

66941

89年11月7日

公告本

申請日期	89.5.25
案 號	89106947
類 別	G06F 9/00

A4
C4

中文說明書修正頁(89年11月)

(以上各欄由本局填註)

發 明 專 利 說 明 書 472209 新 型		
一、發明 名稱	中 文	於模擬器系統中更新使用者記憶體之方法及系統
	英 文	METHOD AND SYSTEM FOR UPDATING USER MEMORY IN EMULATOR SYSTEMS
二、發明 人	姓 名	布雷恩
	國 籍	美國
	住、居所	美國亞利桑納州麻塞城塔伯街4220號
三、申請人	姓 名 (名稱)	美商微晶片科技公司
	國 籍	美國
	住、居所 (事務所)	美國亞歷桑那州85224-6199錢德勒市西錢德勒大道2355號
	代 表 人 姓 名	菲利浦·查普曼

裝 訂 線

經濟部智慧財產局員工消費合作社印製

五、發明說明 (/)

發明背景

1. 發明之技術領域

本發明係有關於一種模擬器系統及模擬器裝置，特別是針對一種可以輕易地更新離晶片(off-chip)以及晶片上(on-chip)記憶體的模擬器系統及模擬器裝置。

2. 相關技藝之敘述

許多微型控制器裝置具有包含晶片上(on-chip)記憶體以及離晶片(off-chip)使用者記憶體的記憶體系統。在模擬器系統中，需要能夠更新上述兩種類型的記憶體之內容。一般而言，模擬器系統可以輕易地更新晶片上記憶體，然而，更新離晶片使用者記憶體卻顯得較為複雜。舉例來說，有一種獲得存取使用者記憶體的習知方法，係透過一種複製之模擬器控制方塊，以提供一種複製的直接存取。使用模擬器系統的主機(host)系統則藉由模擬器控制方塊的使用，而直接控制使用者記憶體。在另一種方式中，一既存的模擬器控制方塊可被裝配有額外的連接功能以及一額外的匯流排。無論是使用複製的模擬器控制方塊或是必須增加額外的連接功能與一額外的匯流排，均會同時增加系統的成本與複雜度。

在另一種習用的系統中，即本申請案受讓人所製造之PIC17C01模擬器裝置，對於晶片上(模擬器程式)記憶體以及離晶片(使用者)記憶體均可存取。然而，模擬器裝置必須產生記憶體存取週期，以藉由操作I/O位元而存取離晶片(使用者)記憶體。更詳而言之，當需要從使用者記憶體

五、發明說明(2)

中讀出時，主機系統會從模擬器程式記憶體中下載程式區段(segment)，並開始於PIC17C01中執行該程式區段。程式區段將資料門鎖(data latch)寫入至埠C、D與E，並且將資料方向暫存器(data direction register, DDR)寫入至埠C、D與E，以將他們配置成輸出端。主機系統由MP模式改變成MC模式，同時將埠C、D與E從系統匯流排模式改變成I/O埠模式。資料方向暫存器(DDR)在之前便已被設置，並且驅動成輸出端。主機系統開始下載程式區段，以在PIC17C01中進行該程式區段的模擬器程式記憶體執行，並且開始於PIC17C01中執行該程式區段。

該程式區段接著寫入至埠C、D與E，以模擬一系統匯流排並且讀出所要的記憶體位置。一隨機存取記憶體(RAM)位址被寫入至埠C與D，而埠E則被設定以使得位址門鎖致能(ALE)訊號位於高位準。資料方向暫存器中的埠C與D被配置成輸入端而被寫入，而資料方向暫存器中的埠E則被設定以使得讀出(OE)訊號位於低位準。資料在埠C與D被讀出，且該資料被儲存於PIC17C01中的隨機存取記憶體內。主機系統再接著由MP模式改變成MC模式，下載程式區段至模擬器程式記憶體中，並且開始在PIC17C01中執行該程式區段。該程式區段將隨機存取記憶體中的資料傳送到主機系統。

在被下載至模擬器程式記憶體的程式區段將一隨機存取記憶體位址寫入至埠C與D，並且設定埠E使得位址門鎖致能(ALE)訊號位於高位準時，寫入程序是類似的。即將

五、發明說明(3)

被寫入至使用者程式記憶體之資料被寫入至資料方向暫存器中的埠C與D，而資料方向暫存器中的埠E則被設定以使得寫入(WR)訊號位於低位準。

發明之概要說明

本發明之一目的乃在於提供一種模擬器系統、裝置及方法，其提供一種簡單而有效率之存取離晶片(off-chip)使用者記憶體的方式。

本發明之另一目的乃在於提供一種模擬器系統與裝置，其編碼係執行於該模擬器記憶體中，且讀出與寫入存取係被導入至離晶片使用者記憶體。

本發明之又一目的乃在於提供一種連接至一模擬器系統的模擬器裝置以及一種使用者系統，其提供一種簡單而有效率之存取該使用者系統中之程式記憶體的方式。

本發明之這些以及其他目的係可藉由一種模擬器裝置而達成，其中該模擬器裝置包括一記憶體介面，用來存取程式記憶體，其中該程式記憶體包括一第一記憶體、一外接至該裝置的第二記憶體；以及一選擇電路，其係被連接至介面，以於該裝置係配置以從該第一記憶體擷取指令時，將該程式記憶體讀出與記憶體寫入存取只導入至該第二記憶體。

該裝置更可包括一連接至該選擇電路的電路，以偵測是否該表區(table)讀出與表區寫入存取之至少一者即將被執行，其中該選擇電路僅將該表區讀出與表區寫入存取之至少一者只導入至該第二記憶體。該裝置亦可包括一模式

五、發明說明(4)

