

公告本

申請日期: 90.11.12

案號: 90128029

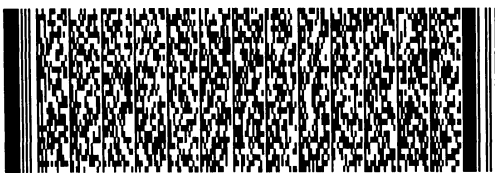
類別: G06F 9/42

(以上各欄由本局填註)

發明專利說明書

559733

一、發明名稱	中文	降低分支等待時間之方法和裝置
	英文	METHOD AND APPARATUS FOR REDUCING BRANCH LATENCY
二、發明人	姓名 (中文)	1. 約翰L, 瑞德佛
	姓名 (英文)	1. REDFORD, John L.
	國籍	1. 美國
	住、居所	1. 美國麻薩諸塞州02138劍橋郡莎薇兒街15號
三、申請人	姓名 (名稱) (中文)	1. 晶片製造設計公司
	姓名 (名稱) (英文)	1. CHIPWRIGHTS DESIGN, INC.
	國籍	1. 美國
	住、居所 (事務所)	1. 美國麻薩諸塞州02462牛頓, 華盛頓街2150號
	代表人姓名 (中文)	1. 約翰L, 瑞德佛
代表人姓名 (英文)	1. REDFORD, John L.	



本案已向

國(地區)申請專利

美國 US

申請日期

2000/11/10 09/710,699

案號

主張優先權

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)

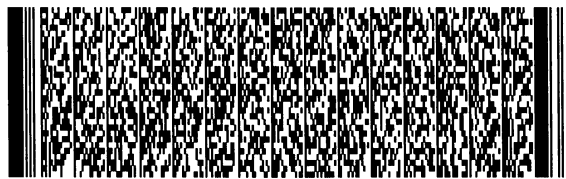
傳統上程序系統以多階(stage)或多週期來執行程式指令。記憶體位址儲存所須執行的指令在提取週期時由程式計數器來讀取並寫入到位址暫存器。位址暫存器的位址用以存取記憶體配置的指令，並從指令記憶體中提取指令且載入至指令暫存器。

一般而言，指令為多位元的字元，傳統上指令包括確定指令的多位元運算碼及記憶體位址資訊。舉例而言，在指令完成時，記憶體位址資訊可包括記憶體參數值所定義用以儲存指令運算元(opreand)體位置或儲存指令結果的記憶體位置。

在提取指令之後並下載至指令暫存器，並執行解碼週期。在解碼週期期間，指令運算碼被解碼成須執行的確認指令和所要求的判定程序步驟。如果有要求的運算元時依據指令，指令的記憶體位址資訊然後可被用於回收運算元。當解碼完成後，就能執行指令。

在一般指令執行的連續流程中，在提取指令之後，遞增程式計數器以指向指令記憶體的下個位置。

分支指令為一種非常普通類型的指令。分支指令用以改變指令執行時的一般連續流程。當到達迴圈的最後指令時，如果未滿足迴圈中止的條件，程式的流程必須回到迴圈的最頂部。在此情況下，分支指令用以載入遞增程式計數器而獲得頂部之位址。分支指令也載入指向程序起點的位址資料來將程序模組分散至位址暫存器而安排程式的執行。



五、發明說明 (2)

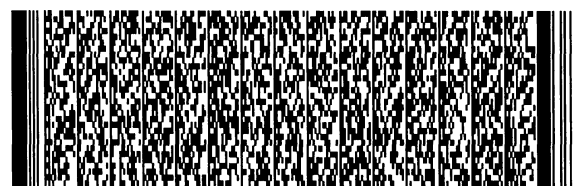
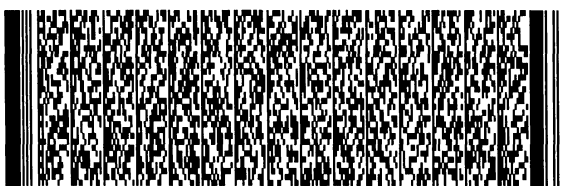
如同大部份傳統的指令一樣，分支指令包括運算碼和位址資訊。位址資訊傳統上採用定義為使用分支時程式將跳過的位址數量之偏移值的方式。偏移值傳統上為有號數目且在解碼週期時附加到目前指令上。在解碼週期之後，產生的總數載入至位址暫存器因此在下個提取週期時便可用以提取分支的第一指令。

程式執行的效率可借由管線化(pipelining)指令執行來加強。在管線化時，在第一提取週期期間提取第一指令，然後，執行第一指令的解碼週期，同時地，執行程式下個指令的提取週期。這是說，當第一指令解碼時提取下個指令。一般而言，此方法明顯地增加程式執行的速度和效率。

在此情況下，由管線化可獲得分支指令實現的效率須與包括計算分支目標位址的等待作妥協。在分支指令解碼週期期間，位移值必須加在目前程式計數器的位址值。這附加動作須要在解碼週期的明顯部份。因此，下個指令位址尚未判定下載至位址暫存器中，所以下個指令的提取週期無法同時執行。相反的，下個指令的提取週期，即分支目標指令，直到解碼週期完成之後才能開始。

上面描述的情況為一般分支等待的參考。而要求減少分支之後的第一指令位址即分支目標指令之等待，可借由在提取分支指令之後迅速載入至位址暫存器使下個提取週期時能提取分支目標指令。

本發明提供一減少分支等待條件的方法。依據本發明



五、發明說明 (3)

