



(21) 申請案號：106119320

(22) 申請日：中華民國 106 (2017) 年 06 月 09 日

(51) Int. Cl. : **G06F9/455 (2018.01)**(71) 申請人：國立交通大學 (中華民國) NATIONAL CHIAO TUNG UNIVERSITY (TW)
新竹市大學路 1001 號

(72) 發明人：楊武 YANG, WUU (TW)；林宗俊 LIN, TSUNG CHUN (TW)

(74) 代理人：李世章；秦建譜

申請實體審查：有 申請專利範圍項數：10 項 圖式數：6 共 22 頁

(54) 名稱

二元碼轉譯裝置及方法

BINARY TRANSLATION DEVICE AND METHOD

(57) 摘要

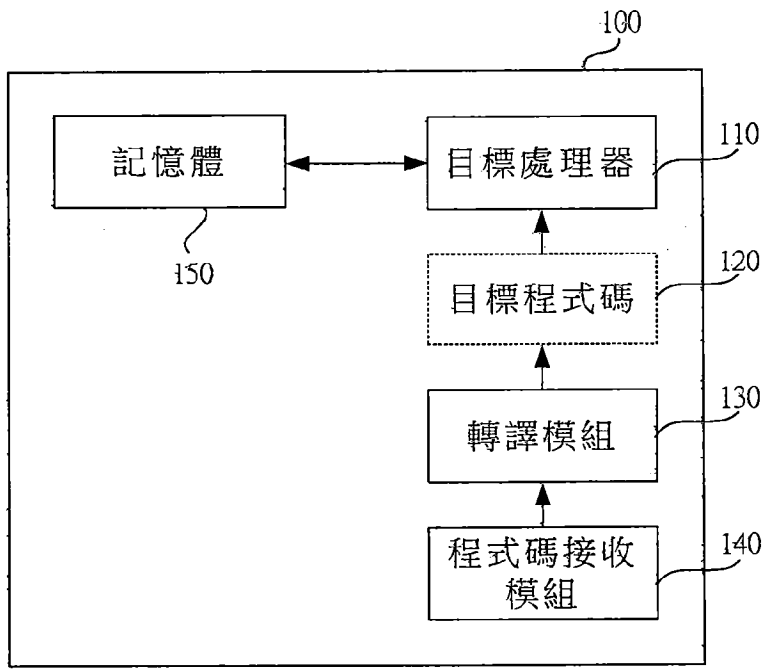
一種二元碼轉譯裝置，包含轉譯模組、記憶體及處理器。轉譯模組用以轉譯第二程式碼至第一程式碼。處理器用以執行轉譯模組已轉譯的第一程式碼。第一程式碼暫存於該記憶體中並具有程式執行位址，至少一函式庫暫存於記憶體中，至少一函式庫包含第一函式以及第二函式，且第一函式與第二函式分別位於記憶體的第一函式位址以及第二函式位址。當處理器仍在執行第一程式碼並呼叫第一函式時，利用轉譯模組轉譯第一程式碼得到接續第一函式之第二函式，且處理器取得第二函式位於記憶體中之第二函式位址。當第一程式碼呼叫第一函式並執行完成時，處理器不返回程式執行位置並直接執行位於第二函式位址上的第二函式。

A binary translation device includes a translation module, a memory and a processor. The translation module is configured to translate a second code to a first code. The processor is configured to execute the first code translated by the translation module. The first code is temporarily stored in the memory and has a program execution address, a library including at least one function is temporarily stored in the memory, the library includes a first function and a second function, and the first function and the second function are located at a first function address and a second function address in the memory. When the processor is still executing the first code and calls the first function, the first code translated by the translation module is utilized to acquire a second function subsequent to the first function, and the processor acquires the second function address located in the memory. While the first code calls the first function and the first function has been executed completely, the processor directly executes the second function located at the second function address without returning to the program execution address.

指定代表圖：

符號簡單說明：

- 100 . . . 目標裝置
- 110 . . . 目標處理器
- 130 . . . 轉譯模組
- 140 . . . 程式碼接收
模組
- 120 . . . 目標程式碼
- 150 . . . 記憶體



第2圖

201903606

106119320

申請案號：

【發明摘要】

申請日： 106/06/09

IPC分類： **G06F 9/45** (2006.01)

【中文發明名稱】 二元碼轉譯裝置及方法

【英文發明名稱】 BINARY TRANSLATION DEVICE
AND METHOD

【中文】

一種二元碼轉譯裝置，包含轉譯模組、記憶體及處理器。轉譯模組用以轉譯第二程式碼至第一程式碼。處理器用以執行轉譯模組已轉譯的第一程式碼。第一程式碼暫存於該記憶體中並具有程式執行位址，至少一函式庫暫存於記憶體中，至少一函式庫包含第一函式以及第二函式，且第一函式與第二函式分別位於記憶體的第一函式位址以及第二函式位址。當處理器仍在執行第一程式碼並呼叫第一函式時，利用轉譯模組轉譯第一程式碼得到接續第一函式之第二函式，且處理器取得第二函式位於記憶體中之第二函式位址。當第一程式碼呼叫第一函式並執行完成時，處理器不返回程式執行位置並直接執行位於第二函式位址上的第二函式。

【英文】

A binary translation device includes a translation module, a memory and a processor. The translation module is configured to translate a second code to a first code. The processor is

configured to execute the first code translated by the translation module. The first code is temporarily stored in the memory and has a program execution address, a library including at least one function is temporarily stored in the memory, the library includes a first function and a second function, and the first function and the second function are located at a first function address and a second function address in the memory. When the processor is still executing the first code and calls the first function, the first code translated by the translation module is utilized to acquire a second function subsequent to the first function, and the processor acquires the second function address located in the memory. While the first code calls the first function and the first function has been executed completely, the processor directly executes the second function located at the second function address without returning to the program execution address.

