



(19) 中華民國智慧財產局

(12) 發明說明書公告本

(11) 證書號數：TW I455023 B

(45) 公告日：中華民國 103 (2014) 年 10 月 01 日

(21) 申請案號：098136712

(22) 申請日：中華民國 98 (2009) 年 10 月 29 日

(51) Int. Cl. : G06F9/30 (2006.01)

(30) 優先權：2008/10/30 美國 12/290,395

(71) 申請人：英特爾股份有限公司 (美國) INTEL CORPORATION (US)  
美國

(72) 發明人：奧席爾 伊多 OUZIEL, IDO (IL)；瑞柏波特 里瑚 RAPPOPORT, LIHU (IL)；瓦倫泰 羅柏 VALENTINE, ROBERT (IL)；賈柏 朗 GABOR, RON (IL)；瑞湖凡希 潘凱 RAGHUVANSHI, PANKAJ (IN)

(74) 代理人：林志剛

(56) 參考文獻：

TW 200424933A US 5860107

US 2007/0174314A1

審查人員：謝進忠

申請專利範圍項數：20 項 圖式數：6 共 0 頁

(54) 名稱

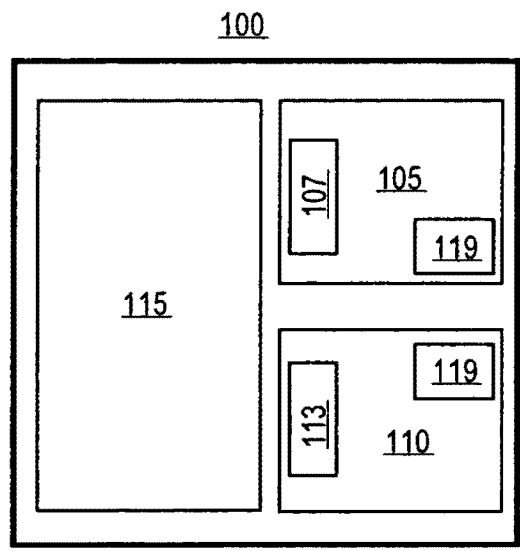
促進處理器中有效指令融合之設備、方法及計算系統

APPARATUS, METHOD AND COMPUTING SYSTEM FOR PROMOTING EFFICIENT  
INSTRUCTION FUSION IN A PROCESSOR

(57) 摘要

一種能在電腦系統內有效指令融合之技術。在一實施例中，若在指令佇列內的第一指令可與第二指令相融合，則處理器邏輯使該第二指令的處理延遲一臨界時間量。

A technique to enable efficient instruction fusion within a computer system. In one embodiment, a processor logic delays the processing of a second instruction for a threshold amount of time if a first instruction within an instruction queue is fusible with the second instruction.



- 100 . . . 微處理器
- 105 . . . 處理器核心
- 107 . . . 本地快取
- 110 . . . 處理器核心
- 113 . . . 本地快取
- 115 . . . 共享快取記憶體
- 119 . . . 邏輯

圖 1

# 發明專利說明書

(本申請書格式、順序，請勿任意更動，※記號部分請勿填寫)

公告本

※申請案號：098136712

※申請日：098 年 10 月 29 日

※IPC 分類：

G06F 9/30  
(2006.01)

一、發明名稱：(中文/英文)

促進處理器中有效指令融合之設備、方法及計算系統

Apparatus, method and computing system for promoting efficient instruction fusion  
in a processor

二、中文發明摘要：

一種能在電腦系統內有效指令融合之技術。在一實施例中，若在指令佇列內的第一指令可與第二指令相融合，則處理器邏輯使該第二指令的處理延遲一臨界時間量。

**三、英文發明摘要：**

A technique to enable efficient instruction fusion within a computer system. In one embodiment, a processor logic delays the processing of a second instruction for a threshold amount of time if a first instruction within an instruction queue is fusible with the second instruction.