而提供分支指令程序的裝置與方法。分支指令包括具有運算碼的運算碼部份和具有分支目標指令的位址相關的位址資訊之位址部份，用以執行程式分支對應的分支指令。分支指令位址的部份包括記憶體區塊確認部份和位移值部份，記憶體區塊確認部份用以確認在記憶體的區塊執行分支對應的分支指令。使用分支指令的位址部份的位移值部份來當作分支指令的位址部份的記憶體區塊確認部份來確認在記憶體區的位址來執行分支至分支指令。

當獲得分支時分支指令的位址部份的記憶體區塊確認部份用以確認執行分支在記憶體的區塊對應的分支指令。在實施例中，區塊可為三可能區塊之其一。區塊之一可能為包括分支指令的區塊，而在此情況下分支目標指令與分支指令在同一區塊。其他區塊為目前領先區塊和目前落後區塊。

在實施例中，區塊確認器包括至少二位元容量以定義至少四碼來確認方塊。其中一碼用以確認分支指令的相同區塊。其中第二碼定義目前落後區塊，第三碼定義目前領先區塊。第四碼用以定義在相同區塊的分支之各別方向。第一碼亦可用以定義在相同區塊的分支為不同方向。因此，在此設定下舉一例子，例如第一碼和第二碼為00和01，可用以確認前進分支，前者在區塊中而後者在下一區塊。例如第三碼和第四碼為10和11，可用以定義後退分支，前者進入領先區塊而後者在目前區塊中。

在實施例中，區塊確認部份可用以分支預測，即預測



五、發明說明 (4)

是否將要選取分支。在實施例中，如果為後退分支，然後可預測將選取分支。如果分支為前進，然後可預測將不會選取分支。因此，在上述第四圖解集中，如果定義碼的第一位元為1，則稱之為後退分支，並預測將選取分支。換句話說，如果定義碼的第一位元為0，則稱之為前進分支，並預測將不會選取分支。

本發明的方法本質上用以減少或消除習知技術的分支等待。由分支指令的位移值位址資訊來直接產生分支目標位址而不須執行例如附加位址偏移參數值至程式計數器的任何算術運作上之時間消耗。所提供的分支目標位址直接送回位址暫存器來當作分支指令的部份提取週期之立即改變效率的硬體功能。因此，在下個週期可用以提取分支目標指令，而不使週期錯失。本結果可用以與習知技術在解碼週期時因為要計算分支目標指令所包括的等待而跳過下個提取週期來作比對。本發明因此提出在分支指令程序時比習知技術更明顯改善效率的方法。

圖式簡單說明：

本發明的前述和其他目的、特徵和優點將出現在本發明較佳實施例中作更個別的描述。如附圖所示類似參考特徵對應到不同觀點的相同部份。附圖無比例上的須要，反而本發明的原則強調附圖中的位移值。

第1圖係圖形圖解管線化結構下傳統的程式指令執行，包括提取時序和解碼週期；

第2圖係圖形圖解管線化結構下其中分支指令的程序



五、發明說明 (5)

以傳統的程式指令執行，包括提取時序和解碼週期；

第3圖係圖形圖解傳統分支指令；

第4圖係圖形圖解傳統結構下指令的執行之功能概要方塊圖；

第5圖係圖形圖解依據本發明的形態而解決第4圖中分支等待問題之指令的執行之功能概要方塊圖；

第6圖係圖解指令記憶體的位置和配置部份之概要方塊圖；

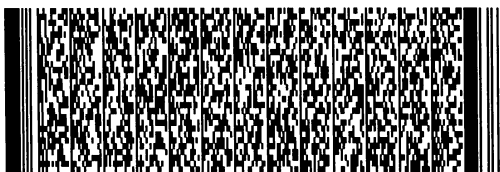
第7圖係圖解依本發明的分支指令的格式之概要圖；

第8圖係圖解依據本發明之結構由分支指令確認分支執行時指令的記憶體區塊而執行指令之功能概要方塊圖；以及

第9圖係圖解依據本發明可降低分支等待的裝置與方法之程式指令執行的時序時間和解碼週期之時序圖；

符號說明：

- | | |
|--------------|-------------------|
| 10 ~ 程序系統； | 12 ~ 程式計數器； |
| 14 ~ 遞增模組； | 16 ~ 多工器； |
| 18 ~ 分支預測； | 20 ~ 位址暫存器 |
| 22 ~ 指令記憶體； | 24 ~ 指令暫存器； |
| 26 ~ 總和模組； | 102、104、106 ~ 區塊； |
| 122 ~ 運算碼欄位； | 124 ~ 位址參數值欄； |
| 126 ~ 區塊欄位； | 128 ~ 位移值欄位； |
| 216 ~ 多工器； | 220 ~ 遞增模組； |
| 222 ~ 遞減模組； | 223 ~ 直接路徑。 |



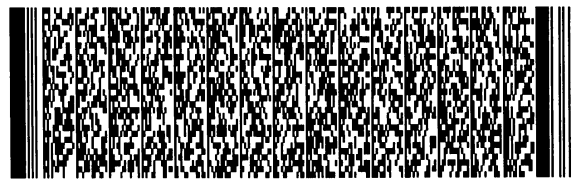
五、發明說明 (6)

實施例之說明：

參考第1圖係圖形圖解管線化結構下傳統的程式指令執行，包括提取時序和解碼週期，如第1圖所示，在第一週期期間，以儲存在程式計數器中的參數值PC當作指令記憶體配置位置來提取第一指令。在下個週期期間，將解碼指令PC且遞增程式計數器參數值PC+1以提取確定的指令。在第三個週期期間，解碼指令PC+1且遞增程式計數器參數值PC+2以提取確定的下個指令。每一週期以遞增程式計數器來連續控制指令提取及解碼的連續程序。因為管線化的架構，指令的程序較有效率，同一時間指令解碼和下個指令提取。一般而言，沒有指令週期會被浪費。