選擇電路，其中該選擇電路包括一切換裝置，其係連接至該第一與第二記憶體，並且係被連接以接收一由該模式選擇電路所輸出之訊號。

該裝置亦可包括一指令解碼器，其輸出一訊號，代表程式記憶體讀出存取指令與程式記憶體寫入存取指令之至少一者即將被解碼。一電路可被連接至該解碼器，該電路係被配置以接收該訊號並且被配置以執行程式記憶體讀出存取指令與程式記憶體寫入存取指令之至少一者。

當該裝置具有該模式選擇電路時，該電路亦可包括一邏輯電路，其係被連接以接收該模式選擇電路之輸出；以及一指令解碼器，其具有一連接至該邏輯電路的輸出端，其中該介面電路係連接至該邏輯電路的輸出端。

該模式選擇電路亦可包括一構件，其用來輸出一代表該裝置之一操作模式的訊號，而該指令解碼器可包括一構件，其用來輸出一代表程式記憶體讀出存取指令與程式記憶體寫入存取指令之至少一者即將被解碼的訊號。該邏輯電路可被連接以接收由該二構件所輸出之訊號，並且輸出一代表對於該第一與第二記憶體存取之何者即將被致能的訊號至該選擇電路。

該記憶體介面可包括一程式記憶體匯流排、以及一連接至該匯流排之程式記憶體匯流排控制器。該選擇電路可包括一多工器，其係連接至該程式記憶體匯流排、一第一記憶體存取匯流排以及一第二記憶體存取匯流排；以及一電路，其係連接至該多工器，以在該第一與第二記憶體存

五、發明說明(5)

取匯流排之間作選擇。該電路可包括一構件，其用來產生一輸出至該多工器的訊號，以於該裝置被配置以從該第一記憶體中擷取指令時，表示僅有存取至該第二記憶體。該構件可包括一模式選擇電路；一電路，其產生一代表程式記憶體存取即將被執行的訊號；以及一第一邏輯電路，其係被連接以接收該模式選擇電路之輸出，並具有一輸入端，其係被連接以接收由該電路所輸出之訊號。

該第一記憶體係可為一模擬器程式記憶體，而該第二記憶體係可為一使用者程式記憶體。

一模擬器系統以及一使用者系統亦可被連接至該裝置。該模擬器系統可包括該第一記憶體，而該使用者系統則可包括該第二記憶體。該第一記憶體可包括一模擬器程式記憶體，而該第二記憶體則可包括一使用者程式記憶體。

上述之目的以及其他目的亦可藉由一種模擬器裝置於該裝置被配置從一模擬器記憶體中讀出指令時而達成，其中該模擬器裝置包括一構件，係用來接收產生自一模擬器記憶體而傳送至該裝置的指令，以及一構件，係被連接至該用來接收的構件，而於該裝置係配置以從該模擬記憶體擷取指令時，用來只瞄準記憶體讀出與寫入指令至連接至該裝置的使用者記憶體。該裝置亦可包括一構件，係用來偵測記憶體讀出與寫入指令，其係連接至該用來接收之構件；以及一構件，係用來選擇該裝置之操作模式，其係連接至該用來瞄準之構件以及該用來偵測之構件。

五、發明說明 (6)

該用來瞄準之構件可包括一構件，係用來偵測該裝置之操作模式；一構件，係用來偵測記憶體讀出與寫入指令；以及一構件，係使用兩個用來偵測的構件之輸出端而用來在該模擬器記憶體以及該使用者記憶體之間作選擇。該裝置亦可包括一構件，係在該用來選擇的構件之控制下，用來在存取至該模擬器記憶體以及該使用者記憶體之間作切換。

上述之目的以及其他目的亦可藉由一種操作一具有僅從一第一記憶體中擷取指令之步驟的模擬器裝置，以及僅對與該第一記憶體分離而外接至該模擬器裝置之一第二記憶體導入記憶體存取的方法而達成。指令可以僅從一模擬器程式記憶體中擷取，而記憶體存取也只可以對一與該模擬器程式記憶體分離之使用者程式記憶體執行。該方法亦可包括將表區讀出與表區寫入存取之至少一者導入該程式記憶體。

該方法亦可包括偵測該裝置之操作模式；偵測一記憶體存取是否即將被執行；以及根據該偵測步驟，選擇該第一記憶體與該第二記憶體之間的存取。偵測一記憶體存取是否即將被執行的步驟係可包括偵測是否表區讀出與表區寫入存取之至少一者即將被執行，而偵測該記憶體存取可包括將表區讀出與表區寫入存取之至少一者導入至該第二記憶體。

該方法亦可包括將指令解碼；使用該解碼步驟，偵測一記憶體存取是否即將被執行；以及使用該偵測步驟，決

五、發明說明(7)

定該第一與第二記憶體之何者將被存取。該裝置之操作模式亦可被偵測，而決定該第一與第二記憶體之何者將被存取係可藉由使用該偵測步驟而被執行。

圖式之簡要說明

本發明之更進一步的了解以及其許多伴隨而來的優點，將可藉由下列參照附圖所作之較佳具體實施例的詳細描述，而更為明白，其中：

圖1係為根據本發明的模擬器系統之一簡化方塊圖；

圖2係為根據本發明的模擬器晶片之一方塊圖；

圖3係為根據本發明的模擬器晶片所包括之電路圖；

圖4A至圖4C係為不同操作模式下的模擬器記憶體配置圖；

圖5係為根據本發明之表區讀出指令圖表；以及

圖6係為根據本發明之表區寫入指令圖表。

較佳具體實施例之詳細描述

現在，請參閱之圖式，特別是圖1，其係根據本發明繪示系統之一具體實施例。該系統包括一模擬器系統10、一模擬器晶片20、與一使用者系統30。模擬器系統10包括一模擬控制電路11、一位址門鎖12、以及一模擬器程式記憶體13。一主機系統40透過連接於主機系統40與模擬控制電路11之間之一匯流排41而與模擬器系統10溝通。透過一匯流排14，來自模擬器晶片20的位址被輸入至位址門鎖12，而且資料被轉換於模擬控制電路11與模擬器晶片20之間。模擬控制電路11亦被連接至一匯流排14。來自位址門