四、指定代表圖：

(一) 本案指定代表圖為：第(1)圖。

(二) 本代表圖之元件符號簡單說明：

100：微處理器

105：處理器核心

107：本地快取

110：處理器核心

113：本地快取

115：共享快取記憶體

119：邏輯

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：無

## 六、發明說明：

### 【發明所屬之技術領域】

本發明的實施例一般有關於資訊處理的領域，且更具體而言，有關於計算系統及微處理器中的指令融合之領域。

### 【先前技術】

指令融合為結合兩個指令成為單一指令之程序，其導致處理器內的一操作（或微操作，「uop」）序列。處理器指令佇列（IQ）中所儲存的指令可在從IQ中被讀出之後且被傳送至指令解碼器之前或由指令解碼器予以解碼之後被「融合」。典型上，發生於指令被解碼之前的指令融合被稱為「巨融合」，而發生於指令被解碼之後（例如，成為uop）的指令融合被稱為「微融合」。巨融合的一例為比較（「CMP」）指令或測試指令（「TEST」）（「CMP/TEST」）與條件跳躍（「JCC」）指令的結合。CMP/TEST與JCC指令對會定期地出現於例如迴圈的末端之程式中，在此實施比較，且依據比較的結果而發生（taken）或不發生分支。因為巨融合可有效地增加指令運轉量（throughput），所以會希望盡可能地多找出融合指令的機會。

對於某些習知技術的處理器微架構中待找出的指令融合機會而言，CMP/TEST與JCC指令兩者會需要同時存在於IQ中，使得當從IQ中讀取指令時，CMP/TEST與JCC指令可

被融合。然而，若在 IQ 中，有可融合的 CMP/TEST 指令且沒有另外的指令已被寫入至 IQ（亦即，CMP/TEST 指令為 IQ 中的最後指令），則即使程式順序中的下個指令為 JCC 指令，但是 CMP/TEST 指令可從 IQ 中被讀取且被傳送至解碼器，而不會被融合。漏失融合機會可能出現的一例是若 CMP/TEST 與 JCC 碰巧通過儲存器邊界（例如，16 位元組邊界），致使在一週期內，CMP/TEST 被寫入 IQ 中，而接下來的週期，JCC 將被寫入 IQ 中。在此情況中，若沒有停止條件，則在 CMP/TEST 正從 IQ 中被讀取的同時或之後，JCC 將被寫入 IQ 中，所以將漏失融合機會，導致 IQ 之多次不必要的讀取，而降低指令運轉率及過多的功率消耗。

#### 【發明內容及實施方式】

本發明的實施例可被使用來改善處理器中的指令運轉率及/或降低處理器的功率消耗。在一實施例中，指令融合的機會不是被漏失，就是被找出且因此發生指令融合。在一實施例中，原先會漏失的指令融合機會被找出係藉由將從指令佇列（IQ）的最後指令的讀取，或發佈從該 IQ 所讀取的該最後指令至解碼階段的延遲達臨界週期數，使得任何隨後的可融合指令可被提取且儲存於該 IQ 中（或至少被辨識，而不必被儲存於該 IQ 中），且其後與最後可融合指令一起被融合。在一實施例中，因為如此實施可避免兩個（否則可融合）指令分別被解碼及處理，而不是如單一指令，所以將第一可融合指令之讀取或發佈延遲達臨界週

期數可改善處理器效能。

臨界等待週期數的選擇可依據使用特定實施例的微架構。例如，在一實施例中，臨界週期數可為2，而在其他實施例中，臨界週期數可為比2更多或更少。在一實施例中，臨界等待週期數提供最大時間量來等待其後的可融合指令被儲存至IQ，而同時在等待處理可融合指令為分離指令結束後之後續可融合指令方面，保持整體的延遲/效能優點。在其他實施例中，雖然等待週期數會使指令運轉率降低（儘管是暫時地），但是例如在電力更緊要之處，臨界數等待週期會更高，以便確保額外電力不被使用來分開處理兩個可融合指令。

圖1繪示可使用本發明的至少一實施例之微處理器。特別而言，圖1繪示微處理器100，其具有一個或多個處理器核心105及110，每一處理器核心分別具有與其相關聯的本地快取107及113。圖1中也繪示的是共享快取記憶體115，其可儲存本地快取107及113之每一本地快取中所儲存的資訊之不同形式的至少某些資訊。在某些實施例中，微處理器100也可包括未顯示於圖1中的其他邏輯，諸如整合的記憶體控制器、整合的圖形控制器、以及實施電腦系統內的其他功能之其他邏輯（諸如I/O控制）。在一實施例中，多處理器系統中的各微處理器或多核心處理器中的各處理器核心可包括邏輯119或者是與邏輯119相關聯，以根據至少一實施例致能中斷溝通技術。此邏輯可包括能使指令的融合比某些習知技術實施中者更有效率之電路、軟體或