第2圖係圖形圖解管線化結構下其中分支指令的程序以傳統的程式指令執行，包括提取時序和解碼週期。如所示的時序圖，在第一週期時提取由程式計數器PC所確認之第一指令。在此例中，指令為分支指令。分支指令的格式如第3圖所示。如第3圖所示，分支指令包括運算碼部份和記憶體位移值部份。運算碼定義分支指令的種類，例如，分支被提取的情況。如果提取分支時位移值部份定義程式流程將繼續的位址。傳統上，位移值為有號數目且附加在分支指令的位址上。例如，目前程式計數器的參數值PC。

因此，在下個週期時，由附加目前位址到位移值中，例如，PC+Disp來將位址將被載入程試計數器以執行計算的分支。因為此附加要大量的時間來完成，分支指令的起始位址，即分支目標指令位址，不會被載入至程式計數器

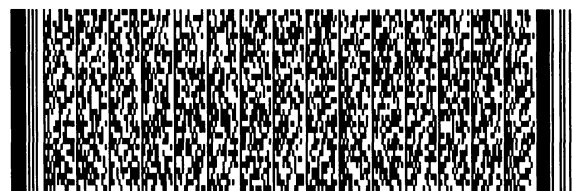
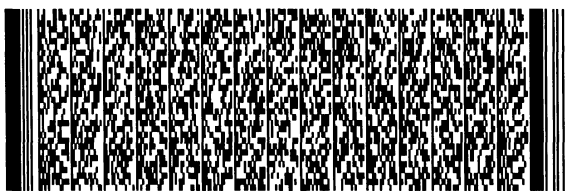


五、發明說明 (7)

中直到第二週期的末端。由於位址參數值PC+Disp所對應的分支目標指令，直到第三週期才會被提取。因此，在計算分支目標指令位址時週期會錯失。此情況通常當作分支等待。

第4圖為圖形圖解傳統結構下指令的執行之功能概要方塊圖。如第4圖所示。程序系統10包括用以產生指令連續執行的位址之程式計數器12來執行程式指令。在本描述中，假設指令位址長度為32位元。可理解的是本發明使用的位址大小可應用在其也位址的大小。由程式計數器12讀到的位址送往遞增模組14和總和模組26。總和模組26將程式計數器的參數值與指令暫存器24相加產生的結果位移值送往多工器16的輸入端。而位址也由遞增模組14來遞增，而遞增的結果提供至多工器16的另一輸入端。由分支預測模組18來透過多工器16的選擇輸入端S來選擇多工器16提供的位址。如果分支預測模組18決定要提取分支時，從多工器26輸入由多工器26將位移值與目前程式計數器參數值相加來產生的多工器16選擇之分支目標指令位址至分支預測模組18並下載至位址暫存器20。否則，選擇遞增的位址並下載至位址暫存器20。

下載至位址暫存器20位址提供指令記憶體22來存取下一個執行的指令。指令由指令記憶體22送往指令暫存器24。指令在解碼或其他程序之後通過。指令位移值部份如果有任何值的話，將可被傳送回多工器26以計算程式流程將要跳到的位址，例如，當提取的指令為分支指令時。如上所



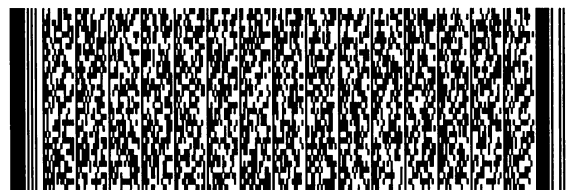
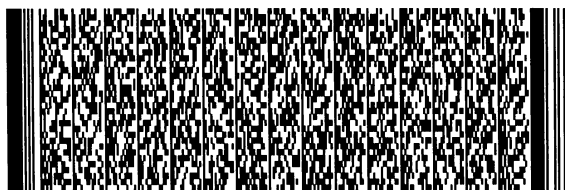
五、發明說明 (8)

述，此方法由於包括執行多工器26的時間而導致分支等待。

第5圖圖形圖解依據本發明而解決第4圖中分支等待問題之結構下指令的執行之功能概要方塊圖。在第5圖中，代替程式計數器參數值的分支指令之位移值部份，位移值直接由指令記憶體22的指令中提取並直接輸入至多工器16的位址輸入端來當作位移值的最低有效位元(LSBs)，在此各別圖解中，標示為15:0的多工器16最低有效位元。從指令記憶體22的記憶體配置中提取分支指令在分支指令的提取週期終止且在下個分提取週期開始之前迅速完成。因此，在下個成功的週期，可以提取為分支目標指令的下個指令，因為在下個成功提取週期開始之前它的位址已出現在位址暫存器20。因此，在成功的提取週期中分支指令和分支目標指令可被提取，而不錯失週期。可消除上面所描述的分支等待。

本方法的特別的缺點由第6圖來圖解，第6圖為傳統指令記憶體22的位址和配置部份之概要方塊圖。指令記憶體22定義為用以建立複數區塊102、104和106。如特別的圖解例子中所示，每一區塊具一組位址配置從 0000_{16} 至 $FFFF_{16}$ 。因此，每一記憶體位址定義的配置16最低有效位元與包括 2^{16} 配置之特別的區塊。

