計模組使得以這種格式編寫的任何程式易於被不具有程式設計語言方面的經驗的其他人員所理解或者可包含在已被編寫的程式中。本發明的另一個目的是建立一個功能利用編碼過程而清楚地標識程式設計作業的結構的程式設計模組，從而能夠花費最少的力量來維護該程式。其結果是，能夠相當多地減少學習一種程式設計語言的培訓成本、編寫程式的時間成本、偵錯程式的時間成本以及修改程式的維護成本。利用一種用戶良好的程式設計結構，希望在程式設計過程中出現較少的程式設計錯誤，從而減少偵錯所需的時間成本。

本發明的另一個目的是建立一種結構程式設計格式，該格式能夠清楚地表達該作業的骨架結構，並使該單個具有特定功能的程式模組以選定的最適於該功能的任何語言編寫。這種方法進一步提高了程式設計效率並減少了將來維護工作的工作量。

本發明的程式設計方法包括以程式資料填寫兩個以上表格的步驟。各種類型的表格被設計成用以執行不同的支援功能。一個表格被定義為一個資料矩陣。一個表格可以



五、發明說明 (5)

是一維或多維的。通常用m行和n列表示一個表格矩陣，其中m和n 是大於等於1的整數。可用標簽、運算式或一組運算式來表示表格的每個元素。

在本發明的第一實施例中，程式設計方法引入了任務表的概念，任務表包括一個用於控制多個任務的操作的資料表格。程式中可能存在一個以上的任務表並且每個表格由一種或多種任務狀態構成。在程式設計過程中無論何時指定一種任務狀態，都可以定義所指定的任務的狀況和/或操作。而且任務優先權資訊可包括在操作任務表或由該計算設備的另一個單獨表格以便在以計算設備的共用資源運行多個任務的同時分配處理優先權。

在本發明的基本方面，形成一個表格來定義包括輸入和/或輸出資訊在內的一種或多種配置狀態。當滿足了一種配置狀態的輸入限定條件時，將以另一個被稱為事件表或路徑表的表格中所表示的一系列特定操作作為回應。將至少一種配置狀態指定為有效狀態。用於微控制器的表格格式程式設計方法的最早版本在美國專利US 5,867,818中有所描述。本申請是指向多年的研究對基本概念的各種改進以擴大該技術的服務範圍。

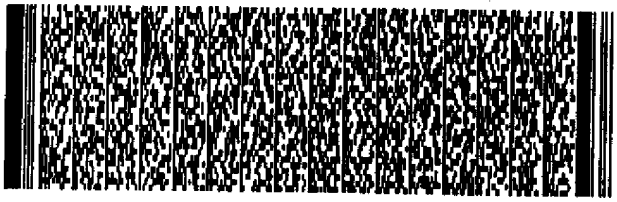
在本發明的另一個實施例中，形成另一個表格來定義觸發一種限定符配置狀態所需的限定條件。在本發明中，首次引入了虛擬限定的概念。一個虛擬限定被定義為表示來自一個物理硬體終端的任何限定條件。在考慮到由一個軟體指令所造成的結果或由整個軟體程式造成的結果時，

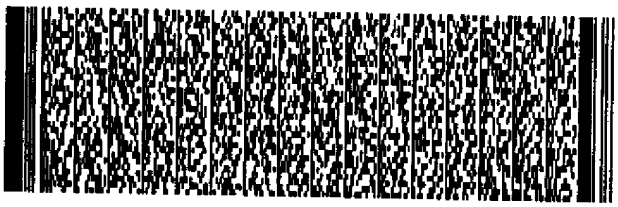
五、發明說明 (6)

虛擬限定是非常重要的。虛擬限定還涉及計算設備內部的硬體，如內部暫存器的溢出。其可利用一個內部軟體或硬體中斷的結果，一個標記的產生、極性信號、表示存在來自另一個系統的資料或存在一個標記的信號而被觸發。虛擬限定的另一個通用例子是滑鼠驅動程式的輸出，它可以指示一個指標隨滑鼠的移動而移動的方向。

在本發明的另一個實施例中，形成表格來定義一個輸出狀態的輸出配置。該輸出狀態可以是將利用物理硬體終端而被發送的信號或一種虛擬計算輸出狀態的配置。虛擬計算輸出狀態被定義成任何一種不描述硬體終端的操作的輸出狀態。虛擬計算輸出狀態的一個例子是觸發一個軟體程式開始運行，或設置用於控制軟體的特殊條件參數，或觸發軟體以操作某些特定功能。

當獲得合格的觸發時，執行一個相應的操作。路徑表或事件表是用於將一個或多個相應操作進行分組的表格。一個路徑可表示在接收到合格條件時所要執行的一個操作或一系列操作。一個路徑可被另一個路徑初始化。這個相應的操作或操作序列被稱為一個路徑方程(path equation)。狀態表與路徑表的互動式組合與狀態圖所示的事件結構相似。除此以外，表格格式程式設計還提供了一種能夠更好地反映出人類思維的程式表示方式，從而使其對用戶更為良好。為了更好地表達程式流的描述內容，可根據程式設計師的意願為路徑表中的每個路徑分配一個有意義的標號。一個有意義的命名標號可被分配給一個配





五、發明說明 (7)

置狀態或一個路徑。所有這些標號將幫助程式設計師或其他人員理解所編寫的程式的操作和邏輯流程。這是本發明的一個重要貢獻，有助於解釋編譯程序和減少將來的維護成本。

所發明的程式設計方法中所包含的另一個表格將輸出狀態下的輸出條件指向於一個特定的操作序列。可用另一個表格來表示這個序列以便簡化程式設計過程。

本發明中所引入的另一種類型的表格使程式設計師能夠將表格格式程式設計語言的運算式和/或語法變成所需的其他形式的運算式或符號。所包括的其他表格定義了一組或多組程式庫或外部程式以支援程式設計工作。

在表格格式程式設計方法的支援之下，傳統的程式設計過程被大大改變。當開始結構程式設計作業時，程式設計師用狀態和路徑表對程式流的骨架結構進行互動式的描述。程式設計師可自由地分配有意義的標號或名稱以便表示每種配置狀態和路徑。對每種狀態配置和路徑方程的有意義命名將有助於描述程式的流程。當包括在一個路徑中的操作難以用可得到的指令集表達或最好用另一種語言的程式模組來描述該操作時，程式設計師自由地分配一個有意義的標號來表達這個所需的操作。在程式設計過程的結尾，程式設計師應當為所分配的每個未定義標號提供可執行的程式模組。以這種方式，利用互動式地組織多組表格而形成了程式模組。一個具有至少一個狀態表和一個路徑表的程式模組被稱為一個程式組。可向每個程式組分配另

五、發明說明 (8)

一個有意義的標號以便描述程式模組的功能。然後在需要跳轉或功能調用時，這個標號可用於另一個程式模組或程式組以指向此程式組。為方便起見，將在程式開始時所執行的預設的第一個程式組稱為"main"程式組。在每個程式組中，應當具有一個"start"路徑，用作在初始化程式或預設啟動程式時的啟動路徑。為了改善程式設計效率，可用不同於主程序組中所用的表格格式程式設計方法的語言來構成支援程式模組。在同一程式中可使用多種不同的語言。這就非常希望按在表格格式程式設計方法中具有能夠在不同語言的程式模組之間傳遞參數和變數為特徵設計。在一個最佳實施例中，表格格式程式被設計成與不同程式設計語言的介面並被用作在以不同語言編寫的程式模組之間進行通信的橋梁。可以說表格格式程式設計語言大大改變了編寫程式的概念和習慣。在一種富有邏輯和較好結構的方式下編寫一個程式。利用自由地分配有意義的標號，程式流程被儘量平滑地組合而成以便描述該程式設計作業。然後每個標號與一個用任何語言編寫的外部程式模組相鏈結。有時，可排列成多級程式組或模組。由於每個表格格式程式組的程式長度通常非常短且該程式具有合理的結構，從而能夠容易地實現表格格式程式設計的上述優越性。

對於表格格式程式設計來說，自由分配或使標號與指令關鍵字等效對於支援多語言程式設計平臺是非常重要的。這是由於多種語言可對同一類型的應用程式使用不同

五、發明說明 (9)

的語法。標號的自由分配和等效特徵使用戶能夠根據他/她的意願而統一標號和語法。

表格格式程式設計的特性使其成為一種用以增強另一種語言的程式設計結構的極佳支援工具。例如，一種高階語言的編譯程序可被修改成包括僅足以提供以該語言編寫而成的一個程式的骨架結構的表格格式程式設計函數。

由於表格格式程式設計方法是唯一和獨立於其協同的第三種語言程式的，所以利用只提供一個主要用於處理所有表格格式程式設計任務的簡單表格格式輔助運算器就能夠簡化程式管理作業。這種多處理器結構將減輕主處理器的工作量並使其集中處理常規作業和提高尤其在多工運行環境中的整體系統性能。表格格式輔助運算器可以是一個位於主處理器外部或與主處理器共存於同一積體電路晶片上的處理器。當表格格式程式設計應用於微處理器或微控制器時，經過編譯或編碼的表格格式程式變成了存儲在諸如ROM、EPROM或快閃記憶體之類的記憶體中的用於微處理器或微控制器運行的數位資料。然後一個由處理器和存儲表格格式程式的編譯形式的記憶體構成的印製電路部件被用於製造可出售的商業成品。

根據表格格式程式設計的結構組織，它特別適用於事件驅動的應用程式，例如在windows環境下的程式設計，網站，互動式遊戲或控制程式設計。用於事件驅動程式設計中的表格格式的一個重要特徵是用高度壓縮的格式來表示一個程式。大多數用於小作業的表格格式程式的長度都

五、發明說明 (10)

少於一頁。證明了表格格式程式設計中的一頁可表示八頁組合語言程式的實驗性結論是根據可用指令集的複雜性而作出的。與很多高階語言相比，還可在相當大的程度上節省代碼的長度。這種特性使其在用作利用一個有限帶寬的通信通道進行通信的媒體時是相當經濟的；在例如與網路或因網際網路應用中的多個本地電腦進行通信的主系統之間。應當注意一個網路，通信連接或通信通道是指將兩個計算設備連接在一起的任何裝置，包括序列埠、平行埠、USB埠、因網際網路、內聯網、外聯網、LAN和任何使兩個計算設備介面的通信裝置。兩端的設備可利用有線、無線或混合連接模式進行連接。在因網際網路程式設計中，在本地電腦執行一個下載程式之前，還需要一個估計用戶電腦的系統配置，然後調整程式設置的步驟。電腦中很多與人介面的設備配置，例如監視器、圖形卡、聲音發生設備、指標控制、遊戲控制器控制等都屬於將被定義的配置設置。由於因網際網路通信線通常是一個系統的瓶頸，所以建議在本地電腦上執行表格格式程式設計的編譯作業，以便利用表格格式程式設計的壓縮代碼的優點。