五、發明說明 (8)

鎖12的位址透過一匯流排15而被輸入至模擬器程式記憶體13。

位址門鎖12被連接至接腳EA、EBA0、與EALE，而模擬控制電路11則被連接至晶片20之若干接腳。模擬器程式記憶體13亦被連接至晶片20之模擬器輸出致能端、模擬器寫入高位準與模擬器寫入低位準接腳。匯流排21係被連接於模擬器系統10、模擬器晶片20、與使用者系統30之間。

使用者系統30包含有一使用者程式記憶體33以及一位址門鎖32。來自晶片20的位址透過匯流排31而被從位址門鎖32輸入至使用者程式記憶體33。晶片20的接腳UAD被連接至使用者程式記憶體33之資料輸入端，而且接腳UA、接腳UBA0、以及接腳UALE被連接至位址門鎖32。使用者記憶體輸出致能端、使用者寫入高位準與使用者寫入低位準接腳亦被連接至使用者程式記憶體33。

必須注意的是，模擬器程式記憶體13與使用者程式記憶體33一般具有不同之尺寸。離晶片(off-chip)使用者記憶體33通常比較大。

晶片20之數個接腳亦被連接至一隨從裝置(slave device)50。隨從裝置50提供了一部分的模擬器功能。模擬器晶片20係被設計以模擬大部分裝置的核心功能。隨從裝置50則模擬裝置的週邊功能。模擬器晶片20與隨從裝置50一起運作以模擬所欲之裝置。模擬器晶片20與隨從裝置50係被設計以分離使用，以允許僅使用一不同之隨從裝置而具有不同之週邊功能的不同類型之裝置的模擬。連接至隨

五、發明說明(7)

從裝置50的連線51至53呈現出模擬器晶片20與隨從裝置50至「鏢靶」系統之間的連線。換言之，此乃模擬器取代使用者系統中的晶片之處。

在本發明中，模擬器晶片20被設置於所欲之操作模式中。在一種模式中，即所謂「微處理器寫穿模式」(microprocessor write-through mode, MP/W)而在下文中詳述者，模擬器晶片20內的程式執行發生自模擬器程式記憶體13，而表區(table)讀出與表區寫入指令則發生於使用者程式記憶體33內。主機系統40使用模擬控制電路11，以下載程式區段(segment)至模擬器程式記憶體13。主機系統40於模擬器晶片20內開始執行程式區段。當讀取使用者程式記憶體33時，程式區段執行一個表區讀出的指令，以讀取使用者程式記憶體33。於模擬器晶片20內執行的程式區段會藉由模擬控制電路11與匯流排41而從模擬器晶片20轉換資料至主機系統40。

類似的操作發生於寫入使用者程式記憶體33時。模擬器晶片20被設置於MP/W模式中，使得程式執行發生自模擬器程式記憶體13，而表區讀出與表區寫入指令則發生於使用者程式記憶體33內。主機系統40使用模擬控制電路11，以下載程式區段至模擬器程式記憶體13。主機系統40於模擬器晶片20內開始執行程式區段。程式區段執行一個表區寫入的指令，以寫入使用者程式記憶體33。儲存於模擬器晶片20內的資料會被轉換至使用者程式記憶體33。

對於模擬器晶片20之較詳細的示意圖係如圖2所示。

五、發明說明 (/ 〇)

一程式記憶體介面60係藉由接腳61而與模擬器程式記憶體13以及使用者程式記憶體33相接。舉例而言，輸入端EA與EAD與模擬器程式記憶體13相接，而輸入端UA與UAD與使用者程式記憶體33相接。輸入至裝置的指令係藉由程式匯流排62，而被載入指令暫存器63內。指令暫存器63係連接至指令解碼與控制器67以及位址多工器76之間。圖2亦顯示了一個模擬控制電路66，其接收數個來自模擬器系統10之模擬控制電路11的輸入。值得注意的是下文所將詳述之3位元模式輸入。

連接至程式記憶體介面60者係為一表區讀出與表區寫入執行邏輯電路83。執行邏輯電路83係藉由一匯流排而連接至程式記憶體介面60。執行邏輯電路83亦被連接至指令解碼與控制器67，但未繪示於圖式中，且執行程式記憶體讀出與寫入指令，亦即表區讀出與表區寫入指令。執行邏輯電路83亦包括暫存器TBLPTR(表區指向器)與TABLAT(表區門鎖)，其使用於執行表區讀出與表區寫入指令時。此一電路的操作將配合圖3與圖5及圖6而在下文中被更詳細說明。

模擬器晶片20亦包括一時序產生器68，以產生各種使用於模擬器晶片20之時序訊號；以及包含有一電源啟動計時器(power-up timer)、一振盪啟動計時器(oscillator start-up timer)、一電源導通重置(power-on reset)、以及一重置計時器(watchdog timer)等元件的電路69。具有工作暫存器70的ALU 71係透過匯流排82而被連接至不同的電路，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(11)

計時器 77、週邊 78、以及資料監視器 79。圖 2 亦繪示了儲體(bank)選擇暫存器(BSR)73、狀態暫存器 74 以及檔案選擇暫存器(FSR)75。一資料記憶體介面 80 係被提供，以藉由接腳 81 而控制資料轉換至一資料記憶體(模擬資料 RAM)。資料記憶體通常會殘留於隨從裝置 50 中。由指令暫存器 63 所接收而留經位址多工器 76 的位址係藉由 RAM 位址匯流排 81，而被輸入至資料記憶體介面 80。

必須了解的是，圖 2 並非模擬器晶片 20 之完整示意圖，而且有許多其他的電路並未繪示於其中。圖 2 係引用以作為本發明的說明，而非用來顯示模擬器晶片 20 之每項特徵。