【指定代表圖】 第2圖

【代表圖之符號簡單說明】

100：目標裝置

130：轉譯模組

110：目標處理器

140：程式碼接收模組

120：目標程式碼

150：記憶體

【特徵化學式】無

【發明說明書】

【中文發明名稱】 二元碼轉譯裝置及方法

【英文發明名稱】 BINARY TRANSLATION DEVICE
AND METHOD

【技術領域】

【0001】 本揭示文件是關於二元碼轉譯裝置及方法，更明確而言是關於動態二元碼轉譯裝置及方法。

【先前技術】

【0002】 隨著雲端運算的蓬勃發展，雲端資料會在不同平台間存取應用。由於不同平台間可能會使用不同處理器，雲端資料會在不同平台間轉換，跨平台虛擬化的技術應運而生，成為雲端技術的重要議題。動態二元碼轉譯(Dynamic Binary Translation)為跨平台虛擬化之核心技術。在動態二元碼轉譯的過程，因函式返回所導致的間接跳躍(indirect branch)，可能會導致處理速度降低以致整體系統效能不佳的問題。

【發明內容】

【0003】 本揭示內容提供一種二元碼轉譯裝置，包含轉譯模組、記憶體及處理器。轉譯模組用以轉譯第二程式碼至第一程式碼。處理器用以執行轉譯模組已轉譯的第一程式碼。第一程式碼暫存於記憶體中並具有程式執行位址，至少一函

式庫暫存於記憶體中，至少一函式庫包含第一函式以及第二函式，且第一函式與第二函式分別位於記憶體的第一函式位址以及第二函式位址。當處理器仍在執行第一程式碼並呼叫第一函式時，利用轉譯模組轉譯第一程式碼得到接續第一函式之第二函式，且處理器取得第二函式位於記憶體中之第二函式位址。當第一程式碼呼叫第一函式並執行完成時，處理器不返回程式執行位置並直接執行位於第二函式位址上的第二函式。

【0004】 本揭示內容亦提供一種二元碼轉譯方法，包含下列步驟。轉譯第二程式碼至第一程式碼，其中第一程式碼暫存於記憶體中並具有程式執行位址，至少一函式庫暫存於記憶體中，至少一函式庫包含第一函式以及第二函式，且第一函式與第二函式分別位於記憶體的第一函式位址以及第二函式位址。由處理器執行已轉譯的第一程式碼。當處理器仍在執行第一程式碼並呼叫第一函式時，轉譯第一程式碼得到接續第一函式之第二函式，且處理器取得第二函式位於記憶體中之第二函式位址。當第一程式碼呼叫第一函式並執行完成時，處理器不返回程式執行位址並直接執行位於第二函式位址上的第二函式。

【0005】 本揭示內容更提供一種二元碼轉譯裝置，包含處理器及記憶體。在處理器執行將變數儲存至記憶體之第一位址之第一指令且變數關聯於記憶體之位址時，若第一指令之後存在將變數儲存至記憶體之第二位址之第二指令且在第一指令及第二指令之間存在與變數無關之若干指令，則處理

器只執行第一指令。

【0006】 本揭示內容亦提供一種二元碼轉譯裝置，包含處理器及記憶體。在處理器執行從記憶體存取變數至第一暫存器之第一指令且變數關聯於記憶體之位址時，若第一指令之後存在從記憶體存取變數至第二暫存器之第二指令時且在第一指令及第二指令之間存在與變數無關之若干指令，則處理器只執行第二指令。

【圖式簡單說明】

【0007】

第1圖係依照本揭示文件一實施例繪示之二元碼於不同裝置間轉譯之示意圖。

第2圖係依照本揭示文件一實施例繪示之二元碼在目標裝置中轉譯及執行之示意圖。

第3圖係依照本揭示文件一實施例繪示之目標程式碼執行函式之示意圖。

第4圖係依照本揭示文件一實施例繪示之二元碼在目標裝置中轉譯及執行的方法之流程圖。

第5圖係依照本揭示文件另一實施例繪示之二元碼在目標裝置中轉譯及執行的方法之流程圖。

第6圖係依照本揭示文件又一實施例繪示之二元碼在目標裝置中轉譯及執行的方法之流程圖。

【實施方式】

【0008】 第1圖係依照本揭示文件一實施例繪示之二元碼於不同裝置間轉譯之示意圖。參考第1圖，來源裝置200包含來源處理器210，來源程式碼220是適於在來源處理器210處理之程式碼。繼續參考第1圖，於一實施例中，目標裝置100利用程式碼接收模組140接收來自來源裝置200之來源程式碼220。一般而言，目標裝置100與來源裝置200會是不同的系統。在目標裝置100中，使用轉譯模組130轉譯程式碼接收模組140中的來源程式碼220以得到適於在目標處理器110處理之目標程式碼120。