圖11是表示一個本地電腦801如何與一個遠端電腦803相連接的方框圖。在很多應用中，一個表格格式程式被存儲在本地電腦中。根據請求，利用一個通信連接802將該程式下載到遠端電腦中。該表格格式程式可存儲在遠端電腦的存儲裝置804中或立刻由遠端電腦進行編譯。編譯文件是一個用以執行遠端電腦的某些預定工作的可執行文

件。這個可執行文件也可存儲在遠端電腦的存儲裝置804中。

當一個編譯器被設計成用於處理表格格式程式設計時，著重建議使用關鍵字來識別每個表格以及相應程式組的狀態和位置。以任務、程式組、限定符、狀態、路徑和程式庫為示例關鍵字用於識別所示實施例中的功能表格。應當注意表格格式程式設計方法的編譯，翻譯，解釋或轉換包括將表格格式程序變換成其他程式格式，例如機器語言或任一種更高階語言的進程。該變換進程可由用另一種程式設計語言編寫或被一個可執行表格格式程式支援的編譯程序執行。

任務表的概念有效地提高了在多工運行環境下進行表格格式程式設計的便利性。本發明的特徵如後面的申請專利範圍所述。對計算設備的限制是指具有計算能力的任何設備，包括電腦、微控制器、微處理器、由微控制器或微處理器構成的印製電路部件。除電腦以外，用以支援本發明技術的其他支援硬體包括偵錯硬體、諸如電纜、通信口、網路集線器之類的通信連接以及特定的網路。表格格式程式可被顯示在顯示終端和印刷品上，並可被編碼為數位資料。表示該表格格式程式的編碼數位資料被存儲在諸如RAM、ROM、磁碟機和CD ROM之類的任何存儲設備中。實施例中所用的技術術語、關鍵字和標號只是作為例子，它可具有各種變形和修改，並且能夠容易地預見到表格格式的重新排列可達到相同的效果，所有這些特徵都包含在後

五、發明說明 (12)

述的申請專利範圍範圍內。結合附圖的詳細說明更有利於本發明的理解。

為了充分利用本發明技術的有益之處，用戶需要為電腦預處理操作進行精心的準備。典型的電腦預處理操作包括分析程式作業說明，將方案簡化為表格格式的相關狀態圖。在提高程式結構清晰度的同時引入表格格式程式組需要用戶在開始應用程式設計技術之前清楚地識別和定義方案的具體功能模組。由於表格格式程式設計方法的優點，另一個介面說明過程需要確證表格格式程式組或模組的介面關係以便為一組程式設計師分配程式設計作業。在網路通信或下載應用中，電腦預處理操作涉及利用一種通信連接或網路向一個遠端電腦發送表格格式程式。

利用表格格式編譯器執行處理中的電腦操作，該編譯器將表格格式程式翻譯成本地電腦、目標微控制器或遠端電腦可執行的代碼。後一電腦操作通常是目標電腦或微控制器可執行的代碼。這個可執行代碼還可被電腦或微控制器運行以便根據原始的程式設計說明來執行功能。編譯後的可執行代碼通常存儲在諸如RAM、ROM、任何可程式設計非易失性記憶體之類的存儲裝置或任何其他商業可用的存儲設備中。在微控制器作為消費品的情況下，用於存儲編譯後的可執行文件的存儲裝置通常位於出售品中而不是位於編譯電腦中。在這種情況下，編譯電腦只是作為一個開發系統或遠端計算設備的程式供應方。

五、發明說明 (13)

實施例：

在下面整個詳細描述中，所有附圖中的相同附圖標記表示相同的元素。

首先參照圖10中所示的由美國專利US 5,867,818公開的基本表格格式程式設計。該表格格式程式由用以對聲音發生可程式設計控制器程式設計的兩部分構成：狀態表和路徑表。在狀態表的第二行中，定義了控制器的相應輸入觸發引腳的順序。狀態0到狀態4中的每一個定義了控制器的一種可能觸發狀態。例如，R：Path1元素指向於狀態0的TG1，表示如果檢測到一個上升沿(用"R"表示)，則執行命名Path1的路徑。在Path1中，有效狀態改變到狀態1，之後產生一個名為"Sound1"的聲音。在發出該聲音之後，控制返回Path1並開始另一個迴圈的聲音發生序列，直到在State1狀態中由TG1接收到一個下降沿觸發(用F：Path11表示)為止。這個程式例表示TG1到TG4的"level hold"功能。即，當壓下TG1到TG4中的一個時，將產生一個聲音。聲音將是迴圈發生的，直到放開觸發按鈕為止。

下面參照圖1，圖1表示一個改進後的表格格式程式實施例中所選定的一個表格的結構。關鍵字101表示程式的開始。程式設計師在關鍵字下面的位置上指定了程式名。關鍵字106表示一個列出了所包括的程式模組的表格的開始部分。關鍵字107表示一個定義了該程式中所用常量的表格的開始部分。關鍵字102對程式中所用的變數進行說明。關鍵字103是一個列出了用戶定義的等效命令和語法

五、發明說明 (14)

運算式的表格。為了使程式關鍵字是可區分的，程式設計師可利用關鍵字104在表格的開始部分定義關鍵字的列印格式。上述所有特徵是為表格格式程式做準備工作。任務表105提供了一種或多種任務狀態，用以定義哪個任務是有效的、暫停的或終止的。程式組111由一個可選的限定表112，至少一個狀態表113和一個路徑表116組成。實際的程式作業由狀態表和路徑表的內容交互構成。可附加輸出狀態表115和輸出方程表以便進一步定義程式的輸出狀態。最後，附加一個程式庫表121以便提供通用的命令串和副程式。應當注意上述所有表格無須按順序排列並且很多表格的建立只是為了提供可選的特徵。另外，一個程式可包含多個相同類型的表格，如任務表和狀態表的情況。可在關鍵字冒號之後隨意指定表格的名稱以便更好地體現該程式的含義。表格格式程式設計中所用的關鍵字可多於一個字並且可以用指令、具體的變數、常量和系統硬體來表示。應當注意所提供的關鍵字只是舉例說明，還可使用其他的關鍵字名稱。除此以外，對表格的範圍進行合理的改進是可能的，應將其視為本申請的保護範圍之內。

現在參照圖2A，表格100代表圖1中的表格103的具體例子。關鍵字201" 定制運算式(Custom Expression)" 表示這個功能表的開始部分，該表列出了等效於正規表格格式運算式的用戶定義運算式。例如，假設邏輯AND功能的正規表格格式運算式是203所指的"AND"；一個習慣用C語言程式設計的程式設計師可隨意地用202所示的C語言指令集

五、發明說明 (15)

中的"&&"來代替表格格式命令"AND"。定制運算式適用於語言或系統設置的任何字或符號，例如指令命令、符號和系統關鍵字。建議每次在定義一個替換運算式時提供一個注釋204。這個功能的優點在於為程式設計師提供個人化的支援，以便總可以使用慣用的符號或運算式。但是，當列印所編寫的程式時，編譯器或編輯器最好以正規運算式列印出程式清單以便於其他人員閱讀。對程式編輯器亦是如此。只要鍵入用戶定義的運算式，將可以顯示出正規的運算式。還建議在預定的正規或用戶定義的運算式之間提供一種轉換功能，使用戶可以選擇是以正式的形式還是以定制的形式顯示或列印程式。利用這種用戶定制運算式的特徵，在保持程式設計師使用縮寫的運算式或較短的符號來表示指令命令和語法的同時，還可利用有意義的長運算式名稱使程式清單對其他人員而言更具有可讀性。利用公知的查表方法，在表格格式程式中建立正式的語法組和相應的用戶定義的等效語法和標號組。一個典型的應用實例如圖9所示，圖中列出了一些較長的描述性指令集。移位操作符"BIT SHIFT LEFT"和"BIT SHIFT RIGHT"清楚地描述了所要執行的操作。但是，這些指令對於有經驗的程式設計師來說太長和不受歡迎。程式設計師將"BIT SHIFT LEFT"指令等效於C語言中的簡潔而描述性較差的"<<"指令。長的指令名使使程式易於理解但對有經驗的程式設計師來說是無用的。"定制運算式"表在保持長指令運算式的優點的同時有效地解決了這個問題。

五、發明說明（16）

由於QWERT鍵盤上可用的符號是非常有限的，因此從鍵盤上很難發現足夠的有意義符號來組成一種新的象表格格式程式設計方法那樣與其他某些通用記號和符號的傳統用法不相矛盾的程式設計方法。利用使用戶能夠按照自己的意願重構指令和符號，定制運算式表可作為一種解決此問題的方式。

元件210表示如何將一個名稱"My Sign"分配給"客戶運算式"表。當為了使一個以上的用戶操作或閱讀程式而提供一組以上的"客戶運算式"表時，特別需要一個名稱。例如，除了包括在程式中的表"My Sign"以外，另一個名為"Johns Sign"的定製表可被加到同一個程式中。如果John想要閱讀該程式，只需將"Johns Sign"運算式表設為預設顯示表，John 將可以以他所喜歡的格式來顯示該程式。此表的新特性使每個用戶都包括他們自己的定制運算式設置，以便於按照他們喜歡的格式來轉換或編輯程式。

儘管圖2A所示的表格提供了一種重構關鍵字和運算式的方法，但需要一種為一個小作業提供另一種語言的替換運算式或命令的簡單方法。這是可以利用在用該語言規定的運算式前指定一個表示該語言的符號來實現的。圖2B表示如何用二進位數字00001111遮罩(mask)暫存器A的內容以便於得到暫存器A最後四位元的內容，然後進一步顯示這個數位。運算式216以元件"(C：&&)"來表示"&&"指令是一個"C"語言指令。然後利用一個預定義的"Display"命令來顯示暫存器A的遮罩值。可替換地，利用一個微處理器

五、發明說明 (17)

的組合語言中的"&"指令可得到執行同一功能的運算式。標記"A：()"是一個表示括弧中的操作是以組合語言編寫而成的運算式。儘管這兩種方法都很方便，但是仍沒有圖2A中所示的能夠使一個全程式的個人化表示式是被轉換或個人化的定制運算式表功能強大。