兩者。

在一實施例中，邏輯 119 可包括降低漏失指令融合機會的可能性之邏輯。在一實施例中，當沒有其後的指令被儲存於 IQ 或其他的提取指令儲存架構中時，邏輯 119 將來自 IQ 的第一指令（例如，CMP）之讀取延遲。在一實施例中，在讀取 IQ，或將第一可融合指令發佈至解碼器或其他處理邏輯之前，邏輯 119 使第一可融合指令的讀取或發佈持續臨界週期數（例如，2 個週期），使得若有尚未被儲存於 IQ 中之可與第一指令融合的第二可融合指令（例如，由於此兩個可融合指令被儲存於不同儲存器邊界中的記憶體或快取中），則不會漏失融合此兩個可融合指令的機會。在某些實施例中，此臨界值可為固定的，而在其他實施例中，此臨界值可為可變的，其可藉由使用者或根據與使用者無關的演算法來予以修改。在一實施例中，第一可融合指令為 CMP 指令，而第二可融合指令為 JCC 指令。在其他實施例中，第一指令與第二指令的任一或兩者可不為 CMP 或 JCC 指令，而是任何可融合的指令。此外，本發明的實施例可被應用來融合超過兩個指令。

例如，圖 2 繪示可使用本發明的一實施例之前端匯流排（FSB）電腦系統。任何處理器 201、205、210、或 215 可存取位於處理器核心 223、227、233、237、243、247、253、257 的一處理器之內，或者是與之相關聯之來自任何本地一階（L1）快取記憶體 220、225、230、235、240、245、250、255 的資訊。再者，任何處理器 201、205、210

、或 215 可存取來自共享二階 ( L2 ) 快取 203、207、213、217，或來自經由晶片組 265 的系統記憶體 260 之任一的資訊。圖 2 中的一個或多個處理器可包括邏輯 219 或者與邏輯 219 相關聯，以根據至少一實施例致能指令融合的改良效率。

除了圖 2 中所繪示的 FSB 電腦系統之外，可使用與本發明的不同實施例 ( 包括點對點 ( P2P ) 互連系統及環互連系統 ) 相結合之其他系統組構。例如，圖 3 的 P2P 系統可包括許多處理器，其中僅兩個處理器 370、380 被顯示作為例子。處理器 370、380 的每一處理器可包括與記憶體 32、34 相連接之本地記憶體控制器中心 ( MCH ) 372、382。處理器 370、380 可使用點對點 ( PtP ) 介面電路 378、388，經由點對點 ( PtP ) 介面 350 來交換資料。處理器 370、380 可使用點對點介面電路 376、394、386、398，經由個別 PtP 介面 352、354 而與晶片組 390 交換資料。晶片組 390 也可經由高效能圖形介面 339 而與高效能圖形電路 338 交換資料。本發明的實施例可被設置於任何具有任何數量的處理器核心之處理器內，或圖 3 的 PtP 匯流排代理器之每一 PtP 匯流排代理器內。在一實施例中，任何處理器核心可包括本地快取記憶體或者與本地快取記憶體 ( 未顯示 ) 相關聯。再者，共享快取 ( 未顯示 ) 可被包含於兩處理器的外部之任一處理器中，但是經由 P2P 互連而與處理器相連接，使得若處理器被設置成低功率模式，則任一處理器或兩個處理器的本地快取資訊可被儲存於此共享快取中。圖 3 中的處理

器或核心之一個或多個可包括邏輯 319 或者與邏輯 319 相關聯，以根據至少一實施例致能改善指令融合的效率。

在至少一實施例中，在諸如 IQ 清除操作的某些中間操作發生（發生在第一可融合指令與第二可融合指令之間）之前，第二可融合指令可以不存入 IQ 內，而導致漏失融合此兩個其他可融合指令的機會。在一實施例中，其中快取（或緩衝器）儲存將被排定（也許多次）來藉由處理器執行之相關序列的解碼指令（在其從 IQ 讀取且被解碼之後）或 uop（例如，「解碼的串流緩衝器」或「DSB」、「追蹤快取」、或「TC」），第一可融合 uop（例如，CMP）可被儲存於快取中，而可融合第二 uop（例如，JCC）不在相同的可定址範圍內（例如，相同的快取方式）。例如，在 JCC 正跨過快取線（由於快取漏失）或跨過頁邊界（由於轉換旁視緩衝器漏失）之處可能發生，在此情況中，此快取會儲存 CMP，而不儲存 JCC。接著，若在 CMP 被儲存於快取中之後而在 JCC 被儲存於快取中之前，清除處理器核心管線（例如，由於宣告「清除」訊號），則快取以其方式的一方式，僅儲存 CMP，而不儲存 JCC。