【0009】 在一實施例中，目標處理器110可為ARM、SoC、MIPS、x86、DSP或PowerPC。

【0010】 在一實施例中，來源處理器210可為ARM、SoC、MIPS、x86、DSP或PowerPC。

【0011】 第2圖係依照本揭示文件一實施例繪示之二元碼在目標裝置100中轉譯及執行之示意圖。目標裝置100包含轉譯模組130、記憶體150及目標處理器110。轉譯模組130用以轉譯來源程式碼220至目標程式碼120。目標處理器110用以執行轉譯模組130已轉譯的目標程式碼120。

【0012】 第3圖係依照本揭示文件一實施例繪示之目標程式碼執行函式之示意圖。目標程式碼120是適於在目標處理器110之程式碼，且來源程式碼220是適於在來源處理器210之程式碼。在來源程式碼220中在呼叫第一來源函式FunS1()後，隨即呼叫第二來源函式FunS2()。第一來源函式FunS1()及第二來源函式FunS2()均存在於來源函式庫(圖

中未示)中。同樣地，從來源程式碼220轉譯後的目標程式碼120中亦存在第一目標函式FunD1()及第二目標函式FunD2()，在目標程式碼120中在呼叫第一目標函式FunD1()後，隨即呼叫第二目標函式FunD2()。第一目標函式FunD1()及第二目標函式FunD2()均存在於來源函式庫160。

【0013】 第4圖係依照本揭示文件一實施例繪示之二元碼在目標裝置中轉譯及執行的方法之流程圖。同時參考第2、3及4圖。在步驟S410，首先，由轉譯模組130轉譯來源程式碼220至目標程式碼120。

【0014】 目標程式碼120暫存於記憶體150中並具有程式執行位址。程式執行位址即第一目標函式FunD1()之返回位址。一般而言，返回位址會暫存於連接暫存器(link register)中，以使得在函式執行完成後，跳躍回原來的程式執行位址而繼續執行程式碼。來源函式庫160暫存於記憶體150中且包含若干函式，其中來源函式庫160包含第一目標函式FunD1()及第二目標函式FunD2()。第一目標函式FunD1()及第二目標函式FunD2()位於記憶體150的第一目標函式位址以及第二目標函式位址。

【0015】 在步驟S430，目標處理器110用以執行轉譯模組130已轉譯的目標程式碼120。

【0016】 一般而言，在靜態二元碼轉譯時，由於程式碼是完整轉譯的，因此執行第一目標函式FunD1()之後跳躍至第二目標函式FunD2()的第二目標函式位址是已知的。因

此，可輕易得知第二目標函式位址而跳躍。然而，在動態二元碼轉譯時，因為程式碼並非完全轉譯的，而是在執行一個區段時同時轉譯下一個區段，所以第二目標函式位址是未知的。

【0017】 因此，在步驟S450，本揭示內容提供能夠提升動態二元碼轉譯處理效益的機制，當目標處理器110仍在執行目標程式碼120並呼叫第一目標函式FunD1()時，轉譯目標程式碼120得到接續第一目標函式FunD1()之第二目標函式FunD2()，且目標處理器110取得第二目標函式FunD2()位於記憶體150中之第二目標函式位址。

【0018】 最後，在步驟S470，當目標程式碼120呼叫第一目標函式FunD1()並執行完成時，目標處理器120不返回程式執行位址並直接執行位於第二目標函式位址上的第二目標函式FunD2()。

【0019】 依照上述步驟，由於在第一目標函式FunD1()返回前，目標處理器110已從來源程式碼220轉譯出第二目標函式位址，藉由第二目標函式位址取代程式執行位址(即返回位址)而直接執行接續於第一目標函式FunD1()的第二目標函式FunD2()，可使得動態二元碼轉譯更有效率。

【0020】 第5圖係依照本揭示文件另一實施例繪示之二元碼在目標裝置中轉譯及執行的方法之流程圖。依照以下步驟，可消除程式碼中對記憶體的冗餘存取。

【0021】 同時參考第2及5圖。首先，在步驟S510，目標處理器110執行以來源程式碼220轉譯之目標程式碼120。在步驟

S530，目標處理器110執行將變數X0儲存至記憶體150之第一位址之第一指令且變數X0關聯於記憶體150之位址。步驟S550中，判斷該第一指令之後是否存在將變數X0儲存至記憶體150之第二位址之第二指令，且判斷在第一指令及第二指令之間是否存在與變數無關之若干指令。若是，則進入步驟S570，目標處理器110只執行該第一指令。若否，則回到步驟S510，繼續執行目標程式碼120。