由於表格格式程式設計中涉及用戶指定的大量的分立標號，並且這些標號分散在程式中並與關鍵字和指令命令相混合，因而對於一個閱讀該程式的用戶來說，難以從其他關鍵字和指令命令中識別出這些標號。因此最好提供用以識別這些標號的裝置以便使該程式對用戶更為良好。圖3表示一個控制如何表現用戶分配標號的程式表。表元件234提供了可選擇的情況。可用於選擇的典型情況包括首字母、所有大寫字母和所有小寫字母。表元件239表示可選擇的字母類型。典型的可選擇類型包括黑體、斜體和下劃線。應當注意元件234和239都為包括黑白印表機在內的各種顯示設備提供了卓越的識別功能。關鍵字231表示用於定義程式部分的識別類型的開始部分。建議在關鍵字231後面設置一個由用戶分配的名稱以便表示下面的設置最適用於某個特定的人員。在程式中可包括多個依據個人喜好而分配的識別類型表，並且可為每個表分配一個用元件232表示的名稱。選擇其中一個識別類型表可將列印格式設置成適用於正在閱讀該程式的特定用戶。

為了表示在多工運行環境下具有較好結構的程式流，在圖4所示的實施例中引入了一個任務控制表(此後稱為任

五、發明說明 (18)

務表）。元件261是用於識別一個任務表的關鍵字。元件262是一個對任務表命名的標號。提供這個名稱是為了在需要兩個或多個任務表時與其他任務表相區別。元件263、264表示可在任務表的控制下運行的不同任務或程式。任務表的每一行表示一個任務狀態。在每種任務狀態下，指定一種任務條件以表示每種任務的運行條件。下面列出了用以描述一種任務條件的運算式的幾個例子：

Start：表示不論該任務或程式是正在運行、中止或已經結束，都從頭開始重新運行該程式；

Continue：表示如果一個任務已經開始運行，則繼續運行之；

Pause：表示任務或程式被置於中止狀態；

Run：表示如果任務還沒有開始運行，則開始運行之；如果任務或程式正在運行，則繼續運行之；如果任務被中止，則恢復該任務的運行；

X：表示結束該任務的運行。

在行265，任務的名稱為"Task Status 1"，它指示"Main"程式開始運行而程式2到n都處於結束狀態。在任務狀態268，指示所有的程式運行。應當注意對每個任務表來說，在任何時刻只有一個任務狀態被指定為有效。在具有有限資源的系統中，向正在運行的有效任務指定優先權是非常重要的。任務狀態273和274為任務們分配優先權。應當注意可以建立一個獨立的表格來描述分配給任務的優先權。由於有效任務表和優先權任務表中每一元件的開頭

五、發明說明 (19)

元件是相同的，所以可以將兩種類型的任務表組合成一個任務表，如圖4所示。在這種情況下，就需要兩種有效任務狀態，一種用於任務有效狀態，而另一種用於任務優先權分配。

圖5A表示一個介紹任務表概念下的實際應用例子。圖5B是圖5A的標號表示。這個例子由三個任務表組成。第一任務表名為"Input"，如元件301所示。它由三個分別名為"Keyboard"、"Mouse"和"Game port"的程式組成。"Keyboard"是一個用於掃描鍵盤上的按鍵的程式。"Mouse"是一個對滑鼠的移動進行解碼的程式，而"Game port"則輸入來自博弈埠的觸發信號。在名為"All"的任務狀態305中，運行三個程式以使計算設備對三個輸入設備都有回應。在名為"Normal"的任務狀態306中，則只認可鍵盤和滑鼠。不使用博弈埠是為了提高計算設備的服務效率。在處於遊戲模式時，名為"Game"的任務狀態309變成唯一的有效程式或任務。不使用鍵盤和滑鼠是為了使計算設備將所有資源集中在遊戲上。第二任務表是一個名為"Ports"用於控制計算設備的串列和平行埠的表格。名為"Device"的第三任務表如元件341所示。它控制驅動器程式操作"CDRom"342、"HardDriveC"343和軟碟驅動器。當任務狀態"ReadCD"345被啟動時，CD Rom和Hard Drive C驅動器程式被啟動而軟碟驅動器程式被終止。在需要硬碟驅動器全速運行的模式中，任務狀態"HDFullSpeed"346變成有效任務狀態而硬碟驅動器變成唯一正在運行的設備。

五、發明說明 (20)

根據這個應用實例，建議只對具有相似內容或相互關聯的任務進行分組以形成一個共用的任務表。應當注意在任何時候，每個任務表中只有一個任務狀態被指定為有效。

參照圖6A，表示表格格式程式的一個主程序組。該程式名為"Web Sale"，提供了一個利用因網路提供的銷售程式的骨架結構。這個例子表示在表格格式程式設計環境下進行多種語言程式設計的概念。為了便於描述該實施例而在程式中插入了行號。應當注意狀態和路徑方程不必是按照順序排列的。現在參照第1行。關鍵字"Group"表示一個表格格式組或程式模組的開始部分。該組的名稱為"Main"。將Main作為一個關鍵字用以表示這個組是在開始運行該程式時所要執行的第一程式組。第2行以一個關鍵字"qualifier"開始，它定義了在配置狀態中所列出的限定符的限定條件。在第3到6行，將術語"Icon"用作一個功能性命令，用於構造一個圖示並在單擊該圖示時觸發該配置狀態。在典型的表格格式程式設計中，按順序是數形成大量圖示。在限定表中定義和命名每個圖示。例如，當Icon(1)指向於名稱"Catalog"時，字"Catalog"被指定給第一圖示並顯示於其上。實際上，一旦將一個名稱指定給一個圖示，其編號就是不重要的，除非在一個程式中提到了術語"icon(n)"並且"n"是一個計算結果。第7行定義了一個名為"First Page"的輸入狀態配置表。在這個表中指定了五種限定符，即"Catalog"、"Purchase"、"Service"、"Home"和"Quit"。每種限定符系指一個圖示

五、發明說明 (21)

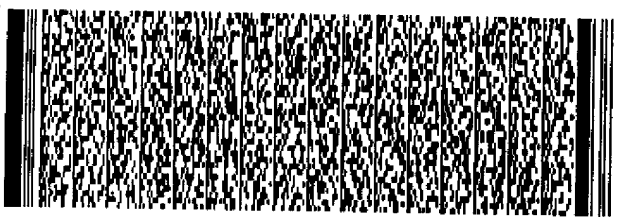
的觸發，如第2行的限定表中所定義的。第一輸入限定符狀態名為"Ready"，如第8行所示。在這種狀態下，當接收到一個代表圖示"Catalog"的限定觸發時，執行名為"P_catalog"的路徑，對其他限定符來說亦是如此。第9行表示另一個名為"Hold1"的輸入配置狀態。狀態方程中的"x"表示相應的限定條件是無關條件，在該條件下當出現限定觸發時，該觸發被阻塞或不需要回應。

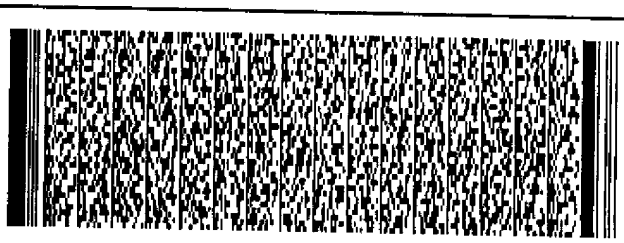
下面參照提供了另一個名為"Response"的配置表的圖6A中的第11行。這是一個具有五個元件的輸出狀態配置表。前四個元件具有一個關鍵字"Group:"，表示構成了一個程式組的程式。冒號後面是程式組的名稱。第一組名為"Info"，用於提供產品資訊。第二組"Order"是一個指導用戶利用諸如登記信用卡號、產品號、定單數量、總量、選擇之類的購買過程，對資料加密並將定單發送到供應商進行解碼的程式。第三組"Service"提供了通用的互動式客戶服務條件。第四組"Register"登記客戶資訊。輸出配置的最後一個元件是一個與計算設備的揚聲器相連接的硬體埠P3.1。當為這個埠分配一個代碼P+時，一個正向脈衝串被發送到揚聲器並聽到一個通知音。埠P3.1是一個硬體終端，因此將其分類為一個指向硬體的輸出。前四組都是指向軟體的輸出條件，因而被分類為虛擬計算輸出。任何與一個硬體輸出無關的輸出條件被定義為一個虛擬輸出。虛擬計算輸出的含義包括任何用以產生資料的與終端無關的操作，信號或資訊的顯示或產生，初始化程式，重

五、發明說明 (22)

起動程式，啟動軟體計時器，或計數器或操作一個內部電路，如暫存器等。第12到17行是在狀態表"Response"下配置的輸出狀態。當一個"Run"命令出現在一個輸出配置狀態中時，運行相應的組程式。當接收到一個"Continue"指令時，繼續運行正在運行的程式或者如果該程式還沒有開始運行或處於暫停狀態，則該程式保持空閒狀態。"x"標記表示不需要輸出操作。利用這些描述，第12到17行的輸出狀態的操作是一目了然的。應當注意在一個程式中可能存在一個以上的輸入或輸出狀態表。多個狀態表簡化了標格的結構並使程式設計作業更為容易。但是，應當注意在任何時候，每種輸入狀態表中只有一種配置狀態被指定為有效。作為一種程式設計技巧，相互聯繫的輸入限定符和輸出條件可被組合成一個狀態表。還應當注意，如果需要，可將輸入狀態和輸出狀態組合成一種混合狀態。

第18行為動操作"Path(s)"。當路徑名被一種配置狀態的任何限定符元件所引用時，每個路徑定義一個或多個要執行的操作。第19行是一個名為"Start"的路徑，在執行Group時，該路徑是一個預設起始路徑。當首先起動該程式時，從所引用的所需操作中開始程式設計過程。開始操作"Check System"檢查輸入顯示驅動器、物理埠之類的本地計算系統的配置以得到運行程式可用的揚聲器和系統資源。要估計的本地計算設備可用資源包括計算機時間、暫存器個數、可用記憶體的存儲量、記憶體配置、所佔用的計時器和計數器、可用的中斷通道以及任何專用硬體電