關於儲存 CMP 的快取線之其後的檢查，快取會將漏失的 JCC 解譯為漏失的存取，且 JCC 會被標示為下個快取填入操作的附加點。然而，因為 CMP+JCC 會自 IQ 中被讀取為融合的，所以不會發現此附加點。因此，所請求的 JCC 不會匹配來自 IQ 之要被填入的任何 uop，因此快取將不能填入漏失的 JCC，而會持續地漏失預期融合的 CMP+JCC 之線

。此外，在使用來儲存 uop 快取填入請求之未決的填入請求佇列（PFRQ）之一實施例中，保留用於特定融合的命令填入之入口不會解配置（因為預期融合的命令填入從不發生），而會保持無用，直到下個清除操作。在一實施例中，PFRQ 入口鎖住於每次存取漏失融合指令入口的時候會發生，因此可防止任何相同位置的後續填入。

爲了防止 PFRQ 入口之錯誤或不想要的鎖住，在一實施例中，可使用狀態機來監測 uop 從 IQ 被讀取，以偵測狀況，其中例如由於到達入口的最後 uop 而未偵測填入開始點，所以完全漏失具有對應的 PFRQ 入口之區域（例如，被標示用於填入的區域）。在一實施例中，當符合此條件時，狀態機可使 PFRQ 入口被解配置。在其他實施例中，可藉由不在快取內產生可從 IQ 讀取的可融合指令而不存在兩種可融合指令，來予以避免不想要的 PFRQ 入口鎖住。例如，若接在 CMP 後的是非 JCC 指令，則在快取中會產生融合指令入口，但是唯有當 CMP 自 IQ 中被單獨讀出（例如，在臨界等待時間終止之後）時，融合指令入口才不被填入至快取。在其他實施例中，狀態機已偵測跳過的填入區域之次數會予以計數，在跳過填入區域之某些臨界次數計數之後，會實施快取刷新或無效操作。填入區域然後會自快取中移除，且然後會重新填入融合指令。

圖 4 繪示根據一實施例之狀態機，其可被使用來避免由於 IQ 中的漏失可融合指令所導致之不想要的 PFRQ 入口鎖住情況。在狀態 401，其中 IQ 中的指令不在被標示爲填

入的區域中，「填入區域開始」訊號表示 IQ 即將處理被映射至填入區域的指令（根據快取雜湊之來自填入區域的指令），但是不會在儲存於 PFRQ（「邊緣（lip）」）405 中的線性指令指標處開始，這會導致狀態機移至狀態 410。若 IQ 中的下個指令（不久將被解碼）終止填入區域（例如，終止如藉由快取所雜湊的線，或為發生分支），則狀態機使對應的 PFRQ 入口解配置 415，且狀態機返回狀態 401。然而，若填入指標器等於狀態 401 或狀態 410 時的填入區域邊緣 430，則狀態機進入狀態 420，其中存取係在填入區域內或在填入開始點之後。自狀態 420，填入區域指示中的最後 uop 425 將使狀態機返回狀態 401，而不解配置對應的 PFRQ 入口。圖 4 的狀態機可被實施於硬體邏輯、軟體、或其某些組合中。在其他實施例中，可使用其他狀態機或邏輯。

圖 5 繪示可結合本發明的至少一實施例來予以使用之操作的流程圖。在操作 501，判定 IQ 中之目前存取的指令是否可與任何後續指令一起融合。若否，則在操作 505，下個指令係自 IQ 中予以存取且重置延遲計數。若是，則在操作 510，增加延遲計數器，且在操作 515，判定是否到達延遲計數臨界值。若其為否，在操作 520，則不實施目前存取的指令之指令融合。若其為是，則在操作 505，下個指令係從 IQ 存取且重置延遲計數。在其他實施例中，可實施其他操作，以改善指令融合的效率。