【0022】 同時參考第2圖及第5圖及以下指令碼。以下指令碼是根據第5圖之步驟之儲存資料到記憶體的指令之實例。記憶體位址位於變數X0中，執行時才可得知此明確位址。在兩個對相同的對暫存器模擬的全域變數(global variable)的存取指令間，第二指令(為存取指令)是無法消除的。由於在靜態二元碼轉譯過程中，無法得知儲存資料到記憶體的指令之位址是否與對暫存器模擬的全域變數其位址相同。然而，在動態二元碼轉譯過程中，記憶體位址是於變數X0當中之儲存資料到記憶體的指令，此位址是自來源記憶體之操作指令轉譯過來，其記憶體指令操作範圍為來源二元碼映像(source binary image)或為二元碼轉譯系統所提供的模擬堆疊(emulated stack)，然而對於對暫存器模擬的全域變數其記憶體位址是在二元碼轉譯系統本身，兩者記憶體區塊是不同的，因此可將第二指令消除，即步驟S570中，目標處理器110只執行該第一指令。

```
%2 = load i64* @X0
store i64 %0, i64* %1
%3 = load i64* @X0
```

【0023】 同時參考第2圖及第5圖及以下程式碼。以下程式碼是根據第5圖之步驟之儲存資料到記憶體之指令之實例。記憶體位址是位於變數X0中，在兩個對相同的對暫存器模擬的全域變數的儲存與存取指令中間，對暫存器模擬的全域變數的存取指令(即第二指令)可被消除(步驟S570)，亦即直接拿取對暫存器模擬的全域變數的儲存指令的資料使用。

```
store i64 %2, i64* @X0
store i64 %0, i64* %1
%3 = load i64* @X0
```

【0024】 第6圖係依照本揭示文件又一實施例繪示之二元碼在目標裝置中轉譯及執行的方法之流程圖。依照以下步驟，可消除程式碼中對記憶體的冗餘存取。

【0025】 同時參考第2圖及第6圖。首先，在步驟S610，目標處理器110執行以來源程式碼220轉譯之目標程式碼120。在步驟S630，目標處理器110執行從記憶體150存取變數X0至第一暫存器之第一指令且變數X0關聯於記憶體150之位址。在步驟S650，判斷第一指令之後是否存在從記憶體150存取該變數X0至第二暫存器之第二指令時，且判斷在第一指令及第二指令之間是否存在與變數X0無關之若干指令。若是，則進入步驟S670，目標處理器110只執行該第二指令。若否，則回到步驟S610，繼續執行目標程式碼120。

【0026】 同時參考第2圖及第6圖及以下程式碼。以下程式碼是根據第6圖之步驟之從記憶體存取資料的指令之實例。記憶體位址是位於在變數X0中，在兩個對相同的對暫存器模擬

的全域變數之儲存指令中間，可將第一儲存指令(即第一指令)消除。即步驟S670，目標處理器110只執行該第二指令。

```
store i64 %1, i64* @X0
%3 = load i64* %0
store i64 %2, i64* @X0
```

【0027】 雖然上文實施方式中揭露了本揭示文件的具體實施例，然其並非用以限定本揭示文件，本揭示文件所屬技術領域中具有通常知識者，在不悖離本揭示文件之原理與精神的情形下，當可對其進行各種更動與修飾，因此本揭示文件之保護範圍當以附隨申請專利範圍所界定者為準。

【符號說明】

【0028】

100：目標裝置	220：來源程式碼
110：目標處理器	FunS1：第一來源函式
120：目標程式碼	FunS2：第二來源函式
130：轉譯模組	FunD1：第一目標函式
140：程式碼接收模組	FunD2：第二目標函式
150：記憶體	S410至S470：步驟
160：目標函式庫	S510至S570：步驟
200：來源裝置	S610至S670：步驟
210：來源處理器	X0：變數

【發明申請專利範圍】

【第1項】 一種二元碼轉譯裝置，包含：

一轉譯模組，用以轉譯一第二程式碼至一第一程式碼；

一記憶體；及

一處理器，用以執行該轉譯模組已轉譯的該第一程式碼；

其中該第一程式碼暫存於該記憶體中並具有一程式執行位址，至少一函式庫暫存於該記憶體中，該至少一函式庫包含一第一函式以及一第二函式，且該第一函式與該第二函式分別位於該記憶體的一第一函式位址以及一第二函式位址；

其中當該處理器仍在執行該第一程式碼並呼叫該第一函式時，利用該轉譯模組轉譯之該第一程式碼得到接續該第一函式之該第二函式，且該處理器取得該第二函式位於該記憶體中之該第二函式位址；且

當該第一程式碼呼叫該第一函式並執行完成時，該處理器不返回該程式執行位址並直接執行位於該第二函式位址上的該第二函式。

【第2項】 如請求項 1 所述之裝置，其中該第二程式碼可在一第二處理器上執行，該第二處理器與該處理器為不同類型的處理器。

【第3項】 如請求項 1 所述之裝置，其中在該轉譯模組

轉譯該第二程式碼為該第一程式碼完成後，該處理器才執行該第一程式碼。

【第4項】 如請求項 1 所述之裝置，其中在該轉譯模組在轉譯該第二程式碼為該第一程式碼尚未完成前，該處理器即執行該第一程式碼。

【第5項】 一種二元碼轉譯方法，包含：

轉譯一第二程式碼至一第一程式碼，其中該第一程式碼暫存於一記憶體中並具有一程式執行位址，至少一函式庫暫存於該記憶體中，該至少一函式庫包含一第一函式以及一第二函式，且該第一函式與該第二函式分別位於該記憶體的一第一函式位址以及一第二函式位址；