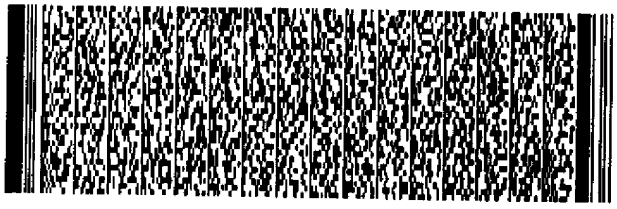


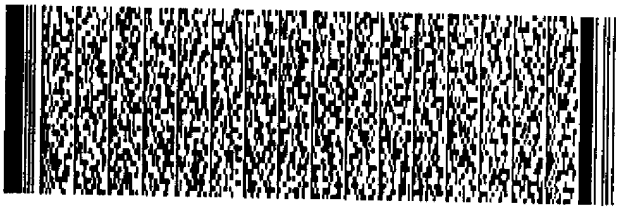
五、發明說明 (23)

路配置。"Check System"操作中應包括一個根據系統參數重構下載程式的過程。下一個步驟是顯示第一頁。在程式中僅將此操作定義為"Display First Page"。然後產生一"滴滴"聲。"Hold2"表示所有的輸出配置置於保持狀態，如"Response"狀態表所示。"Ready"指令始於輸入狀態表"First Page"的"Ready"狀態。在"Ready"狀態中，無論何時接收到圖示"Catalog"、"Purchase"或"Service"的一個限定觸發，都執行相應路徑20到22中的一個。在上述每個路徑中，顯示指示該操作的視窗並啟動促銷程式。"Buy Solicit"是一個用於請求銷售公司產品的互動式程式。在路徑"P_purchase"中，操作"Hold1"和"Hold3"限制除圖示"Home"和"Quit"之外的來自本地用戶終端的可允許回應。"Grey Button"是一個用以改變圖標顏色的操作，它不適用於例如由狀態命令"Hold1"所指定的圖示"Catalog"、"Purchase"和"Service"。當執行第23行的路徑"Bye"時，結束程式"Terminate"和該程式。一個意味著"程式組結束"的關鍵字"EOG"位於程式組的結尾，用於通知編譯器該程式組到此結束。

應當看到所討論的程式設計方法根據組成狀態和路徑而互動式地描述了程式操作。使用了由程式設計師分配的有意義的術語，如"Beep"、"Check System"和"Terminate"。這個過程就象編寫一篇用以精確描述程式所需操作的論文一樣自然。

在此程式的編譯過程中，很多程式設計師分配的術語





五、發明說明 (24)

仍然是未標識的，例如"Check System"、"Display First Page"和"Buy Solicit"。這些程式設計師分配的標號都是計算設備不可執行的，除非它們與一個可執行程式鏈結。圖6B中所示的下一個步驟就是需要進一步定義未標識標號的描述。使這些標號可執行的典型方法是將它們與一個外部可執行程式相鏈結或以一個來自程式庫的程式來定義該標號。這就非常需要編譯器提供一種能夠識別所有程式設計師分配標號的功能。該標識最好在利用一個黑白印表機列印該程式時是可識別的。典型的最佳識別方法包括改變字母字格式和字型，如黑體、斜體或下劃線。然後分析每種未標識標號的要求並選擇一種最適宜的程式設計語言來編寫一個程式以便提供所需的操作。

可以用任何語言，甚至另一個表格格式程式來編寫支援程式。這些支援程式被"包括"在用於編譯器的程式中以使所有程式集中在一起。圖6B的第18行表示"Check System"最好是一個用Java語言編寫的字首為"EJ"的程式，其中"E"表示它是一個被包括在內的外部程式。操作"Display First Page"最好用Visual Basic語言編寫。在第19行中，操作"Buy Solicit"，一個徵求購買的視窗最好來自一個局部或總程式庫。第23行表示局部程式庫的開始部分。第24行是一個精心設計的路徑方程，它描述了徵求客戶的操作。這個操作包括執行一個用"C"編寫的外部程式"Check Record"和一個用Visual C++編寫的程式"Solicit Window"，以便利用人機對話的形式徵求購買產

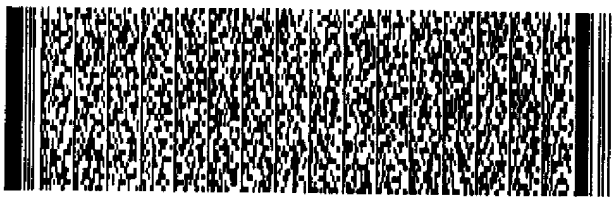
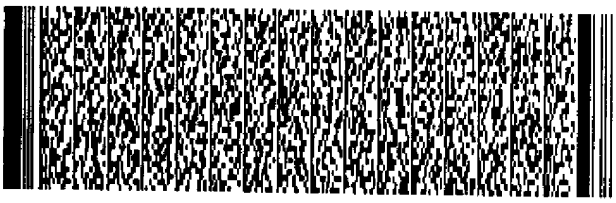
品的客戶。第20行中的元件"Grey Button"是指一個位於局部程式庫中的路徑，它由一個用於識別哪個圖示具有"x"標記的"C"程式和另一個用Visual Basic編寫的程式用於將這些圖標變成灰暗顏色以表示這些圖示不能被觸發。

當建立了一個具有大量支援程式的程式庫時，一個熟悉表格格式程式設計的程式設計師可利用選擇和引用各種通用支援程式來開始程式設計作業。圖7表示構成一個包括視窗和收發器的程式時的引用的表示例。在這個引用表中的支援程式是必須遵循的，並因此需要編譯器排除不用於所要構成的程式中的任何引用程式。

利用表格格式程式設計方法來管理用其他語言編寫而成的支援程式需要更高的技術要求，例如在程式之間傳送參數和使變數相等的方法。如果不同類型的程式是由不同的編譯器翻譯的，則要特別考慮對這些程式的正常管理。所屬技術領域的技術人員能夠理解所公開的表格格式程式設計方法的優點並建立一個編譯系統來完成所需的操作。例如，在引用一個特殊外部程式時，可分配預先定義的暫存器或記憶體塊來處理參數的傳遞。

從上述實施例中總結出表格格式程式設計的主要優點如下：

1. 利用編寫描述性標號可比較容易地編寫程式。
2. 程式的結構較好以致出現故障的機會很小。
3. 對於硬體終端來說，簡化了與虛擬軟體輸出的組合和對高級程式流的影響。

五、發明說明（26）

4. 表格格式程式提供了清楚簡潔的表達方式並易於為第三者讀懂。它所提供的良好而清楚的表達對於進一步減少程式偵錯時間和縮減維護成本是非常重要的。

5. 表格格式程式設計方法使利用多種語言來構造一個程式變得更加簡單。根據應用環境和每種語言的特點來選擇語言。

6. 用表格格式編寫的簡明程式提供了較高的資料壓縮率並使利用具有有限帶寬和資料處理效率的通信通道而從一個遠端的主機終端到一個本地計算設備的傳輸更加理想。

上述本發明的最佳實施例只是舉出的例子，可以預想到各種修改、規定的改變、表格的重排、指令和關鍵字的分配都能達到相同的效果，從而這些修改和改進都應包括在後述的申請專利範圍範圍內。

圖面之說明：

圖1表示表格格式程式的一個實施例的結構。

圖2A是用於定義等效於用戶定義運算式的表格格式運算式的表格。

圖2B表示如何將一個名稱分配給圖2A所示的表格。

圖3是用於定義將被識別的字的不同列印格式的表格。

圖4是表示一個任務表的不同任務狀態的實施例。

圖5A是表示一個程式中的多個任務表的實施例。

五、發明說明 (27)

圖5B是圖5A的數位化表示。

圖6A是表示一個表格格式組的一個實施例的程式。

圖6B是圖6A的實施例的另一種開發形態。

圖7是表示所包括的文件列表的表格。

圖8是圖7所示的表格的另一種形態。

圖9表示用於表格格式程式設計中的一些建議長度和描述性的指令命令。

圖10表示現有技術中用於聲音發生微控制器中的基本表格格式程式。

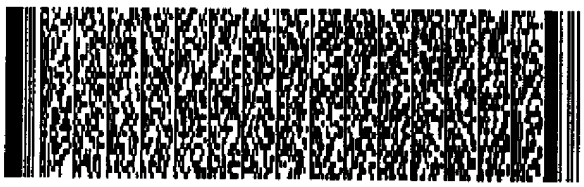
圖11是表示一個本地電腦利用一種通信連接將表格格式程式下載到遠端電腦中的方框圖。

四、中文發明摘要　(發明之名稱：表格格式程式設計)

本發明適用於利用多種語言並利用因網際網路和網路環境下載的表格格式程式設計方法提供了一個用於定義定制運算式的可選表格(103、200)，一個用於規定多工應用的可選表格(105、260)，一個用於規定標號的列印字格式的可選表格(104、230)以及一個與路徑表(116)交互作用的配置狀態表(113)。可選限定表(112)定義了一種輸入配置狀態的限定條件。硬體實施包括一個用於處理表格格式程式設計過程並釋放主處理器用於其他應用的輔助運算器。在程式設計過程中，程式設計師提供了用戶定制標號來描述所需的功能和操作。這些標號還可用於其他程式組中開發。

英文發明摘要　(發明之名稱：Table Format Programming)

Table format programming method suitable for use with multiple languages and download through internet and network environments provides an optional table (103,200) to define custom expressions, an optional table (105,260) to specify multitask application, an optional table (104,230) to specify printing style of labels and a table of configuration states (113) which interact with a table of paths (116). An optional qualifier table (112) defines qualifying



四、中文發明摘要　（發明之名稱：表格格式程式設計）

英文發明摘要　（發明之名稱：Table Format Programming）

conditions of an input configuration state. Hardware embodiments includes a coprocessor to handle the table format programming process and relieve the main processor for other application. During programming, programmers provide user defined labels to describe the function and action desired. The labels are then further developed in additional groups of program.