在程式執行的期間，在任何給予的期間，程式計數器存取儲存在其中一區塊的其中一配置指令。以區塊104為例。當分支指令開始計算時，依據上面所描述的方法，



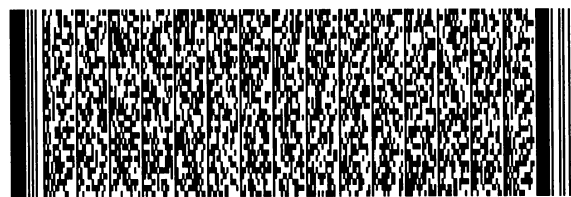
五、發明說明 (9)

指令的16位元位移值的部份將以他的16最低有效位元位置來代替下個位址。執行並繼續區塊104其中一個配置。此情況的缺點來自於實際上分支建立的配置必須在同一區塊中。因為這樣，所以可跳躍的大小可能有一些限制。依據程式計數器的目前參數值，即分支提取的配置。例如，當開始計算分支指令時如果程式目前執行靠近區塊的末端，則前進分支只能作小距離即小數量的配置。同樣的，如果程式目前執行靠近區塊的起點，後退分支僅限制在可能的距離。此情況限制了程式的系統彈性。

為了解決此問題，本發明的實施例中使用分支指令的位移值部份之一部份來定義分支指令應建立的指令記憶體中之區塊。第7圖圖解依本發明的分支指令的格式之概要圖。以第7圖為例，使用具有16位元位移值欄位的32位元指令。可理解的本發明可應用在其他的大小。

對應到第7圖，指令120的格式包括具有位元16-31的運算碼欄位122和具有位元0-15的位移值欄位128。128更區分成具有位元0-13的位址參數值欄位和具有位元14-15的區塊欄位126。2位元區塊欄位定義是否分支應以目前的區塊(對應到PC)、目前領先(對應到PC-1)或目前連結的區塊(對應到PC+1)來取代相同的區塊。位址參數值欄位124以分支目標指令應提取的定義區塊來定義位址。

因此，在本實施例的圖解中，區塊欄位126包括至少2位元容量來定義用以確認區塊的至少4個碼。其中一個碼定義分支指令在相同的區塊。第二碼定確認目前落後區

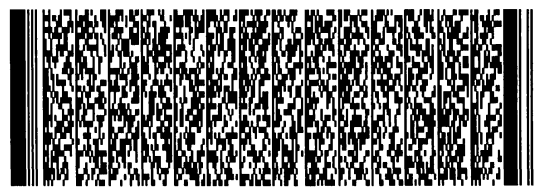
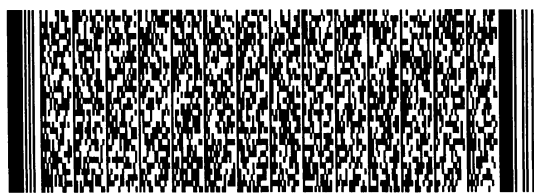


五、發明說明 (10)

塊，第三碼確認目前領先區塊，第四碼可用以定確認分支在相同的區塊而不同方向，而第一碼可用以定確認分支在相同的區塊而相反方向。因此，以此結構下舉一例子，第一和第二碼例如為00和01，可用以確認前進分支，前者在此區塊中而後者進入下個區塊。第三和第四碼例如為10和11，可用以確認後退分支，前者進入區塊中而後者在目前區塊中。因此，位址15的位元0可表示前進分支，位址15的位元1可表示後退分支。

第8圖係圖解依據本發明之結構由分支指令確認分支執行時指令的記憶體區塊而執行指令之功能概要方塊圖。在此結構下，14最低有效位元0-13從指令記憶體22連結到多工器216的4輸入端之其中3輸入端。其他18位元31-14由程式計數器12中取得並結合指令記憶體22的14最低有效位元。其他18位元31-14連結到遞增模組220、遞減模組222和直接路徑223，並將位元結果與指令記憶體22的14最低有效位元在多工器216的輸入端結合。遞增模組220用以產生當分支為記憶體下個區塊的位址；遞減模組222用以產生當分支為記憶體目前領先區塊的位址；直接路徑223用以產生當分支為記憶體目前區塊的位址；多工器216的第四輸入端直接從程式計數器中接收位元31-0並供一般連續程式的執行所使用。

分支偵測模組18用以選擇載入位址暫存器20的位址。如果沒有選取分支，則選擇14的位元31-0位址。如果分支選擇至下個區塊，則選擇220的位元31-14位址。如果分支



五、發明說明 (11)

選取至先前區塊，則選擇222的位元31-14位址。如果分支選取至目前的區塊，則選擇223的位元31-14位址。

依據本發明的具有區塊確認碼之分支指令可用以作分支偵測的幫助。如上所示，位址15的位元0可表示前進分支，位址15的位元1可表示後退分支。一般分支預測的方法中僅取後退分支而不取前進分支。因此，依據本發明，如果位址15的位元0則不取分支，如果位址15的位元1則取分支。

因此，依據本發明在第8圖的實施例中，提取分支的位址可能範圍超過第5圖的實施例。使用後者的方法，位址參數值的位移值的部份僅特定的 2^{14} 可能位址，而非第5圖前者方法中的 2^{16} 可能位址。然而，在此例中，使用第8圖的方法，三個特定的記憶體區塊中的每一個之 2^{14} 可能位址。所以大大地增加可能分支距離和實現程式彈性的結果。

第9圖係圖解依據本發明降低分支等待的裝置與方法之程式指令執行的時序時間和解碼週期之時序圖。如第9圖所示，依據本發明，可在成功的週期內提取分支指令和分支目標指令。消除了傳統方法中所發現的分支等待。