由一處理器執行已轉譯的該第一程式碼；

當該處理器仍在執行該第一程式碼並呼叫該第一函式時，轉譯該第一程式碼得到接續該第一函式之該第二函式，且該處理器取得該第二函式位於該記憶體中之該第二函式位址；且

當該第一程式碼呼叫該第一函式並執行完成時，該處理器不返回該程式執行位址並直接執行位於該第二函式位址上的該第二函式。

【第6項】 如請求項 5 所述之方法，其中該第二程式碼可在一第二處理器上執行，該第二處理器與該處理器為不同類型的處理器。

【第7項】 如請求項 5 所述之方法，其中在轉譯該第二程式碼為該第一程式碼完成後，該處理器才執行該第一程式碼。

【第8項】 如請求項 5 所述之方法，其中在轉譯該第二程式碼為該第一程式碼尚未完成前，該處理器即執行該第一程式碼。

【第9項】 一種二元碼轉譯裝置，包含：

一處理器；及

一記憶體，

其中在該處理器執行將一變數儲存至該記憶體之一第一位址之一第一指令且該變數關聯於該記憶體之一位址時，若該第一指令之後存在將該變數儲存至該記憶體之一第二位址之一第二指令且在該第一指令及該第二指令之間存在與該變數無關之若干指令，則該處理器只執行該第一指令。