六、申請專利範圍

1. 一種由計算設備執行的電腦程式，包括具有x種配置狀態的第一表格，所述配置狀態中的至少一種定義了一個或多個限定條件；

規定了y個路徑的第二表格，當滿足所述第一表格中列出的一個限定條件時，執行所述路徑中的至少一個；所述程式還包括下述表格中的至少一個：

（1）一個規定用於表示一種程式設計語言的相應預定義指令的用戶定制運算式的表格；

（2）一個定義關鍵字的可取字格式的表格；

（3）一個用m種任務狀態定義n個任務的有效性的表格，所述任務狀態中的至少一種規定了k個有效的選定任務；

（4）一個用p種任務狀態定義q個任務的表格，所述任務狀態中的至少一個規定了將被服務的有效任務的優先權；

（5）一個定義x個限定運算式的表格，每個限定運算式表示一種配置狀態的限定條件；以及

（6）一個定義y個輸出運算式的表格，每個輸出運算式表示一種配置狀態或所述第二表格中列出的一個路徑元素的輸出條件。

2. 如申請專利範圍第1項所述之電腦程式，其中用存儲在用於存儲數位資料的存儲裝置中資料來表示。

3. 如申請專利範圍第2項所述之電腦程式，其中該程式被嵌入用於銷售商品的計算設備中。

六、申請專利範圍

4. 一種用於對回應一個或多個限定條件執行一個或多個路徑的計算設備進行程式設計的程式設計方法；所述程式設計方法最少包括步驟：

（1）規定x種配置狀態；

（2）規定一個或多個限定條件到至少一種配置狀態，

（3）規定y個由所述計算設備執行的路徑；

（4）分配z個標號來表示步驟(3)中的一個或多個路徑元素，其中z是大於等於1的整數；

（5）對於步驟(2)中的每個限定條件，當滿足一個特定限定條件時，就規定步驟(3)中的一個將被計算設備執行的路徑；

（6）將配置狀態中的至少一種規定為有效配置狀態；

（7）為步驟(4)中的每個分配標號，定義一個用所述標號表示的可執行程式。

5. 如申請專利範圍第4項所述之程式設計方法，其中步驟(4)中的標號最初不能被所述計算設備執行並且另一步驟(7)被配置成能夠被所述計算設備執行所述標號。

6. 如申請專利範圍第5項所述之程式設計方法，其中還包括識別不能被所述計算設備執行的所述標號以用於構成步驟(7)的程式的步驟。

7. 如申請專利範圍第4項所述之程式設計方法，其中用包括表格格式程式設計語言在內的任何可用程式設計語言來構成步驟(7)的程式。

8. 如申請專利範圍第7項所述之程式設計方法，還包括定

義用於在步驟(1)到(6)所表示的程式和標號所表示的可執行程式之間傳遞參數 的裝置的步驟。

9. 如申請專利範圍第4項所述之程式設計方法，還包括規定指向於所述x種配置狀態中的至少一種的一個或多個輸出條件的步驟。

10. 如申請專利範圍第4項所述之程式設計方法，其中在一個路徑或一種配置狀態中規定了至少一個輸出條件。

11. 如申請專利範圍第4項所述之程式設計方法，還包括識別所述計算設備管理其狀態和路徑所需的資源的步驟。

12. 如申請專利範圍第4項所述之程式設計方法，還包括指示所述計算設備用於構成步驟(7)的程式的資源的步驟。

13. 如申請專利範圍第4項所述之程式設計方法，其中所述計算設備由兩個或多個處理器構成；利用第一處理器提供管理特定狀態和路徑的資源並利用第二處理器資源運行步驟(7)的程式的至少一部分。

14. 如申請專利範圍第4項所述之程式設計方法，其中所述計算設備被定義為第一計算設備；所述程式設計方法還包括利用通信連接從與所述第一計算設備距離遙遠的第二計算設備接收表示上述步驟的數位資料的步驟。

15. 如申請專利範圍第4項所述之程式設計方法，其中至少部分所述步驟按照表格格式進行組織。

16. 如申請專利範圍第15項所述之程式設計方法，還包括將步驟(1)和(2)的配置狀態分成一個或多個表格的步驟。

17. 如申請專利範圍第15項所述之程式設計方法，其中配

六、申請專利範圍

置狀態或路徑不必按照順序彼此依次列出。

18. 如申請專利範圍第4項所述之程式設計方法，還包括將配置狀態和路徑說明轉換成存在所述計算設備中的數位資料以便用於執行的步驟。

19. 如申請專利範圍第4項所述之程式設計方法，還包括用於轉換利用不同格式的第二程式設計語言的所述步驟的至少部分說明的步驟。

20. 如申請專利範圍第4項所述之程式設計方法，還包括用於識別由所述步驟構成的程式位置的步驟。

21. 如申請專利範圍第7項所述之程式設計方法，其中步驟(7)的程式是一個位於由申請專利範圍1中的步驟構成的程式之外的程式。

22. 如申請專利範圍第21項所述之程式設計方法，還包括用於識別所述外部程式位置的步驟。

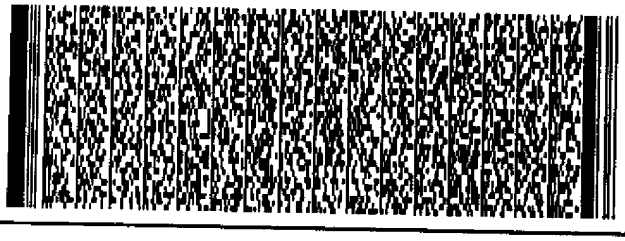
23. 如申請專利範圍第4項所述之程式設計方法，還包括用於形成一個獨立的用於規定由步驟(1)到(7)構成的程式是有效還是無效的表格的步驟。