在微處理器中讀出與寫入至程式記憶體係通常透過稱作表區讀出與表區寫入的指令而被執行。這些指令允許資訊轉換於一資料記憶空間與一程式記憶空間之間。在本發明中，模擬器晶片 20 之邏輯重新引導表區讀出與表區寫入的指令，以允許對於使用者記憶體的存取。是以，使用者記憶體 33 很容易被存取。此點在以下的敘述中將更為明顯。

模擬器晶片 20 所包括之電路的更進一步描述係如圖 3 所示。一模式解碼邏輯電路 90 接收來自模擬控制電路 66 之 3 位元模式訊號的輸入。模式解碼邏輯電路 90 將該 3 位元模式訊號解碼，並且輸出一個邏輯"1"訊號於對應至所欲之操作模式的適當輸出線上。在這種情形裡，係說明了一種微控制器模式、一種微處理器模式、以及一種微處理器寫

五、發明說明(12)

透模式。用於這些模式之任一者的記憶體映射係繪示於圖4A至4C，並且在下文中更加詳細描述。必須了解的是，這三種模式僅用以作為本發明之說明，而其他的操作模式亦是可能的。

圖4A至4C係顯示在不同操作模式中的模擬記憶體映射。圖4A顯示保護之微控制器／微處理器模式，其中存取僅提供至模擬器程式記憶體。在微處理器模式(圖4B)中，存取僅提供至使用者程式記憶體。在另一方面，圖4C顯示一種稱作微處理器寫透模式的模式，其中所有的程式執行指令產生自模擬器程式記憶體，而表區讀出與表區寫入指令係產生於使用者程式記憶體內。

對於圖4A至4C所示之映射的說明僅用以理解本發明，其並不表示該使用者記憶體與模擬器記憶體係為同樣尺寸，或者必須為同樣尺寸。典型地，離晶片使用者記憶體遠大於模擬器程式記憶體。

圖3之電路一包括一多工器100，其係連接至模擬器系統匯流排14以及使用者系統匯流排21。多工器100係由邏輯電路95之輸出所控制，該邏輯電路95輸出一訊號於訊號線101上，使得該多工器100允許ESB的存取以及USB的存取。邏輯電路95包括及閘(AND gate)91與93、反向器94、以及或閘(OR gate)92。透過程式記憶體匯流排而連接至多工器100者係為一程式記憶體匯流排控制器99，其控制程式記憶體之讀出與寫入。從程式記憶體所接收到的指令係被輸入至指令解碼電路67。

五、發明說明 (13)

表區讀出／表區寫入指令執行邏輯電路83係透過標示為TBLRD(表區讀出)以及TBLWT(表區寫入)之訊號線，而被連接至指令解碼電路67。執行邏輯電路83包括兩個暫存器TBLPTR(表區指向器) 97與TABLAT(表區門鎖) 98，其係使用於執行表區讀出與表區寫入指令時，其操作係於下文中更進一步描述。執行邏輯電路83係藉由一程式記憶體讀出／寫入匯流排而被連接至程式記憶體匯流排控制器99。標示為TBLRD(表區讀出)以及TBLWT(表區寫入)之訊號線係被連接至一或閘96，其輸出則被連接至及閘91之一輸入端。訊號線102代表所有其他解碼之指令的輸出，其係被連接至模擬裝置的適當電路以進行執行操作。用以執行算術操作的ALU即為一例。

圖3之操作在此將被描述。在圖3所示的電路中，可能發生之記憶體週期有三種類型，分別為指令擷取、根據TBLRD(表區讀出)指令之表區讀出、以及根據TBLWT(表區寫入)指令之表區讀出。指令係被傳送至指令解碼電路67。這些指令被解碼成表區讀出、表區寫入、以及其他指令，其係如圖3所示之輸出端102上的群組者。當TBLRD(表區讀出)或TBLWT(表區寫入)被檢測時，指令執行邏輯電路83便由訊號啟動。執行邏輯電路83會傳送程式記憶體存取至程式記憶體匯流排控制器99。取決於模式接腳輸入端上的訊號，如果多工器控制訊號係為邏輯"0"，多工器將引導程式記憶體存取至ESB，而如果多工器控制訊號係為邏輯"1"，則多工器將引導程式記憶體存取至USB。

五、發明說明(14)

模式選擇決定了所將存取的記憶體。在微控制器模式中，總是會引導記憶體存取至ESB。因此，該微控制器模式訊號係被反向而接著被傳送至及閘86，使得多工器控制訊號永遠為邏輯"0"。在微處理器模式中，總是會引導記憶體存取至USB。因此，該微處理器模式訊號係被傳送至或閘88，使得多工器控制訊號永遠為邏輯"1"。

及閘91接收微處理器模式寫透訊號以及一產生自或閘96的訊號，以作為輸入。由於邏輯"1"訊號係由表區讀出或表區寫入線所輸出，邏輯"1"之或閘96訊號係產生於一讀出或寫入指令已被指令解碼電路67所解碼時。或閘96之此一輸出被傳送至及閘91，其亦接收模式解碼邏輯電路90之微處理器寫透輸出以作為輸入。當輸入至及閘91的訊號均為高位準，及閘91便會輸出一邏輯"1"訊號，使得或閘92亦輸出一邏輯"1"訊號。接著，由於在微處理器寫透模式中，為控制器線以及微處理器線上的訊號均定義為邏輯"0"，使得及閘93輸出一邏輯"1"訊號。在微處理器寫透模式中，讀出與寫入指令係朝向USB，而關於任何其他指令之所有其他記憶體存取係朝向ESB。因此，晶片的操作係藉由從ESB系統10擷取指令，而任何表區讀出或寫入指令係執行於USB系統30中。根據本發明，在本模式中，模擬器裝置允許使用者僅從模擬器程式記憶體中執行指令，而從使用者程式記憶體中讀出或寫入。