【第10項】 一種二元碼轉譯裝置，包含：

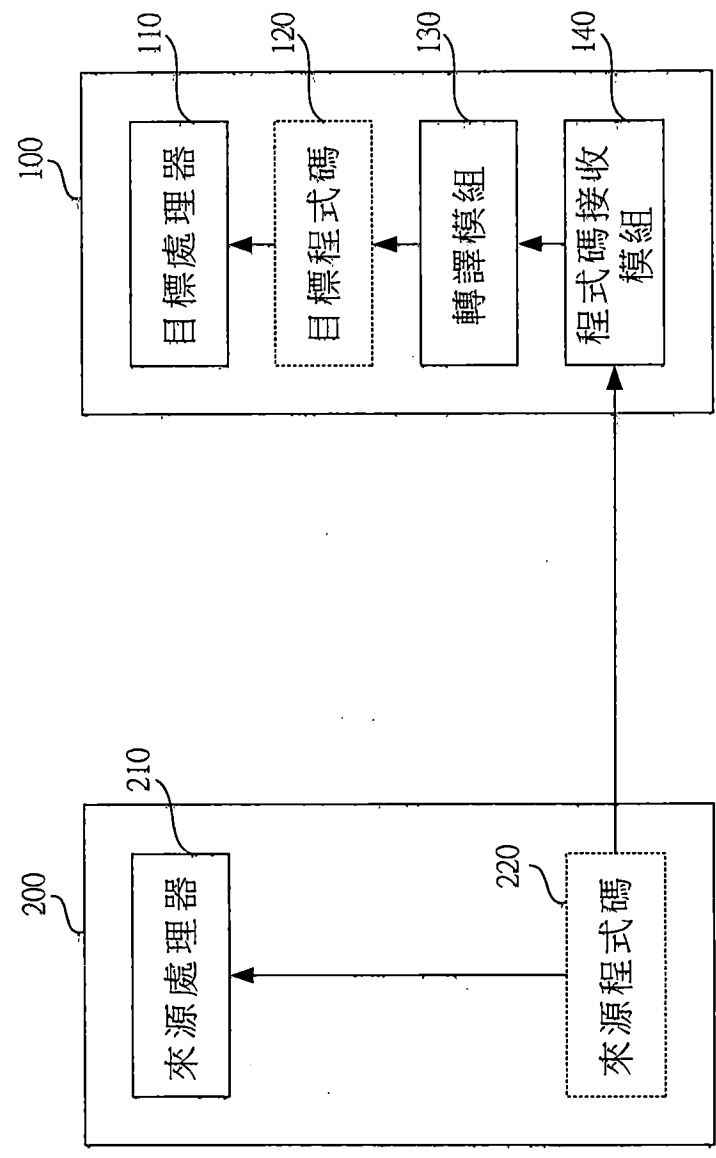
一處理器；及

一記憶體，

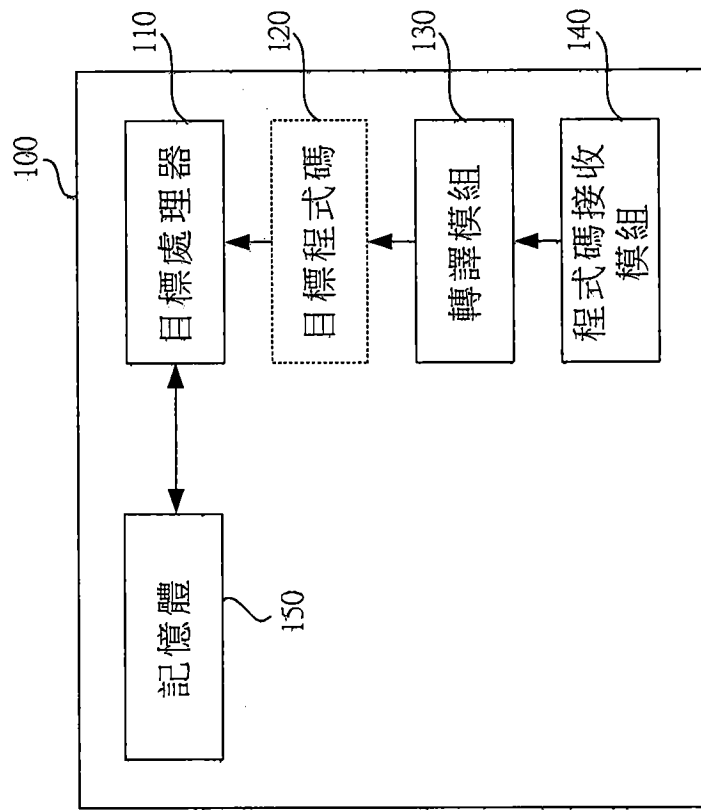
其中在該處理器執行從該記憶體存取一變數至一第一暫存器之一第一指令且該變數關聯於該記憶體之一位址時，若該第一指令之後存在從該記憶體存取該變數至一第