24. 如申請專利範圍第4項所述之程式設計方法，還包括用於分配一個表示一種配置狀態的標號的步驟。

25. 如申請專利範圍第4項所述之程式設計方法，還包括用於分配一個表示一個路徑的標號的步驟。

26. 一種用於對執行多個程式的計算設備進行程式設計的程式設計方法，包括步驟：

(1)規定n個可由所述計算設備執行的程式；



六、申請專利範圍

(2)定義m種任務狀態，每種任務狀態規定k個有效的選定程式，其中k是大於等於0的整數；以及

(3)規定步驟(2)中的一種任務狀態為有效任務狀態。

27. 如申請專利範圍第26項所述之程式設計方法，其中至少一個程式是一個表格格式程式，包括：

x種程式配置狀態，所述配置狀態中的至少一種定義了一個或多個限定條件；

y個由所述計算設備執行的路徑；以及

當滿足一個特定的限定條件時，所述計算設備執行所述路徑中的一個。

28. 如申請專利範圍第26項所述之程式設計方法，其中步驟(1)的程式是用不同的語言編寫而成。

29. 如申請專利範圍第26項所述之程式設計方法，還包括用於提供一個關鍵字以表示構成步驟(1)到(3)的一個表格格式程式組的步驟。

30. 如申請專利範圍第26項所述之程式設計方法，其中所述任務狀態不必按照順序彼此依次列出。

31. 如申請專利範圍第26項所述之程式設計方法，其中步驟(1)到(3)定義第一任務表，所述程式設計方法還包括定義一個不同的第二任務表的步驟。

32. 一種用於對一個遠端計算設備程式設計的方法，包括步驟：

(1)規定x種配置狀態，所述配置狀態中的至少一種定義一個或多個限定條件；

六、申請專利範圍

（2）規定y個由所述計算設備執行的路徑；

（3）對步驟(1)的每個限定條件，當滿足一個特定的限定條件時，規定一個由所述計算設備執行的路徑；

（4）規定一種限定配置狀態變成有效配置狀態；

（5）將表示上述步驟的數位資料存入一個本地計算設備；以及

（6）利用通信連接將步驟(5)的數位資料下載到所述遠端計算設備中。

33. 如申請專利範圍第32項所述之方法，其中所述通信連接是為網路。

34. 如申請專利範圍第33項所述之方法，其中所述網路包括因網際網路、內聯網、外聯網或LAN。

35. 如申請專利範圍第32項所述之方法，還包括將一個路徑元素指向於以不同格式的第二程式設計語言編寫而成的程式的步驟。

36. 如申請專利範圍第32項所述之方法，還包括評估所述遠端計算設備結構的步驟和將上述步驟配置成同所述遠端計算設備結構運行的步驟。

37. 如申請專利範圍第32項所述之方法，還包括將所述步驟中規定的至少部分資料形成一個表格格式的步驟。

38. 如申請專利範圍第32項所述之方法，還包括將表示所述配置狀態和路徑的數位資料存入所述遠端計算設備中以便執行的步驟。

39. 如申請專利範圍第32項所述之方法，還包括將配置狀

六、申請專利範圍

態和路徑的至少部分說明轉換成不同格式的第二程式設計語言的步驟。

40. 如申請專利範圍第32項所述之方法，其中配置狀態或路徑不必按照順序依次列出。

41. 一種由第一處理器和第二處理器構成的多處理器計算設備，其中所述第一處理器用於執行一個具有m種配置狀態和n個路徑的表格格式程式的至少一部分。

42. 如申請專利範圍第41項所述之多處理器計算設備，其中第二處理器執行用不是表格格式的第二語言編寫而成的程式。

43. 如申請專利範圍第42項所述之多處理器計算設備，其中第二處理器用於執行一個由所述第一處理器執行的表格格式程式所指導的程式。

44. 如申請專利範圍第41項所述之多處理器計算設備，其中所述第一和第二處理器位於積體電路上。

45. 如申請專利範圍第41項所述之多處理器計算設備，其中處理器之一可執行的至少一個指令不同於另一個處理器可執行的指令集。

46. 如申請專利範圍第41項所述之多處理器計算設備，還包括用於在所述第一和第二處理器之間傳遞參數的裝置。

47. 一種用於構成一個適於對具有m種配置狀態和n個路徑的表格格式程式進行編譯的編譯器的方法，包括步驟：

(1)識別表示配置狀態的區域；

(2)識別表示路徑的區域；

六、申請專利範圍

(3)識別一種配置狀態的至少一個限定條件並將其與具體的路徑相連接；以及

(4)將配置狀態A與一個將引用所述配置狀態A作為其元素的路徑相連接。

48. 如申請專利範圍第47項所述之方法，還包括用於將步驟(1)到(4)的功能綜合為一個現有語言的編譯器的過程。

49. 如申請專利範圍第47項所述之方法，在步驟(1)中還包括一個識別表示配置狀態開始部分的關鍵字的過程。

50. 如申請專利範圍第47項所述之方法，在步驟(2)中還包括一個識別表示路徑開始部分的關鍵字的過程。

51. 如申請專利範圍第47項所述之方法，還包括使一個用戶定制運算式與表格格式程式設計語言的一個具體指令相等的步驟。

52. 如申請專利範圍第51項所述之方法，還包括識別一個使用戶定制運算式與表格格式程式設計語言的預定義指令集相等的表格的步驟。

53. 如申請專利範圍第47項所述之方法，還包括根據所述表格格式程式的指令集識別不可執行的標號的步驟。

54. 如申請專利範圍第53項所述之方法，還包括將所述不可執行的標號與一個外部程式相連接的步驟。

55. 如申請專利範圍第54項所述之方法，其中所述外部程式是用包括表格格式程式設計語言在內的任何可用程式設計語言編寫而成的程式組成。

56. 如申請專利範圍第47項所述之方法，還包括區分兩個

六、申請專利範圍

或多個程式組的步驟，其中每個程式組由至少一個配置狀態表和一個路徑表組成。

57. 如申請專利範圍第56項所述之方法，還包括對所述多個程式組之間的交互作用進行編譯的步驟。

58. 如申請專利範圍第57項所述之方法，還包括定義一個用於啟動一個程式組的指令的步驟。

59. 如申請專利範圍第58項所述之方法，其中所述指令用一個或多個任務表表示。

60. 一種對計算裝置程式設計之方法，包括步驟：

(1)選擇一種具有預先定義的指令集的程式設計語言；

(2)定義用於表示選定語言的一個具體指令的替換運算式；

(3)用所述替換運算式編寫程式；以及

(4)將所述程式配置成所述計算裝置可執行的程式。

61. 如申請專利範圍第60項所述之方法，其中選定的程式設計語言是表格格式程式設計語言，包括：

x種配置狀態，所述配置狀態中的至少一種定義了一個或多個限定條件；

y個能被所述計算設備執行的路徑；以及

當滿足一個特定限定條件時，所述計算設備就執行一個路徑。

62. 如申請專利範圍第60項所述之方法，其中步驟(4)是一個由編輯器、編譯器、譯碼器或一個翻譯程式執行的翻譯

六、申請專利範圍

過程。

63. 如申請專利範圍第62項所述之方法，還包括顯示用經過翻譯的選定語言的預定的指令集組成的程式的步驟。

64. 如申請專利範圍第62項所述之方法，還包括顯示用用戶定制運算式組成的程式的步驟。

65. 如申請專利範圍第60項所述之方法，還包括提供一個位於程式內部的表格以便將所述替換運算式與相應的特定指令進行鏈結的步驟。

66. 一種用於對回應一個或多個虛擬限定執行一個或多個路徑的計算設備進行程式設計之程式設計方法，所述程式設計方法包括步驟：

(1)規定x種配置狀態，所述配置狀態中的至少一種定義了一個或多個限定條件；

(2)定義步驟(1)中的限定條件中的至少一個以表示一個虛擬限定；

(3)規定y個由所述計算設備執行的路徑；

(4)對於步驟(1)中的每個限定條件，當滿足所述限定中的一個具體限定條件時，再規定一個由所述計算設備執行的路徑；

(5)規定限定配置狀態中的一個變成有效配置狀態。

67. 如申請專利範圍第66項所述之程式設計方法，其中所述計算設備被定義為第一計算設備；所述程式設計方法還包括利用通信連接從與所述第一計算設備距離遙遠的第二計算設備接收表示上述步驟的數位資料的步驟。

六、申請專利範圍

68. 如申請專利範圍第66項所述之程式設計方法，還包括規定一種或多種配置狀態以構成一種輸出配置的步驟。

69. 如申請專利範圍第68項所述之程式設計方法，其中所述輸出配置定義計算設備的一個或多個輸出終端的輸出條件。

70. 如申請專利範圍第68項所述之程式設計方法，其中所述輸出配置定義由計算設備產生的虛擬計算輸出。

71. 如申請專利範圍第66項所述之程式設計方法，還包括將配置狀態和路徑說明轉換成所述計算設備可執行的數位資料的步驟。

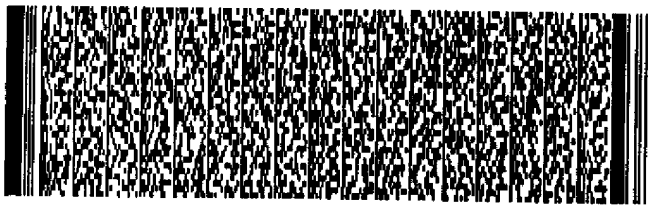
72. 如申請專利範圍第66項所述之程式設計方法，還包括用於轉換利用不同格式的第二程式設計語言的配置狀態和路徑的至少部分說明的步驟。

73. 如申請專利範圍第66項所述之程式設計方法，其中x種配置狀態和y個路徑不必按照順序彼此依次列出。

74. 如申請專利範圍第66項所述之程式設計方法，還包括提供一個用於識別由所述步驟組成的程式位置的關鍵字的步驟。

75. 如申請專利範圍第66項所述之程式設計方法，還包括將步驟(1)中的配置狀態分成一個或多個表格的步驟，每個表格中有一種配置狀態被規定為有效。

76. 如申請專利範圍第66項所述之程式設計方法，還包括構成一個獨立的用於確定由步驟(1)到(7)組成的程式是有效還是無效的表格的步驟。



六、申請專利範圍

77. 如申請專利範圍第66項所述之程式設計方法，還包括下列步驟：

(6)提供一個預先定義的指令集以便對路徑和配置狀態程式設計；

(7)定義用於表示步驟(6)中的指令集的一個特定指令的替換運算式；

(8)用替換運算式編寫程式；以及

(9)將所述程式配置成所述計算裝置可執行的程式。

78. 如申請專利範圍第77項所述之程式設計方法，其中步驟(9)是由編輯器、編譯器、譯碼器或一個翻譯程式執行的翻譯過程。

79. 如申請專利範圍第77項所述之程式設計方法，還包括提供一種於替換運算程式與對應特定指令間相互參照之表格的步驟。

80. 如申請專利範圍第66項所述之方法，還包括規定一種配置狀態為有效的步驟。

81. 如申請專利範圍第66項所述之方法，其中將一個路徑定義為在初始化時將被執行的預設路徑。

82. 一種用於在計算設備的程式列表中識別預定類型的字(wordings)之程式設計方法，包括從多種預定字格式(wording style)選擇中選擇一種字格式的步驟。

83. 如申請專利範圍第82項所述之程式設計方法，還包括步驟：

(1)定義x種配置狀態，所述配置狀態中的至少一種定

六、申請專利範圍

義了一個或多個限定條件；

（2）定義由計算設備執行的y個路徑；以及

（3）當滿足一個具體限定條件時，向計算設備分配一個路徑以執行。

84. 如申請專利範圍第82項所述之程式設計方法，其中利用一個具有關鍵字的表格來定義所述字格式。

85. 如申請專利範圍第82項所述之程式設計方法，其中所述預定字型被配置成高亮度的用戶定制標號。

86. 如申請專利範圍第82項所述之程式設計方法，還包括形成兩組或多組用戶定制字格式並且其中一組被定義為有效的步驟。

87. 一種用於對回應一個或多個限定條件執行一個或多個路徑的計算設備進行程式設計之程式設計方法；所述程式設計方法包括步驟：

（1）規定x種配置狀態，其中x 是大於等於1的整數；

（2）規定一個或多個到於所述x種配置狀態中的至少一種的限定標號；

（3）使步驟(2)中的限定標號指向一個用於描述限定條件或所述標注所表示的條件的獨立的運算式；以及

（4）規定y個可執行路徑，其中回應步驟(3)中規定限定條件來執行至少一個路徑。

88. 一種計算裝置，包括：

用於執行程式的計算裝置；

用於訪問於一個遠端計算設備相連接的通信連接的裝

六、申請專利範圍

置；以及

用於存儲所述計算裝置或所述遠端計算設備所執行的數位資料的存儲裝置；其中所述數位資料包括具有x種配置狀態和y個路徑的表格格式程式表示；所述配置狀態中的至少一種定義了一個或多個限定條件；而在滿足一個提單限定條件時，所述計算裝置執行所述路徑中的一個。

89. 一種為計算設備進行程式設計之程式設計工具，其包含：

具有第一多數配置狀態的第一表格(113)，所述配置狀態中的至少一種定義了一個或多個限定條件；

規定了第二多數路徑的第二表格，當滿足所述第一表格中列出的一個限定條件時，執行所述路徑中的至少一個；及規定為有效狀態之一種配置狀態；上述工具之特色更包含下述元件及特性之至少一個：

(1)至少一個配置狀態包含虛擬限定；

(2)至少一個第二多數路徑具有一個或多個標號，其中每個標號代表一個執行程式；

(3)一個或多個表格(103,200)規定使用者定義運算式以代表對應預先定義指令之程式語言；

(4)所述計算設備(803)是經由一通信連接(802)而與本地計算設備(801)連接之遠程電腦；其中數位資料是代表儲存於遠程計算設備(803)中之第一及第二表格經由通信連接而下載至本地計算設備(801)；

(5)所述計算設備(803)包含至少一個第一處理器及一

六、申請專利範圍

個第二處理器；其中第一處理器是翻譯第一及第二表格關係，同時給予第二處理器執行其他程式之指向；

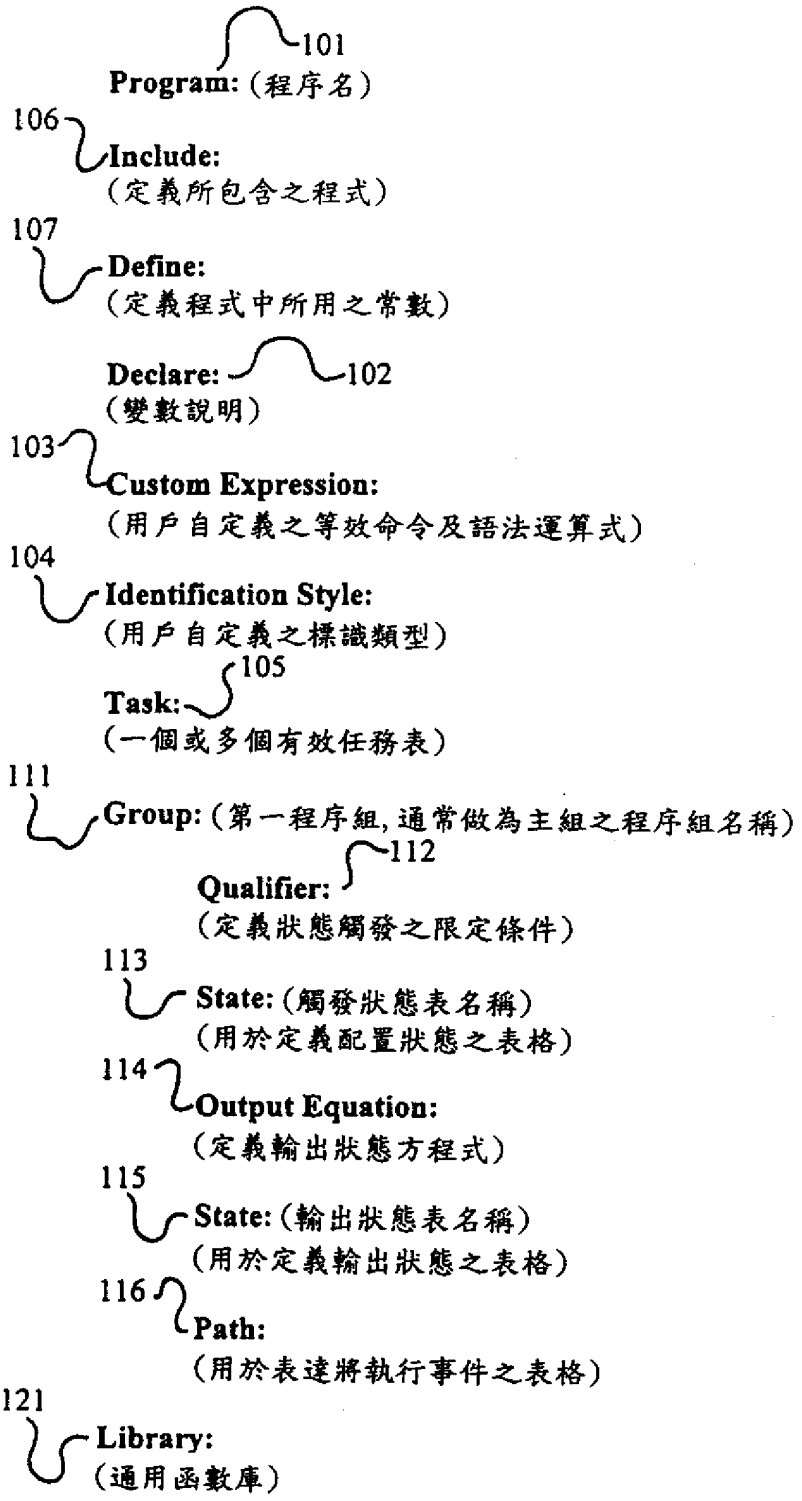
(6)至少一個配置狀態包含有一個表格及所述表格相等於該表格所代表獨立運算式所定義之限定條件；