在此註解本發明本以上述不同的方法來實行。舉例，本發明可在指令快取記憶體載入欲執行之指令開始執行前立即實行，而不必以程式編譯或連結來個別改變指令本身。在後者方法中，區塊確認欄位在示範的實施例中為2位元欄位，當載入指令時附加到適當的指令快取記憶體配



五、發明說明 (12)

置中。

在此實施例中，舉例所述的32位元指令，分支指令的16位元位址參數值和2位元方塊確認參數值。可理解的這些位元的數量在不超過本發明的範圍之內可以改變。

本發明的較佳實施例描述如上，然而，在不脫離本發明之精神和範圍內，當可做些許的更動與潤飾。此外，其餘實施例皆在下列申請專利範圍內。

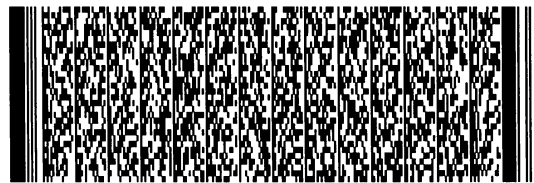
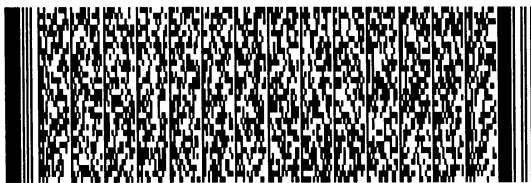


四、中文發明摘要 (發明之名稱：降低分支等待時間之方法和裝置)

提供一種用以降低分支等待時間之方法和裝置。分支指令包括運算碼部份和位址部份。位址部份包括位移和用以確定分支目標指令配置在指令記憶體區塊的碼，而區塊用以執行對應到分支指令的分支。在提取週期期間取得分支指令，分支指令的位移部份插入位暫暫存來當作提取下個指令的位址。碼所確定的區塊包括用以確定位址暫存器指向正確的區塊的分支目標指令。因為如此，在下個指令的提取週期可提取分支目標指令來執行。因此，先前系統的下個提取週期之分支程序等待可被省略且程式計數值中遞增偏移量來計算分支目標位址亦可消除。

英文發明摘要 (發明之名稱：METHOD AND APPARATUS FOR REDUCING BRANCH LATENCY)

A method and apparatus for reducing latency in execution of branch instructions are provided. A branch instruction includes an opcode portion and an address portion. The address portion includes a displacement and a code which identifies a block in the instruction memory in which the branch target instruction is located and, therefore, the block to which execution will branch in response to the branch instruction. During the fetch cycle in which the branch instruction is fetched, the displacement



四、中文發明摘要 (發明之名稱：降低分支等待時間之方法和裝置)

英文發明摘要 (發明之名稱：METHOD AND APPARATUS FOR REDUCING BRANCH LATENCY)

portion of the branch instruction is reinserted into the address register as the address of the next instruction to be fetched. The code that identifies the block containing the branch target instruction is used to ensure that the address register is pointing to the correct block. As a result, during the next instruction fetch cycle, the branch target instruction is fetched for execution. Hence, the branch processing latency found in prior systems in which the next fetch



四、中文發明摘要 (發明之名稱：降低分支等待時間之方法和裝置)

英文發明摘要 (發明之名稱：METHOD AND APPARATUS FOR REDUCING BRANCH LATENCY)

cycle is skipped while the branch target address is computed, such as by adding an offset to the program counter value, is eliminated.



六、申請專利範圍

1. 一種分支指令程序的方法，上述分支指令為儲存在一指令記憶體的一區塊之一程式的複數指令其中一種，上述方法包括：

提供具有一運算部份包括一運算碼和具有一位址部份包括分支目標指令位址相關的位址資訊之上述分支指令而一程式的執行對應到上述分支指令的分支；

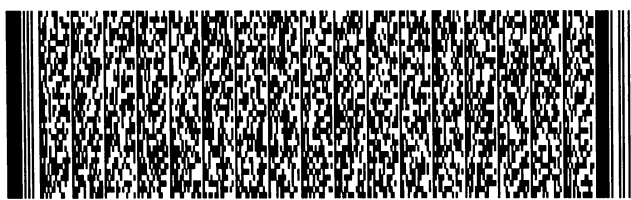
提供上述分支指令的上述位址部份和一記憶體區塊定義部份和一位移值部份，上述記憶體區塊確認部份確認了上述分支指令對應的上述分支執行在上述記憶體的一區塊；以及

使用上述分支指令上述位址部份的上述位移值部份來分支至上述分支目標指令當作上述分支指令的上述位址部份之上述記憶體區塊確認部份確認的上述記憶體區塊的一位址。

2. 如申請專利範圍第1項所述之方法，其中分支至上述分支目標指令包括使用上述分支指令的上述位址部份之上述位移值部份和上述分支指令的上述位址部份之上述記憶體區塊確認部份來產生一分支目標位址。

3. 如申請專利範圍第2項所述之方法，其中上述分支目標位址在上述分支指令提取週期間產生。

4. 如申請專利範圍第1項所述之方法，其中上述分支指令的上述位址部份之上述記憶體區塊確認部份確認了分支執行的區塊為一上述記憶體區塊領先上述分支指令的記憶體區塊，上述記憶體區塊落後上述分支指令的記憶體區