表區讀出與寫入指令係更詳細地顯示於圖5與圖6中。在表區讀出指令中，如圖5所示，晶片20之兩個暫存器係

五、發明說明(15)

被描述。TABLAT(表區門鎖)暫存器係為一表區門鎖，並且維持(hold)八個位元。該暫存器係維持(hold)載入21位元表區指向器暫存器TBLPTR(表區指向器)23的位址所指向之記憶體位置的內容。TBLRD(表區讀出)指令有四種選擇。在其中的三種中，TBLPTR(表區指向器)所指向之使用者記憶體33中的記憶體位置處之資料係被載入TABLAT(表區門鎖)。從運算元(operand)的角度觀之，在被載入TABLAT(表區門鎖)後，TABLAT(表區門鎖)中之值保持不變、或是增加、或是減少。在第四種情形中，TBLPTR(表區指向器)的值增加，而且TBLPTR(表區指向器)之增加值所指向之記憶體33中的記憶體位置係被載入TABLAT(表區門鎖)。

表區寫入指令係以類似的方法操作。如圖6所示，TBLWT(表區寫入)指令亦有四種選擇。在其中的三種中，TABLAT(表區門鎖)中的資料係被載入TBLPTR(表區指向器)所指向之使用者記憶體33中的記憶體位置。從運算元(operand)的角度觀之，TBLPTR(表區指向器)中之值保持不變、或是增加、或是減少。在第四種情形中，TBLPTR(表區指向器)的值增加，而且TABLAT(表區門鎖)中的資料係被載入TBLPTR(表區指向器)之增加值所指向之使用者記憶體33中的記憶體位置。

本發明之圖式與描述以較佳實施例說明如上，僅用於藉以幫助了解本發明之實施，非用以限定本發明之精神，而熟悉此領域技藝者於領悟本發明之精神後，在不脫離本發明之精神範圍內，當可作些許更動潤飾及同等之變化替

五、發明說明(16)

換，其專利保護範圍當視後附之申請專利範圍及其等同領域而定。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明(／／)

元件符號說明

10	模擬器系統	69	電路
11	擬控制電路	70	工作暫存器
12	位址門鎖	71	ALU
13	模擬器程式記憶體	73	儲體選擇暫存器
14	匯流排	75	檔案選擇暫存器
15	匯流排	76	位址多工器
20	模擬器晶片	77	計時器
21	匯流排	78	週邊
30	使用者系統	80	資料記憶體介面
31	匯流排	81	接腳
32	位址門鎖	82	匯流排
33	使用者程式記憶體	83	執行邏輯電路
40	主機系統	86	及閘
41	匯流排	88	或閘
50	隨從裝置	90	模式解碼邏輯電路
51-53	連線	91	及閘
60	程式記憶體介面	92	或閘
61	接腳	93	及閘
62	匯流排	94	反向器
63	指令暫存器	95	邏輯電路
66	模擬控制電路	96	或閘
67	指令解碼與控制器	97	暫存器 TBLPTR
68	時序產生器	98	暫存器 TABLAT

五、發明說明(18)

99	程式記憶體匯流排控	100	多工器
	制器	101	訊號線

(請先閱讀背面之注意事項再填寫本頁)

訂
線

四、中文發明摘要（發明之名稱：

）

發明名稱：於模擬器系統中更新使用者記憶體之方法及系統

發明摘要：

一種用來提供存取至模擬器系統內的使用者記憶體之裝置、系統、與方法。該模擬器系統包含有一模擬器系統記憶體、一使用者系統記憶體、以及一模擬器裝置。該模擬器裝置係操作於一種模式，其中程式執行指令係產生於該模擬器記憶體，而讀出與寫入指令則瞄準該使用者記憶體。於指令由該模擬器記憶體中擷取時，該模擬器晶片內所包括的邏輯則將讀出與寫入記憶體存取至該使用者記憶體。

英文發明摘要（發明之名稱：

METHOD AND SYSTEM FOR UPDATING USER
MEMORY IN EMULATOR SYSTEMS

A device, system and method for providing access to user memory in emulator systems. The emulator system contains an emulator system memory, a user system memory and an emulator device. The emulator device operates in a mode where program execution instructions originate in the emulation memory while read and write instructions target the user memory. Logic included in the emulator chip directs the read and write memory accesses to the user memory while instructions are fetched from the emulator memory.

（請先閱讀背面之注意事項再填寫本頁名稱）

裝

訂

線

六、申請專利範圍

申請專利範圍：

1. 一種模擬器裝置，包括：

一記憶體介面，用來存取程式記憶體，其中該程式記憶體包括一第一記憶體、與一外接至該裝置的第二記憶體；以及

一選擇電路，其係被連接至介面，以於該裝置係配置以從該第一記憶體擷取指令時，將該程式記憶體寫入與記憶體讀出存取只導入至該第二記憶體。

2. 如申請專利範圍第1項之裝置，包括：

一連接至該選擇電路的電路，以偵測是否該表區(table)讀出與表區寫入存取之至少一者即將被執行；

其中該選擇電路僅將該表區讀出與表區寫入存取之至少一者只導入至該第二記憶體。

3. 如申請專利範圍第1項之裝置，包括：

一模式選擇電路；

其中該選擇電路包括一切換裝置，其係連接至該第一與第二記憶體，並且係被連接以接收一由該模式選擇電路所輸出之訊號。

4. 如申請專利範圍第1項之裝置，包括：

一指令解碼器，其輸出一訊號，代表程式記憶體讀出存取指令與程式記憶體寫入存取指令之至少一者即將被解碼；以及

一被連接至該解碼器的電路，其係被配置以接收該訊號並且被配置以執行程式記憶體讀出存取指令與程式

(請先閱讀背面之注意事項再填寫本頁)