(7)表格(104)定義一個較佳形態之關鍵字；

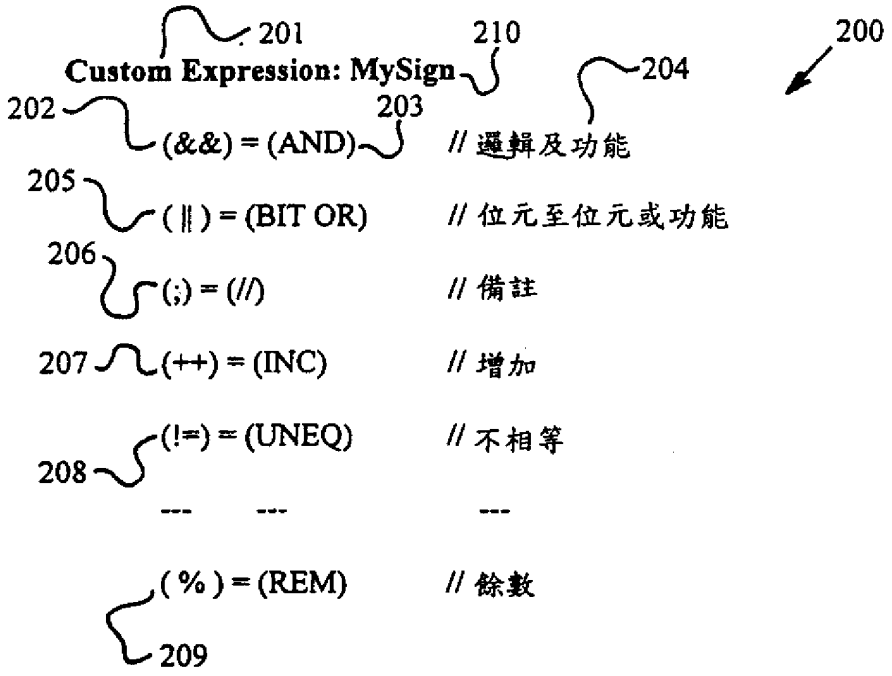
(8)表格(105)具有第三多數之任務狀態以定義第四多數任務之有效；

(9)表格(114)定義輸出運算式之第五多數，每個輸出運算式代表配置狀態之輸出狀況或列於第二表格(116)中之路徑元件；

(10)一個更進一步狀態之表格定義配置狀態之第六多數及一個更進一步路徑表格定義路徑之第七多數；其中第一及第二表格如第一表格組執行第一功能般被組合；及更進一步之狀態路徑表格如第二表格組執行不同第二功能般被組合。

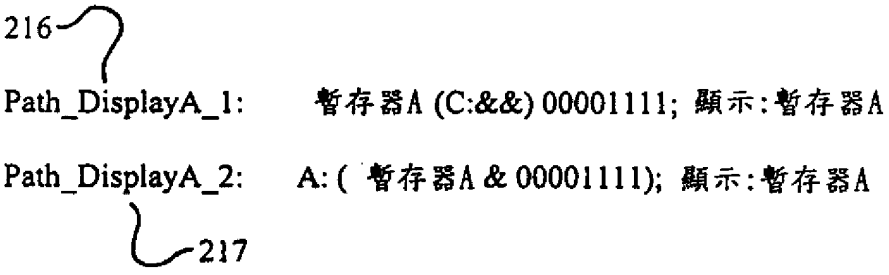

101
Program: (程序名)
106
Include:
(定義所包含之程式)
107
Define:
(定義程式中所用之常數)
Declare:
102
(變數說明)
103
Custom Expression:
(用戶自定義之等效命令及語法運算式)
104
Identification Style:
(用戶自定義之標識類型)
105
Task:
(一個或多個有效任務表)
111
Group: (第一程序組, 通常做為主組之程序組名稱)
112
Qualifier:
(定義狀態觸發之限定條件)
113
State: (觸發狀態表名稱)
(用於定義配置狀態之表格)
114
Output Equation:
(定義輸出狀態方程式)
115
State: (輸出狀態表名稱)
(用於定義輸出狀態之表格)
116
Path:
(用於表達將執行事件之表格)
121
Library:
(通用函數庫)

第一圖

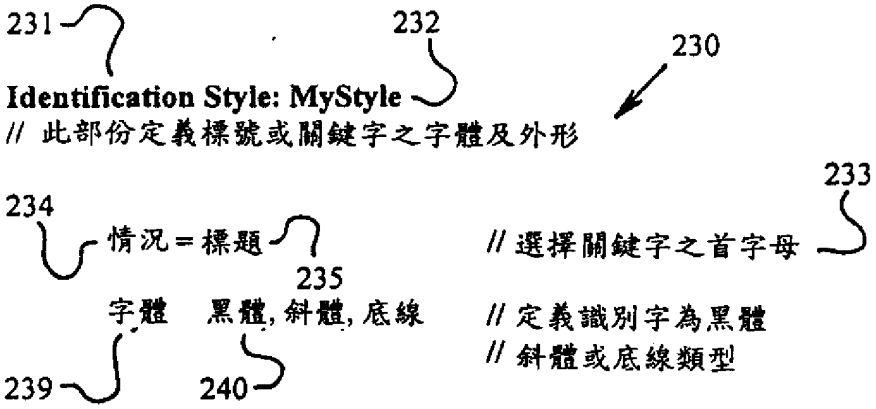

201
210
200
Custom Expression: MySign
204
202
203
(&&) = (AND)
// 邏輯及功能
205
(||) = (BIT OR)
// 位元至位元或功能
206
(;) = (//)
// 備註
207
(++) = (INC)
// 增加
(!=) = (UNEQ)
// 不相等
208

(%) = (REM)
// 餘數
209

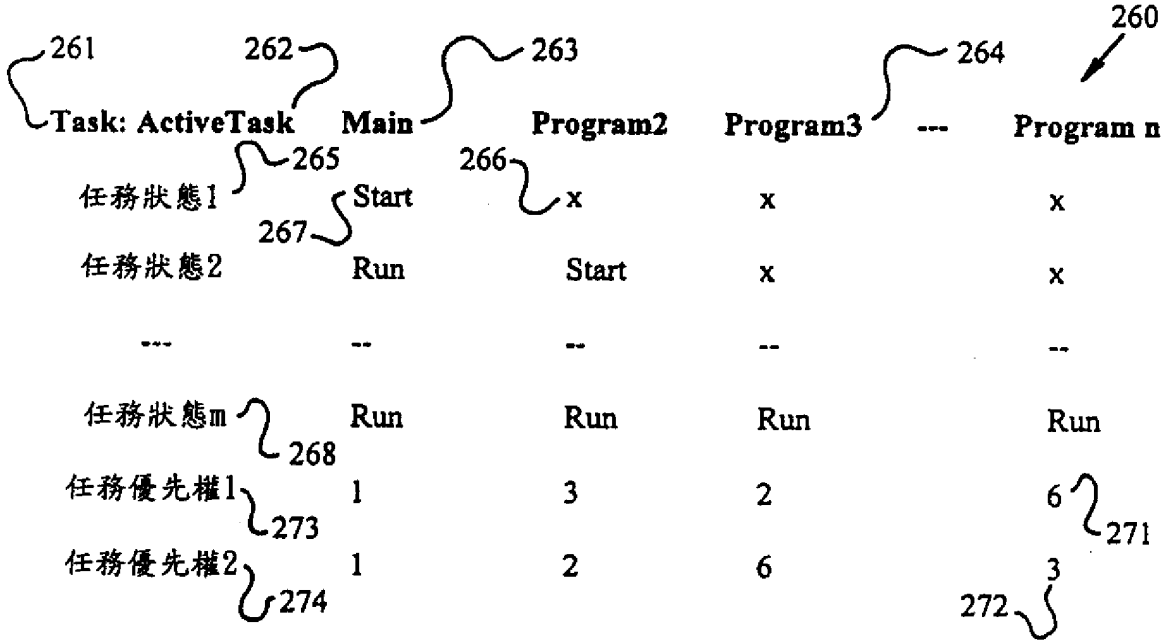
第二A圖


216
Path_DisplayA_1: 暫存器A (C:&&) 00001111; 顯示:暫存器A
Path_DisplayA_2: A:(暫存器A & 00001111); 顯示:暫存器A
217

第二B圖


231
232
230
Identification Style: MyStyle
// 此部份定義標號或關鍵字之字體及外形
233
234
情況＝標題
235
// 選擇關鍵字之首字母
字體 黑體, 斜體, 底線
// 定義識別字為黑體
// 斜體或底線類型
239
240