二暫存器之一第二指令時且在該第一指令及該第二指令之間存在與該變數無關之若干指令，則該處理器只執行該第二指令。

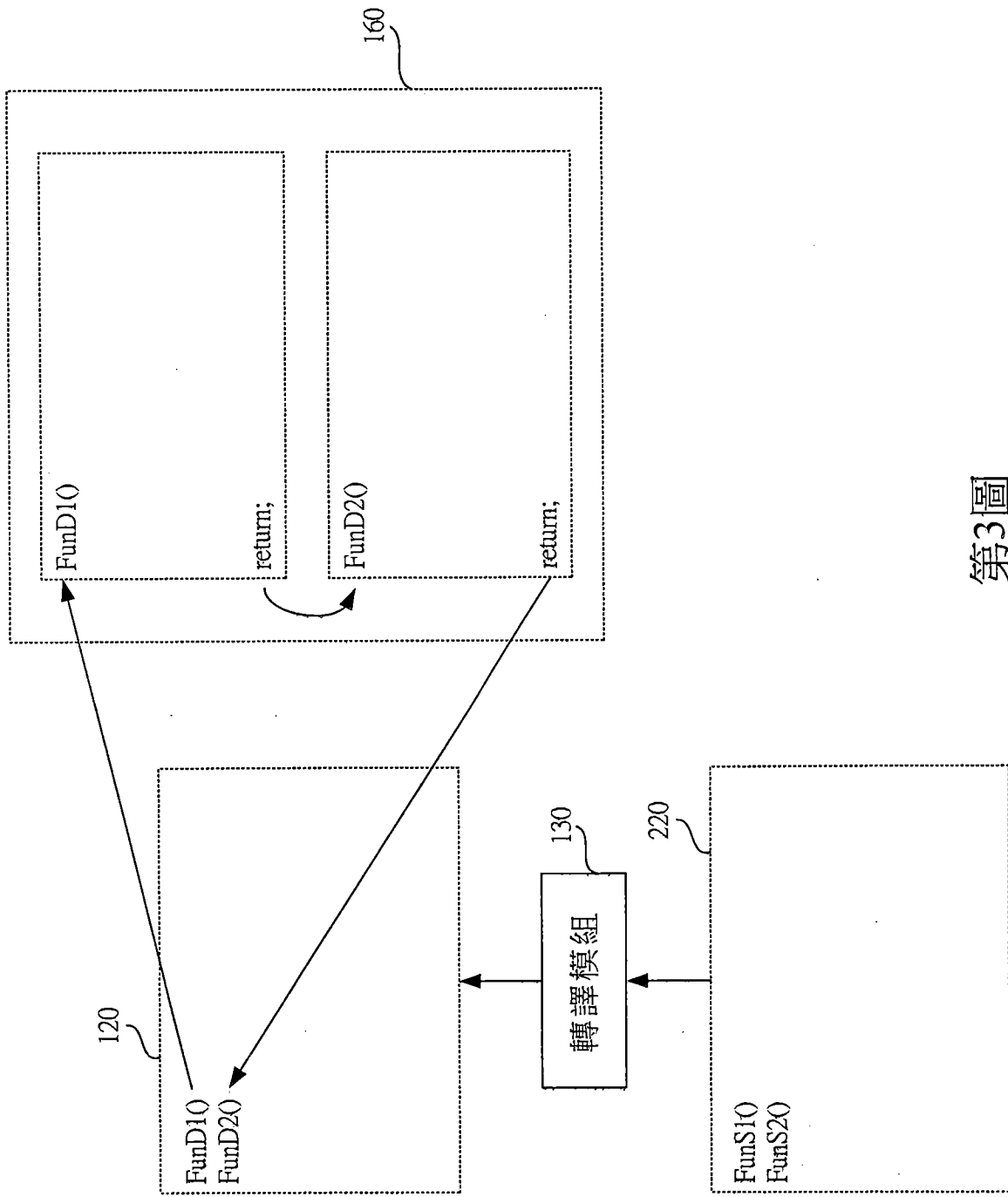
圖式



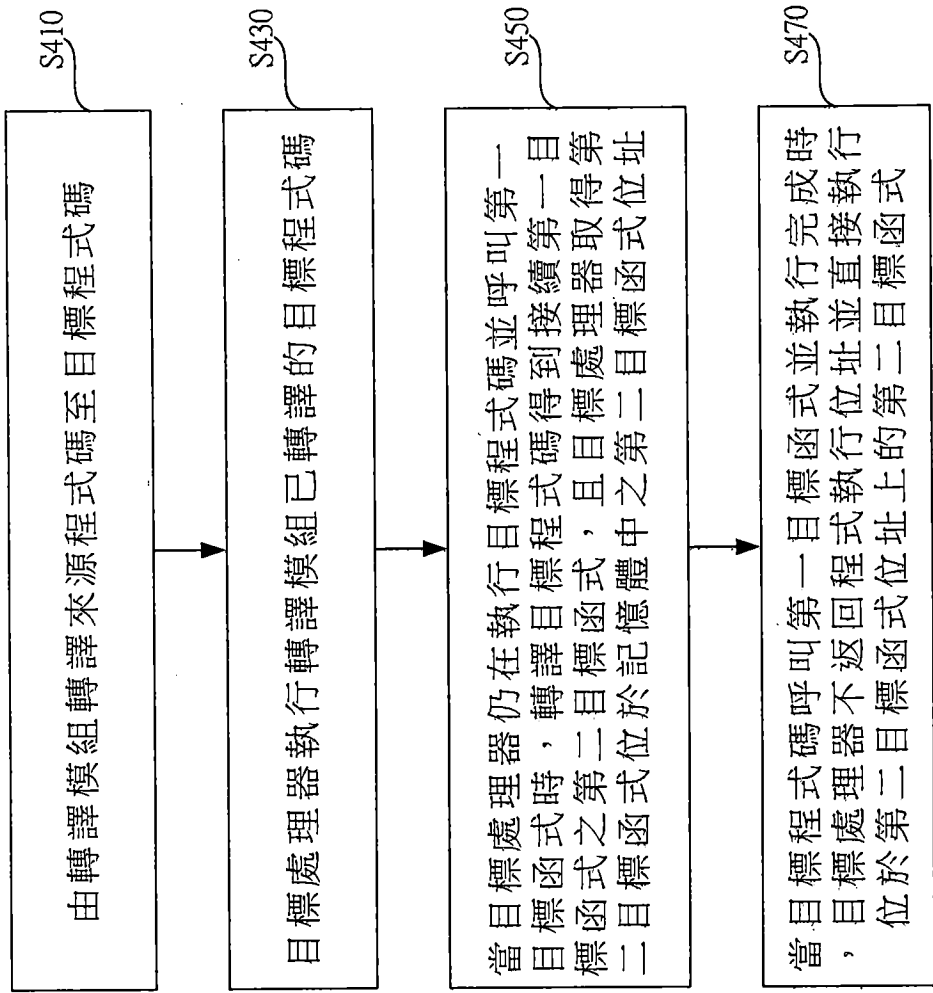