六、申請專利範圍

塊和上述分支指令的記憶體區塊。

5. 如申請專利範圍第1項所述之方法，其中更包括使用上述分支指令的上述位址部份之上述記憶體區塊確認部份來預測是否提取上述分支。

6. 如申請專利範圍第1項所述之方法，其中上述分支指令的上述位址部份之上述記憶體區塊確認部份確認了至少4碼來預測是否要提取上述分支。

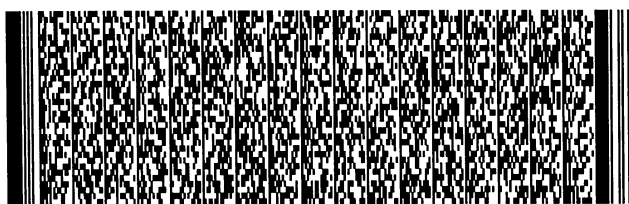
7. 如申請專利範圍第6項所述之方法，其中上述4碼包括前進分支的一第一對碼和落後分支的一第二對碼。

8. 如申請專利範圍第7項所述之方法，其中上述第一對碼包括一碼用以前進分支至下一記憶體區塊和一碼用以前進分支至上述分支指令的上述區塊。

9. 如申請專利範圍第7項所述之方法，其中上述第二對碼包括一碼用以後退分支至一領先記憶體區塊和一碼用以後退分支至上述分支指令的上述區塊。

10. 一種分支指令的程序裝置，上述分支指令為程式的複數指令其中一種，上述裝置包括：

一指令記憶體，用以儲存指令，上述分支指令儲存在上述指令記憶體的複數區塊其中之一，上述分支指令具有包括運算一運算碼的一運算碼部份和包括分支目標指令相關的位址資訊之一位址部份而程式的執行分支至對應的分支指令，且上述分支指令的上述位址部份具有一記憶體區塊確認部份和一位移值部份，上述記憶體區塊確認部份確認了執行上述分支指令對應的分支在上述記憶體的一區



六、申請專利範圍

塊；以及

一處理器，用以執行儲存在上述指令記憶體中的指令，上述處理器使用上述分支指令的上述位址部份之上述位移值部份來使指令執行分支至分支目標指令而由上述分支指令的上述位址部份之上述記憶體區塊定義部份來定義上述記憶體區塊的一位址。

11. 如申請專利範圍第10項所述之裝置，其中上述處理器使用上述分支指令的上述位址部份之上述位移值部份和上述分支指令的上述位址部份之上述記憶體區塊確認部份來產生一分支目標位址。

12. 如申請專利範圍第11項所述之裝置，其中在分支指令提取週期期間產生上述分支目標位址。

13. 如申請專利範圍第10項所述之裝置，其中上述分支指令的上述位址部份之上述記憶體區塊確認部份確認了分支執行的區塊為一上述記憶體區塊領先上述分支指令的記憶體區塊，上述記憶體區塊落後上述分支指令的記憶體區塊和上述分支指令的記憶體區塊。

14. 如申請專利範圍第10項所述之裝置，其中上述處理器使用上述分支指令的上述位址部份之上述記憶體區塊確認部份來預測是否提取上述分支。

15. 如申請專利範圍第10項所述之裝置，其中上述分支指令的上述位址部份之上述記憶體區塊確認部份確認了至少4碼來預測是否提取上述分支。

16. 如申請專利範圍第15項所述之裝置，其中上述4碼

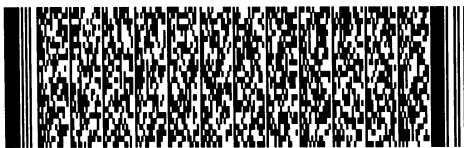


六、申請專利範圍

包括前進分支的一第一對碼和落後分支的一第二對碼。

17. 如申請專利範圍第16項所述之裝置，其中上述第一對碼包括一碼用以前進分支至一下個記憶體區塊和一碼用以前進分支至上述分支指令的上述區塊。

18. 如申請專利範圍第16項所述之裝置，其中上述第二對碼包括一碼用以後退分支至一領先記憶體區塊和一碼用以後退分支至上述分支指令的上述區塊。



92年6月7日
修正
補充

90128029 號中文圖式修正本

修正日期：92.6.27

	1	2	3	4	
提取	PC	PC+1	PC+2	PC+3	……
解碼	-	PC	PC+1	PC+2	……

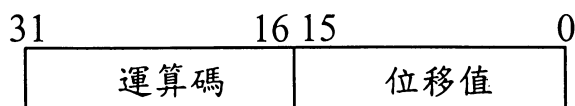
→ 時序圖(週期)