訂
線

六、申請專利範圍

記憶體寫入存取指令之至少一者。

5. 如申請專利範圍第1項之裝置，包括：

一模式選擇電路；

一邏輯電路，其係被連接以接收該模式選擇電路之輸出；以及

一指令解碼器，其具有一連接至該邏輯電路的輸出端；其中

該介面電路係連接至該邏輯電路的輸出端。

6. 如申請專利範圍第1項之裝置，其中：

該模式選擇電路包括一第一構件，其用來輸出一代表該裝置之一操作模式的訊號；

該指令解碼器包括一第二構件，其用來輸出一代表程式記憶體讀出存取指令與程式記憶體寫入存取指令之至少一者即將被解碼的訊號；以及

該邏輯電路係被連接以接收由該第一與第二構件所輸出之訊號，並且輸出一代表對於該第一與第二記憶體存取之何者即將被致能的訊號至該選擇電路。

7. 如申請專利範圍第1項之裝置，其中：

該記憶體介面包括：

一程式記憶體匯流排，以及

一連接至該匯流排之程式記憶體匯流排控制器；

以及

該選擇電路包括：

一多工器，其係連接至該程式記憶體匯流排、一

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

六、申請專利範圍

第一記憶體存取匯流排以及一第二記憶體存取匯流排；以及

一電路，其係連接至該多工器，以在該第一與第二記憶體存取匯流排之間作選擇。

8. 如申請專利範圍第7項之裝置，其中該電路包括一構件，其用來產生一輸出至該多工器的訊號，以於該裝置被配置以從該第一記憶體中擷取指令時，表示僅有存取至該第二記憶體。

9. 如申請專利範圍第8項之裝置，其中該構件包括：

一模式選擇電路；

一電路，其產生一代表程式記憶體存取即將被執行的訊號；以及

一第一邏輯電路，其係被連接以接收該模式選擇電路之輸出，並具有一輸入端，其係被連接以接收由該電路所輸出之訊號。

10. 如申請專利範圍第1項之裝置，其中：

該第一記憶體係為一模擬器程式記憶體；以及

該第二記憶體係為一使用者程式記憶體。

11. 如申請專利範圍第1項之裝置，更包括：

一模擬器系統，其係被連接至該裝置；以及

一使用者系統，其係被連接至該裝置。

12. 如申請專利範圍第11項之裝置，其中：

該模擬器系統包括該第一記憶體；以及

該使用者系統包括該第二記憶體。

六、申請專利範圍

13. 如申請專利範圍第12項之裝置，其中：

該第一記憶體包括一模擬器程式記憶體，其包含即將被該裝置所擷取之指令；以及

該第二記憶體包括一使用者程式記憶體，其中於該裝置被配置以從該第一記憶體中擷取指令時，只有該程式記憶體寫入記憶體讀出存取被導入該使用者程式記憶體。

14. 一種模擬器裝置，包括：

一構件，係用來接收產生自一模擬器記憶體而傳送至該裝置的指令；以及

一構件，係被連接至該用來接收的構件，而於該裝置係配置以從該模擬記憶體擷取指令時，用來只瞄準記憶體讀出與寫入指令至連接至該裝置的使用者記憶體。

15. 如申請專利範圍第14項之裝置，包括：

一構件，係用來偵測記憶體讀出與寫入指令，其係連接至該用來接收之構件；以及

一構件，係用來選擇該裝置之操作模式，其係連接至該用來瞄準之構件以及該用來偵測之構件。

16. 如申請專利範圍第14項之裝置，其中該用來瞄準之構件包括：

一構件，係用來偵測該裝置之操作模式；

一構件，係用來偵測記憶體讀出與寫入指令；以及

一構件，係使用兩個用來偵測的構件之輸出端而

六、申請專利範圍

用來在該模擬器記憶體以及該使用者記憶體之間作選擇。

17. 如申請專利範圍第16項之裝置，包括：

一構件，係在該用來選擇的構件之控制下，用來在存取至該模擬器記憶體以及該使用者記憶體之間作切換。

18. 一種操作模擬器裝置的方法，包括以下之步驟：

僅從一第一記憶體中擷取指令；以及

僅對與該第一記憶體分離而外接至該模擬器裝置之一第二記憶體導入記憶體存取。

19. 如申請專利範圍第18項之方法，包括以下之步驟：

僅從一模擬程式記憶體中擷取指令；以及

僅對與該模擬程式記憶體分離而外接至該模擬器裝置之一使用者程式記憶體導入記憶體存取。

20. 如申請專利範圍第19項之方法，包括以下之步驟：

將表區讀出與表區寫入存取之至少一者導入該程式記憶體。

21. 如申請專利範圍第18項之方法，包括以下之步驟：

偵測該裝置之操作模式；

偵測一記憶體存取是否即將被執行；以及

根據該偵測步驟，選擇該第一記憶體與該第二記憶體之間的存取。

六、申請專利範圍

22. 如申請專利範圍第21項之方法，其中：

偵測一記憶體存取是否即將被執行的步驟係包括偵測是否表區讀出與表區寫入存取之至少一者即將被執行；以及

偵測該記憶體存取的步驟包括將表區讀出與表區寫入存取之至少一者導入至該第二記憶體。

23. 如申請專利範圍第22項之方法，包括以下之步驟：

僅從一模擬程式記憶體中擷取指令；以及

僅對與該模擬程式記憶體分離而外接至該模擬器裝置之一使用者程式記憶體導入記憶體存取。

24. 如申請專利範圍第18項之方法，包括以下之步驟：

將指令解碼；

使用該解碼步驟，偵測一記憶體存取是否即將被執行；以及

使用該偵測步驟，決定該第一與第二記憶體之何者將被存取。

25. 如申請專利範圍第24項之方法，包括以下之步驟：

偵測該裝置之操作模式；以及

使用該偵測步驟，決定該第一與第二記憶體之何者將被存取。

26. 如申請專利範圍第25項之方法，包括以下之步

六、申請專利範圍

驟：

僅對與該模擬程式記憶體分離而外接至該模擬器裝置之一使用者程式記憶體導入記憶體存取。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