第1圖



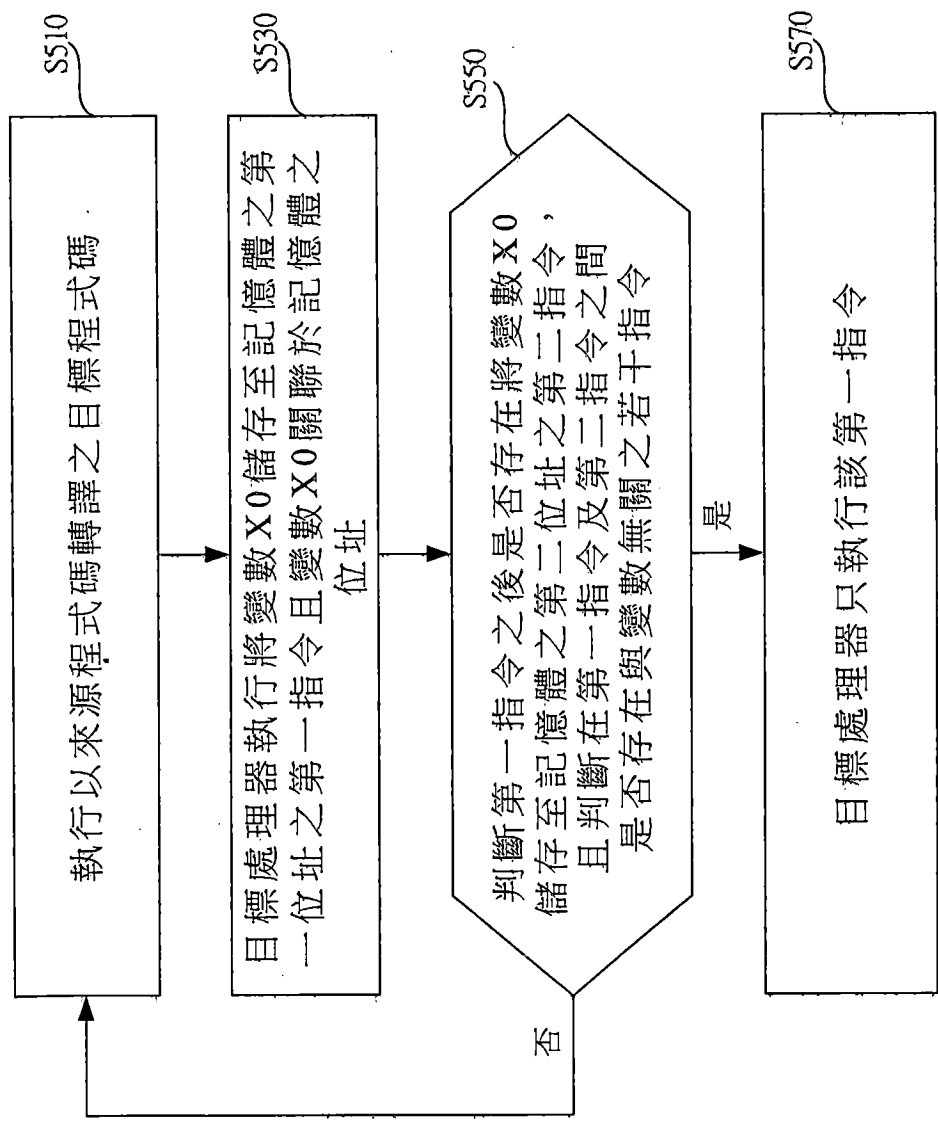
第2圖



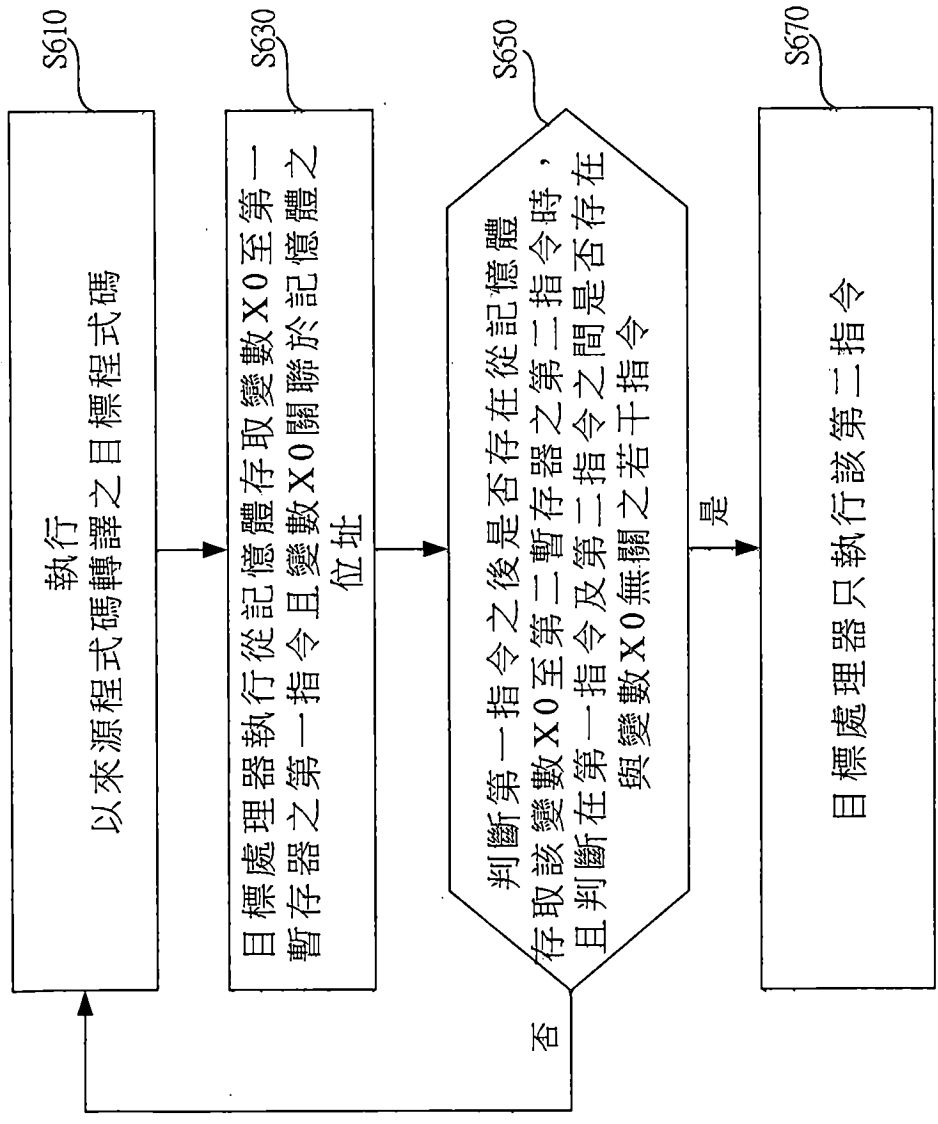
第3圖



第4圖



第5圖



第6圖