第三圖


260
261
262
263
264
Task: ActiveTask
Main
Program2
Program3

Program n
265
266
任務狀態1
Start
x
x
x
267
任務狀態2
Run
Start
x
x

--
--
--
--
任務狀態m
Run
Run
Run
Run
268
任務優先權1
1
3
2
6
271
273
任務優先權2
1
2
6
3
274
272

第四圖

Task: Input	Keyboard	Mouse	GamePort
All:	Run	Run	Run
Normal:	Run	Run	x
Keyboard:	Run	x	x
Mouse:	x	Run	x
Game:	x	x	Start

Task:Ports	COM1	COM2	ParallelPort
All:	Run	Run	Run
SrMouse:	Start	x	x
ExModem:	x	Continue	x
Printer:	x	x	Continue

Task:Device	CDRom	HardDriveC	Floppy
ReadCD:	Start	Run	x
HDFullSpeed:	x	Run	x
RWFloppy:	x	Run	Run
ALL:	Run	Run	Run

//可能之任務狀態:開始, 繼續, 進行, 暫停, x

第五A圖

301 Task: Input	302 Keyboard	303 Mouse	304 GamePort
305 All:	Run	Run	Run 310
306 Normal:	Run	Run	x
307 Keyboard:	Run	x	x
308 Mouse:	x	Run	x 311
Game: 309	x	x	Start

321 Task:Ports	322 COM1	323 COM2	324 ParallelPort
325 All:	Run	Run	Run
326 SrMouse:	Start	x	x
327 ExModem:	x	Continue	x 329
Printer: 328	x	x	Continue

341 Task:Device	342 CDRom	343 HardDriveC	Floppy
ReadCD: 345	Start	Run	x 344
346 HDFullSpeed:	x	Run	x
347 RWFloppy:	x	Run	Run
ALL: 348	Run	Run	Run

349 //可能之任務狀態:開始, 繼續, 進行, 暫停, x

第五B圖

```
1   Group: Main                 //WebSale主程序組

2   Qualifier:
3   Catalog = Icon(1)
4   Purchase = Icon(2)
5   Service = Icon(3)
6   Home = Icon(4)
7   Quit = Icon(5)

8   State: FirstPage    Catalog     Purchase    Service     Home        Quit
9   Ready:              P_catalog   P_purchase  P_service   Start       Bye
10  Hold1:              x           x           x           Start       Bye

11  State: Response     Group:Info  Group:Order Group:Service Group:Register P3.1
12  WindowCatalog:      Run         x           x           x           x
13  WindowPurchase:     x           Run         x           Run         x
14  WindowService:      x           x           Run         Run         x
15  Hold2:              x           x           x           x           x
16  Hold3:              x           Run         x           Run         x
17  Beep:               Continue    Continue    Continue    Continue    P+

18  Path:
19  Start:              CheckSystem; DisplayFirstPage; Beep; Hold 2; Ready; END
20  P_catalog:          WindowCatalog; BuySolicit; END
21  P_purchase:         WindowPurchase; Hold1; Hold3; GrayButton; END
22  P_service:          WindowService; BuySolicit; END
23  Bye:                Terminate; END

24  EOG                 //表示程序組結束之關鍵字

//注意:行號是為了方便描述,在實際中不需要
//程式設計:狀態及路徑方程式不必按照順序排列
```

第六A圖

```
1  Group: Main                    //WebSale 主程序組

2  Qualifier:
3  Catalog = Icon(1)
4  Purchase = Icon(2)
5  Service = Icon(3)
6  Home = Icon(5)
7  Quit = Icon(4)

8  State: FirstPage   Catalog     Purchase     Service      Home          Quit
9  Ready:             P_catalog   P_purchase   P_service    Start         Bye
10 Hold1:             x           x            x            Start         Bye

11 State: Response    Group:Info  Group:Order  Group:Service Group:Register P3.1
12 WindowCatalog:     Run         x            x            x             x
13 WindowPurchase:    x           Run          x            Run           x
14 WindowService:     x           x            Run          Run           x
15 Hold2:             x           x            x            x             x
16 Hold3:             x           Run          x            Run           x
17 Beep:              Continue    Continue     Continue     Continue      P+

18 Path:
19 Start:             EJ_CheckSystem; EVB_DisplayFirstPage; Beep; Hold 2; Ready; END
20 P_catalog:         WindowCatalog; Lbry:BuySolicit; END
21 P_purchase:        WindowPurchase; Hold1; Hold3; Lbry:GrayButton; END
22 P_service:         WindowService; Lbry:BuySolicit; END
23 Bye:               EJ_Terminate; END

24 Library: Local
25 BuySolicit:        EC_CheckRecord; EVC_SolicitWindow; END
26 GrayButton:        EC_CheckX; EVB_X_Is_Gray; END

27 EOG                //表示程序組結束之關鍵字

// 注意:行號是為了方便描述,在實際中不需要
// 程式設計:狀態及路徑方程式不必按照順序排列
```

第六B圖

```
Include:
ReadIR                  //從遠程控制讀取信號
ReadPanel               //從控制面板讀取信號
ReceiveFromLine         //從電纜線輸入數據
SendToLine              //向電纜線傳輸數據
CompressBuffer          //在向電纜線發送之前壓縮數據
DecompressBuffer        //對數據文件進行解壓縮
CompareTime             //比較暫存器時間及實時時鐘
GenIcon                 //在電視銀螢幕上生成圖標
GenWindow               //建立一個新視窗
InputCursor             //輸入並譯碼游標之移動
OutputCursor            //更新游標在電視螢幕上之位置
DisplayData             //在電視螢幕上顯示數據
DisplayPicture          //在電視螢幕上顯示圖形文件
.....                   //.....
.....                   //.....
ErrorMessage            //對錯誤代碼進行譯碼並傳輸錯誤消息
Help                    //顯示幫助螢幕
```

第七圖

```
Include:
EA_ReadIR                 //從遠程控制讀取信號
EE_ReadPanel              //從控制面板讀取信號
EA_ReceiveFromLine        //從電纜線輸入數據
EA_SendToLine             //向電纜線傳輸數據
EC_CompressBuffer         //在向電纜線發送之前壓縮數據
EC_DecompressBuffer       //對數據文件進行解壓縮
EA_CompareTime            //比較暫存器時間及實時時鐘
EJ_GenIcon                //在電視銀螢幕上生成圖標
EJ_GenWindow              //建立一個新視窗
EVC_InputCursor           //輸入並譯碼游標之移動
EVB_OutputCursor          //更新游標在電視螢幕上之位置
EVB_DisplayData           //在電視螢幕上顯示數據
EVB_DisplayPicture        //在電視螢幕上顯示圖形文件
.....                     //.....
.....                     //.....
EA_ErrorMessage           //對錯誤代碼進行譯碼並傳輸錯誤消息
EVB_Help                  //顯示幫助螢幕
```

第八圖

功　能	表格格式指令
指令分離器	
標識指令之結束	
指令之繼續	
標識後續行	&（在行首）
一元運算器	
方向	VALUE（地址）
地址	ADDRESS（值）
負的	
邏輯否定	NOT（運算式）
1的餘數	COMPLEMENT（運算式）
增加	INCREMENT
減少	DECREMENT
大小	SIZEOF（運算式）
移位操作器	
左移	BIT SHIFT LEFT（操作數, n）
右移	BIT SHIFT RIGHT（操作數, n）
關係操作器	
小於	<
大於	>
小於或等於	<= or =<
大於或等於	>= or =>
邏輯操作器	
位元 AND	BIT AND
邏輯 AND	AND
唯一 OR	XOR
位元 OR	BIT OR
邏輯 OR	OR
條件分支	運算式?:[正確語句][錯誤語句]
註解	
	//（至文件末端）

第九圖

輸入狀態:

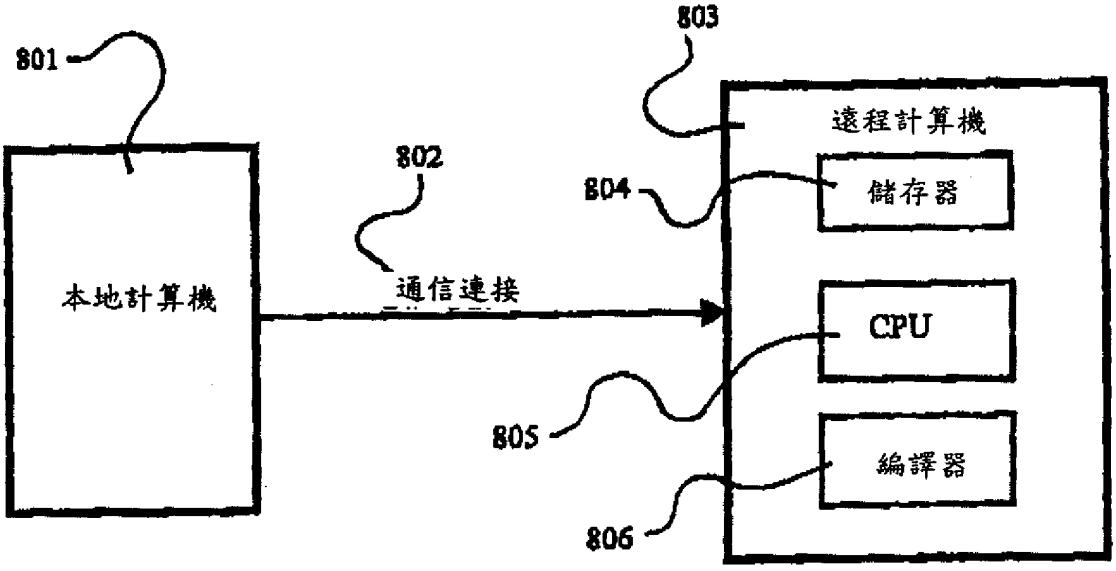
觸發端子:	TG1	TG2	TG3	TG4	TG5	TG6
State0:	R:Path1	R:Path2	R:Path3	R:Path4	X	X
State1:	F:Path11	R:Path2	R:Path3	R:Path4	X	X
State2:	R:Path1	F:Path11	R:Path3	R:Path4	X	X
State3:	R:Path1	R:Path2	F:Path11	R:Path4	X	X
State4:	R:Path1	R:Path2	R:Path3	F:Path11	X	X

;

Path:

Path 1:	State1	Sound1	Path1
Path 2:	State2	Sound2	Path2
Path 3:	State3	Sound3	Path3
Path 4:	State4	Sound4	Path4
Path11:	State0	結束	

第十圖
(現有技術)



第十一圖