圖 1

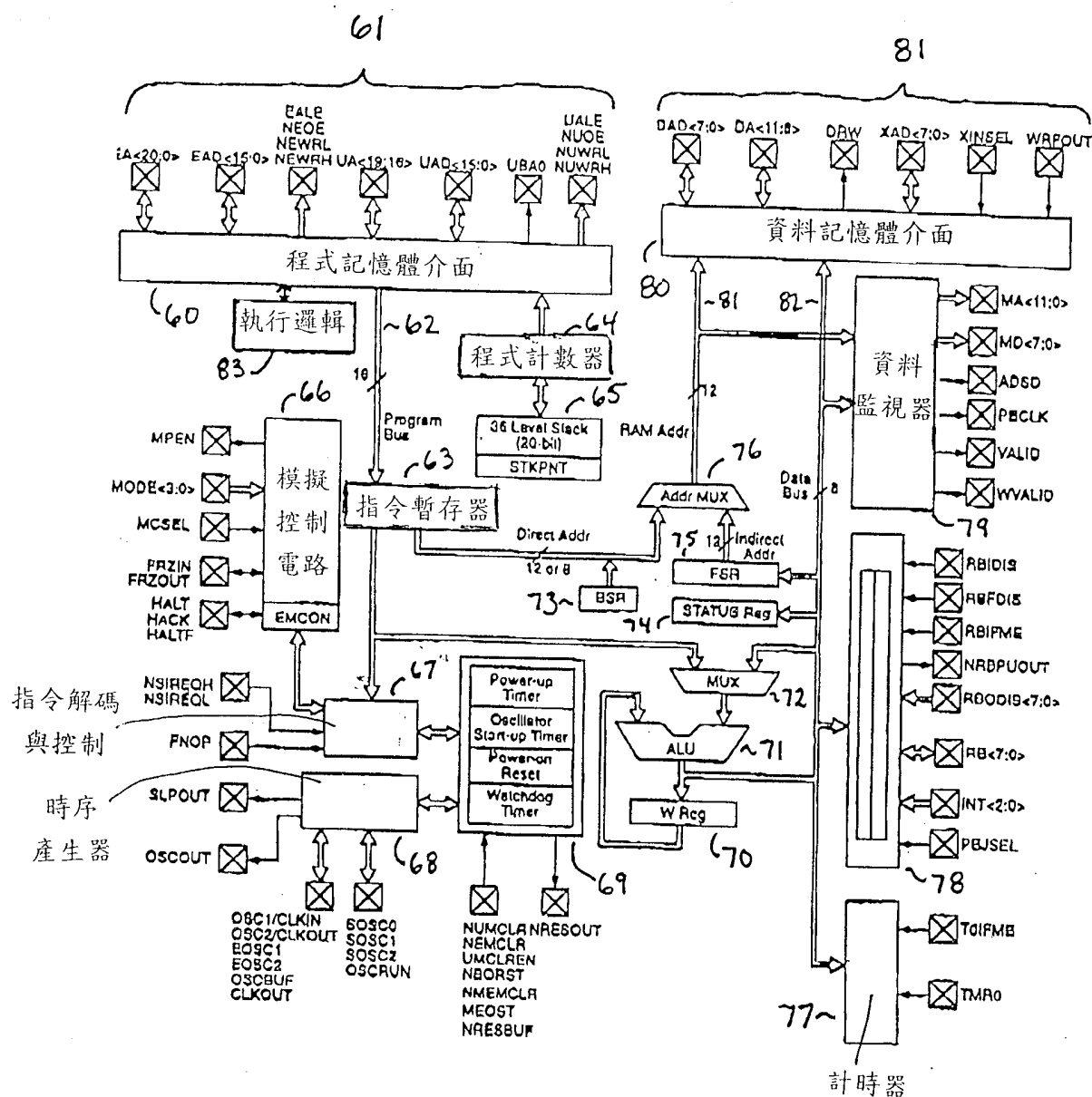


圖 2



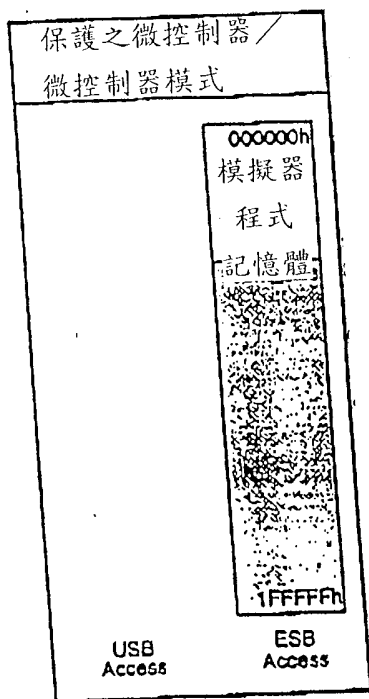


圖 4A

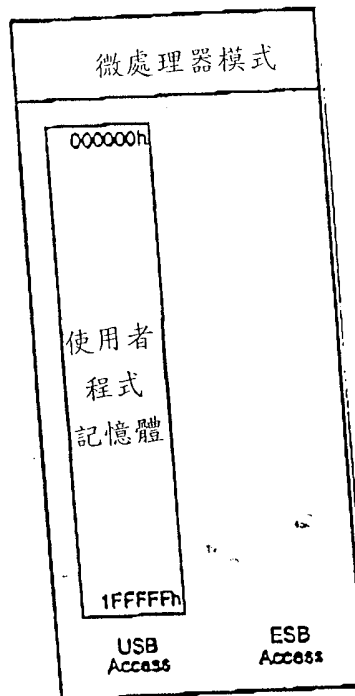


圖 4B

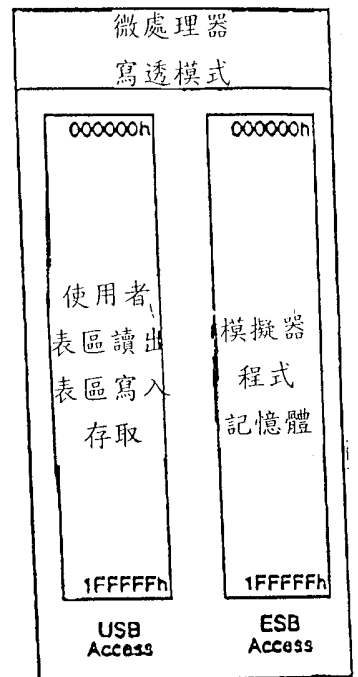


圖 4C

OBLON ET AL (703) 413-3000
DOCKET # 7452-0012-2
SHEET 5 OF 6

TBLRD Table Read

Syntax: [*label*] TBLRD (":", "+", "-", "-")

Operands: None

Operation: if TBLRD ":",
 (Prog Mem (TBLPTR)) → TABLAT;
 TBLPTR - No Change;
 if TBLRD "+",
 (Prog Mem (TBLPTR)) → TABLAT;
 (TBLPTR) + 1 → TBLPTR;
 if TBLRD "-",
 (Prog Mem (TBLPTR)) → TABLAT;
 (TBLPTR) - 1 → TBLPTR;
 if TBLRD "+",
 (TBLPTR) + 1 → TBLPTR;
 (Prog Mem (TBLPTR)) → TABLAT;

Status Affected: None

Encoding:

0000	0000	0000	10nn nn=0 ~ =1 ~+ =2 ~- =3 ~+
------	------	------	---

Description: There are four options with a TBLRD instruction to determine what happens to the 21-bit Table Pointer (TBLPTR): no change, post-increment, post-decrement and pre-increment. The current option is determined and the TBLPTR is modified appropriately, and the contents of the program memory location pointed to by the TBLPTR are loaded into the 8-bit Table Latch (TABLAT). The LSB of the TBLPTR selects which byte of the program memory location will be read. If LSB = 1, the high byte will be loaded into the TABLAT. If LSB = 0, the low byte will be loaded into the TABLAT.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Table Pointer on Address bus)	No operation	No operation (OE goes low) TABLAT updated

TBLRD Table Read

Example1: TBLRD "+";

Before Instruction

TABLAT	=	0x55
TBLPTR	=	0x00A358
MEMORY(0x00A358)	=	0x34

After Instruction

TABLAT	=	0x34
TBLPTR	=	0x00A357

Example2: TBLRD "-";

Before Instruction

TABLAT	=	0xAA
TBLPTR	=	0x01A357
MEMORY(0x01A357)	=	0x12
MEMORY(0x01A358)	=	0x34

After Instruction

TABLAT	=	0x34
TBLPTR	=	0x01A358

圖 5

OBLON ET AL (703) 413-30
DOCKET # 7452-0012-2
SHEET 6 OF 6

TBLWT	Table Write
Syntax:	{label} TBLWT (":", "+", "-", "+")
Operands:	None
Operation:	if TBLWT ":", (TABLAT) → Prog Mem(TBLPTR); TBLPTR - No Change; if TBLWT "+", (TABLAT) → Prog Mem(TBLPTR); (TBLPTR) +1 → TBLPTR; if TBLWT "-", (TABLAT) → Prog Mem(TBLPTR); (TBLPTR) -1 → TBLPTR; if TBLWT "+", (TABLAT) → Prog Mem(TBLPTR); (TBLPTR) +1 → TBLPTR; (TABLAT) → Prog Mem(TBLPTR);