第 1 圖

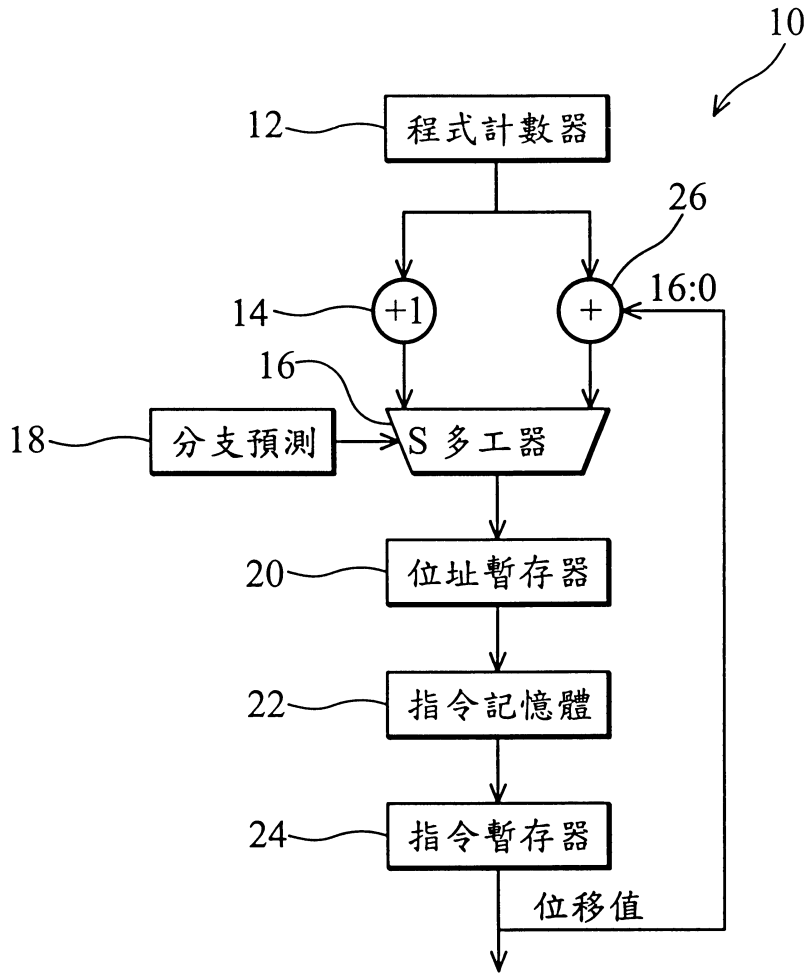
	1	2	3	4	
提取	PC	-	PC+Disp.	-	……
解碼		PC	-	PC+Disp.	……

→ 時序圖(週期)