圖 6 繪示可結合至少一實施例來予以使用之操作的流

程圖。爲了實施具有一些解碼器電路的處理器中之一實施例，確保第一可融合指令於特定解碼器電路（其可解碼融合指令）上進行解碼會是有幫助的。在圖6中，於操作601，判定特定指令是否可爲指令的融合對之第一個。若否，則在操作605，發佈融合指令。若是，則在操作610，判定接在第一可融合指令之後者是否是IQ中的有效指令。若是，則在操作610，發佈融合指令。若否，則在操作615，判定第一可融合指令是否要被發佈至能支援融合指令的解碼器。在一實施例中，解碼器-0可解碼融合指令。若第一可融合指令未被發佈至解碼器-0，則在操作620，第一可融合指令被移動，或「清除（nuked）」至不同的解碼器，直到其對應於解碼器-0。在操作625，計數器被設定爲初始值N，且在操作630，若此指令之後是有效指令或計數器爲零，則在操作635，發佈融合指令。否則，在操作640，降低計數器且清除無效指令。在其他實施例中，計數器會增加至最終值。在其他實施例中，除了清除操作之外，其他操作可清除無效指令。

至少一實施例的一個或多個態樣可藉由儲存於代表處理器內的不同邏輯之機器可讀取媒體上之表示資料來予以實施，其當藉由機器來予以讀取時，使此機器產生邏輯，以實施在此所述的技術。被稱爲「IP核心」的此類表示可被儲存於有形的機器可讀取媒體（「磁帶」）上，且供應給不同客戶，或裝入實際做成邏輯或處理器的製造機器之製造設備。

因此，指示微架構記憶體區域存取的方法及設備已予以說明。要瞭解的是，以上的說明係意謂例示而非限制。在讀取及瞭解以上的說明之後，許多其他實施例對於熟習此項技藝者是顯然可知的。因此，本發明的範圍應該參考後附的申請專利範圍，以及此類申請專利範圍之等效的完整範圍來予以決定。

#### 【圖式簡單說明】

本發明的實施例係範例，並非限制而繪示於附圖的圖式中，且其中類似元件符號表示於類似元件，其中：

圖 1 繪示本發明的至少一實施例可予以使用之微處理器的方塊圖。

圖 2 繪示本發明的至少一實施例可予以使用之共享匯流排電腦系統的方塊圖。

圖 3 繪示本發明的至少一實施例可予以使用之點對點互連電腦系統的方塊圖。

圖 4 繪示可被使用來實施本發明的至少一實施例之狀態機的方塊圖。

圖 5 繪示可被使用於實施本發明的至少一實施例之操作的流程圖。

圖 6 繪示可被實施於至少一實施例中之操作的流程圖。

。

#### 【主要元件符號說明】

- 100 : 微處理器
- 105 : 處理器核心
- 107 : 本地快取
- 110 : 處理器核心
- 113 : 本地快取
- 115 : 共享快取記憶體
- 119 : 邏輯
- 201 : 處理器
- 203 : 共享二階 ( L2 ) 快取
- 205 : 處理器
- 207 : 共享二階 ( L2 ) 快取
- 210 : 處理器
- 213 : 共享二階 ( L2 ) 快取
- 215 : 處理器
- 217 : 共享二階 ( L2 ) 快取
- 219 : 邏輯
- 220 : 本地一階 ( L1 ) 快取記憶體
- 223 : 處理器核心
- 225 : 本地一階 ( L1 ) 快取記憶體
- 227 : 處理器核心
- 230 : 本地一階 ( L1 ) 快取記憶體
- 233 : 處理器核心
- 235 : 本地一階 ( L1 ) 快取記憶體
- 237 : 處理器核心

- 240：本地一階（L1）快取記憶體
- 243：處理器核心
- 245：本地一階（L1）快取記憶體
- 247：處理器核心
- 250：本地一階（L1）快取記憶體
- 253：處理器核心
- 255：本地一階（L1）快取記憶體
- 257：處理器核心
- 260：系統記憶體
- 265：晶片組
- 319：邏輯
- 32：記憶體
- 338：高效能圖形電路
- 339：高效能圖形介面
- 34：記憶體
- 350：點對點介面
- 352：個別PtP介面
- 354：個別PtP介面
- 370：處理器
- 372：本地記憶體控制器中心（MCH）
- 376：點對點介面電路
- 378：點對點介面電路
- 380：處理器
- 382：本地記憶體控制器中心（MCH）