Status Affected: None

Encoding:

0000	0000	0000	11nn
			nn=0 *
			=1 *-
			=2 *-
			=3 *-

Description:

There are four options with a TBLWT instruction. These options determine what happens to the Table Pointer (TBLPTR): no change, post-increment, post-decrement and pre-increment. The current option is determined and the TBLPTR is modified appropriately.

The contents of Table Latch (TABLAT) are written to the program memory location pointed to by TBLPTR.

If TBLPTR points to an external program memory location, then the instruction executes in two cycles.

Since the TABLAT is only one byte wide, a multiple of two TBLWT instructions must be executed to program internal memory locations. For example, if the device is defined to program one word at a time, an internal memory location is programmed in the following manner:

- 1) Set TBLPTR to an even byte
- 2) Write low byte to TABLAT
- 3) Execute TBLWT "+ (2-cycle)
- 4) Write high byte to TABLAT
- 5) Execute TBLWT "+ (long write)

A long write to an internal EPROM location is terminated when an interrupt is received. The post increment TBLWT instruction is the only TBLWT instruction that is recommended for writes to internal memory. (Writes to internal EPROM are only available on devices with 64 or more pins.)

Words:

1

TBLWT	Table Write
Cycles:	2 (many if long write is to on-chip EPROM program memory)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Table Pointer on Address bus)	No operation	No operation (Table Latch on Address bus, WR goes low)

Example 1: TBLWT "+";

Before Instruction

TABLAT = 0x55
TBLPTR = 0x00A356
MEMORY(0x00A356) = 0xFF

After Instructions (table write completion)

TABLAT = 0x55
TBLPTR = 0x00A357
MEMORY(0x00A356) = 0x55

Example 2: TBLWT "-";

Before Instruction

TABLAT = 0x34
TBLPTR = 0x01369A
MEMORY(0x01369A) = 0xFF
MEMORY(0x01369B) = 0xFF

After Instruction (table write completion)

TABLAT = 0x34
TBLPTR = 0x01369B
MEMORY(0x01369A) = 0xFF
MEMORY(0x01369B) = 0x34

圖 6