第 2 圖

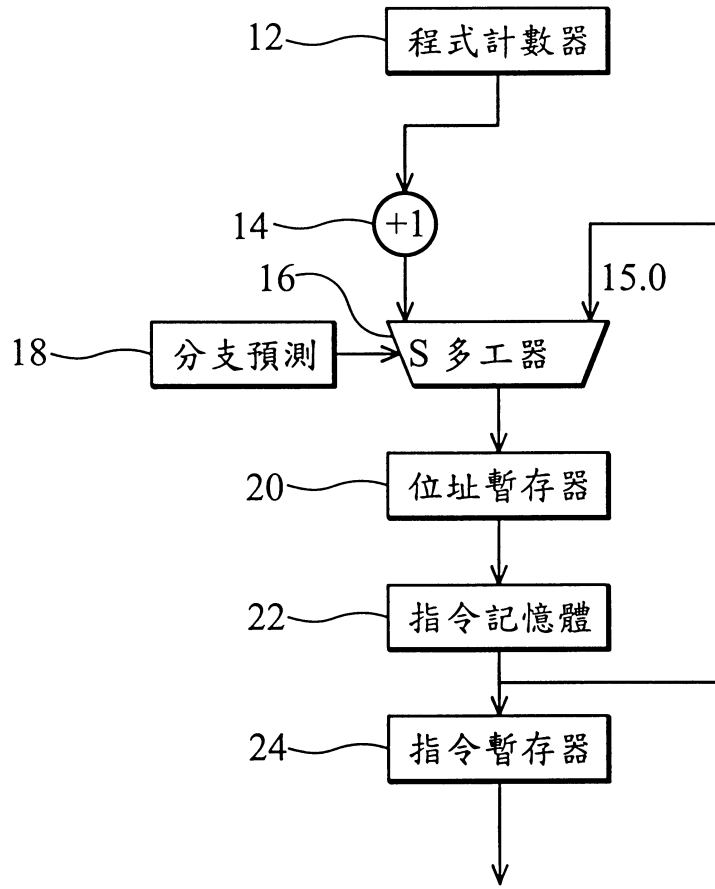


第 3 圖



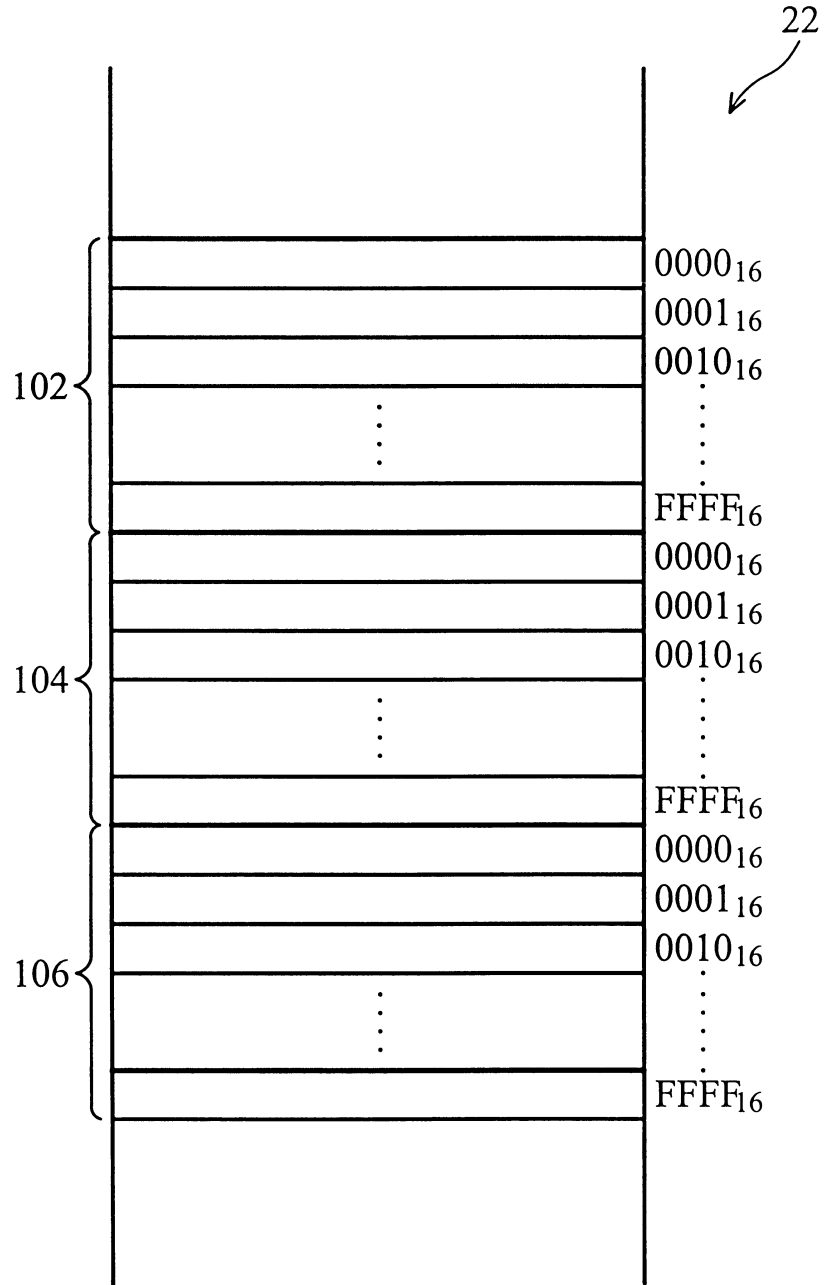
第 4 圖

a² b 21



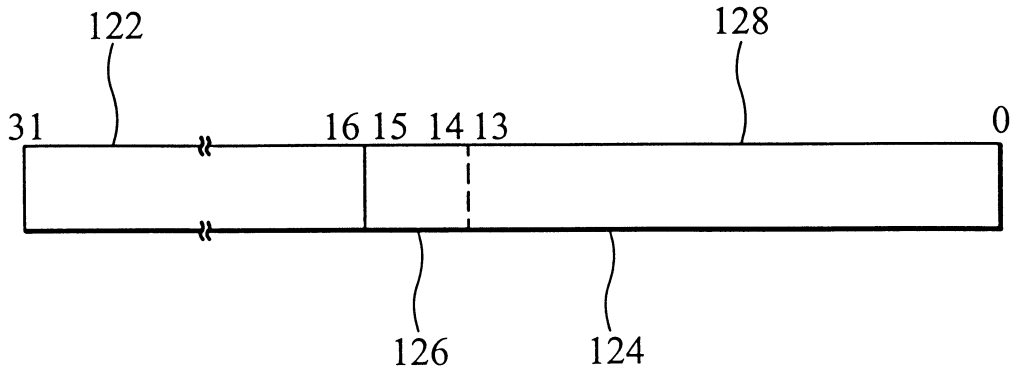
第 5 圖

26 27

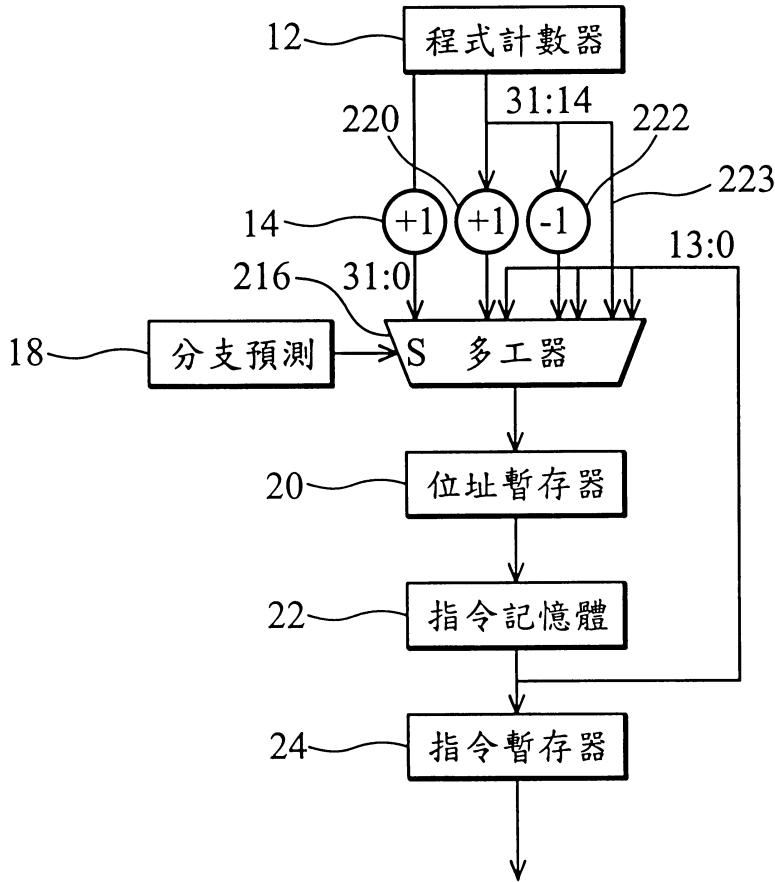


第 6 圖

q2. b. 21



第 7 圖



第 8 圖

	1	2	3	4	→
提取	PC	PCw+Disp.	PCw+Disp.+1	PCw+Disp.+2	⋯⋯
解碼		PC	PCw+Disp.	PCw+Disp.+1	⋯⋯

第 9 圖