386 : 點對點介面電路

388 : 點對點介面電路

390 : 晶片組

394 : 點對點介面電路

398 : 點對點介面電路

## 七、申請專利範圍：

1. 一種促進處理器中有效指令融合之設備，該設備包含：

指令佇列（IQ）；以及

邏輯，用以判定該 IQ 中之目前存取的指令是否為可與任何尚未被儲存於該 IQ 中之後續指令一起融合的第一可融合指令，以及用以該第一可融合指令的處理延遲達一臨界時間量，使得若一第二可融合指令在該臨界時間量內被儲存於該 IQ 內，則與該第一可融合指令可相融合但尚未被儲存於該 IQ 中之該第二可融合指令可與該第一可融合指令融合。

2. 如申請專利範圍第 1 項之設備，其中該第一可融合指令及該第二可融合指令在被儲存於該 IQ 中之前，會通過提取邊界而被儲存。

3. 如申請專利範圍第 1 項之設備，其中該邏輯係用以只有當該第一可融合指令為該 IQ 中所儲存的唯一指令時，才延遲該第一可融合指令的處理。

4. 如申請專利範圍第 1 項之設備，其中該邏輯包括計數器，在該第一可融合指令被儲存於該 IQ 中且為該 IQ 中的最後指令之後的每一週期，該計數器增加計數一次，直到到達與該臨界時間量相對應的臨界週期數。

5. 如申請專利範圍第 1 項之設備，更包含狀態機，若在被儲存於快取中的該第一可融合指令與被儲存於該快取中的該第二可融合指令之間實施中間指令，則該狀態機防

止未決的填入請求佇列 (PFRQ) 鎖住與該第一可融合指令及該第二可融合指令相對應的人口。

6. 如申請專利範圍第 5 項之設備，其中該中間指令係用以致使該 IQ 被清除。

7. 一種促進處理器中有效指令融合之方法，該方法包含：

判定指令佇列 (IQ) 內之目前存取的指令是否可與該 IQ 中待儲存之任何隨後的指令相融合；

若判定該目前存取的指令不可與該 IQ 中待儲存之任何隨後的指令相融合，則自該 IQ 中存取下個指令，且重置延遲計數器；以及

若判定該目前存取的指令為可融合且若該目前存取的指令為該 IQ 中的最後指令，則增加該延遲計數器之計數。

8. 如申請專利範圍第 7 項之方法，更包含若該目前存取的指令與隨後的指令為可相融合且該延遲計數器尚未到達臨界值，則將該目前存取的指令與該隨後的指令相融合。

9. 如申請專利範圍第 8 項之方法，更包含若該目前存取的指令與隨後的指令為不可相融合，則使該目前存取的指令與該隨後的指令分開處理。

10. 如申請專利範圍第 8 項之方法，更包含若該延遲計數器已到達該臨界值，則使該目前存取的指令與該隨後的指令分開處理。

11. 如申請專利範圍第 7 項之方法，更包含若該目前存

取的指令與該隨後的指令為可相融合，並在該隨後的指令被儲存於快取中之前且在該目前存取的指令被儲存於該快取中之後實施中間事件，則防止未決的填入請求佇列（PFRQ）鎖住與該目前存取的指令及該隨後的指令相對應之入口。

12. 一種促進有效指令融合之計算系統，該計算系統包含：

儲存器，用以儲存分別在第一存取邊界及第二存取邊界內的第一可融合指令及第二可融合指令；

處理器，具有提取邏輯，用以將該第一可融合指令及該第二可融合指令提取至指令佇列（IQ）中；

發佈邏輯，用以若判定出目前存取的指令不是該第一可融合指令，則不延遲地自該 IQ 讀取要被發佈的該目前存取的指令；

每當該第二可融合指令已經被儲存於該 IQ 中時，該發佈邏輯不延遲地自該 IQ 擇一讀取要被發佈的該第一可融合指令；

延遲邏輯，用以若該第二可融合指令尚未被儲存於該 IQ 中，則使來自該 IQ 的該第一可融合指令之讀取延遲達一臨界週期數；以及

指令融合邏輯，用以若該第二可融合指令在該第一可融合指令之後且在已到達該臨界週期數之前，被儲存於該 IQ 中，則該指令融合邏輯用以融合該第一可融合指令及該第二可融合指令。

13.如申請專利範圍第 12 項之系統，更包括計數器，若該第一可融合指令為該 IQ 中的唯一指令，則該計數器之計數增加，而當已到達該臨界週期數時，該計數器停止計數。

14.如申請專利範圍第 13 項之系統，其中若該第二可融合指令在已到達該臨界週期數之前，被儲存於該 IQ 中，則該計數器重置。

15.如申請專利範圍第 13 項之系統，其中若該第二可融合指令在已到達該臨界週期數之前，被儲存於該 IQ 中，則該指令融合邏輯在當該該臨界週期數已到達時融合該第一可融合指令及該第二可融合指令。

16.如申請專利範圍第 12 項之系統，其中該儲存器包括指令快取，且該第一存取邊界及該第二存取邊界之個別的大小為 64 位元組。

17.如申請專利範圍第 12 項之系統，其中該儲存器包括動態隨機存取記憶體，且該第一存取邊界及該第二存取邊界之個別的大小為 4096 位元組。

18.如申請專利範圍第 12 項之系統，其中該第一可融合指令為比較或測試 (CMP/TEST) 指令，而該第二可融合指令為條件跳躍 (JCC) 指令。

19.如申請專利範圍第 18 項之系統，其中該臨界週期數為 2。

20.如申請專利範圍第 12 項之系統，更包括狀態機，若在被儲存於快取中的該第一可融合指令與被儲存於該快

取中的該第二可融合指令之間實施中間事件，則該狀態機防止未決的填入請求佇列（PFRQ）鎖住與該第一可融合指令及該第二可融合指令相對應的入口。

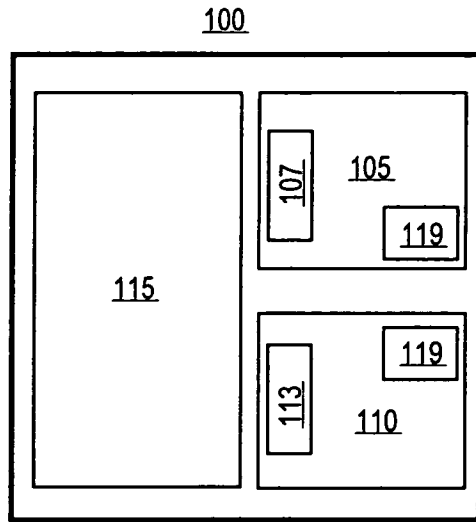


圖 1

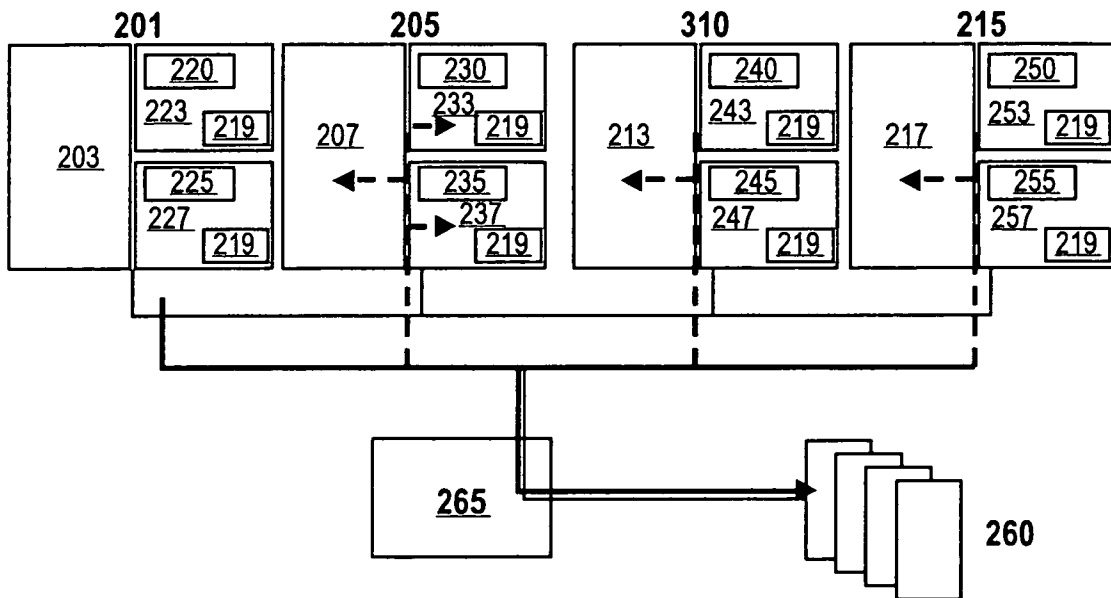


圖 2

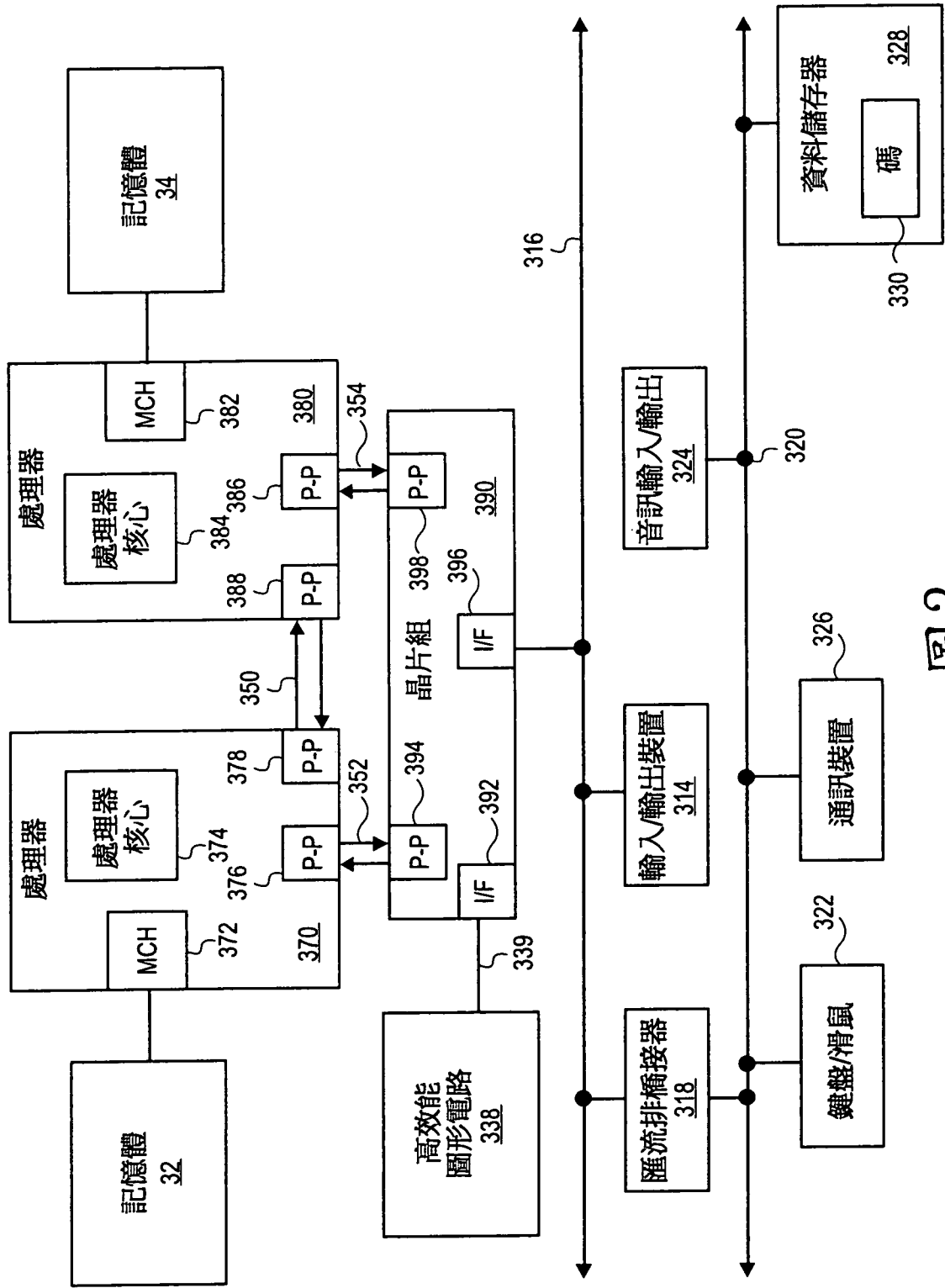


圖3

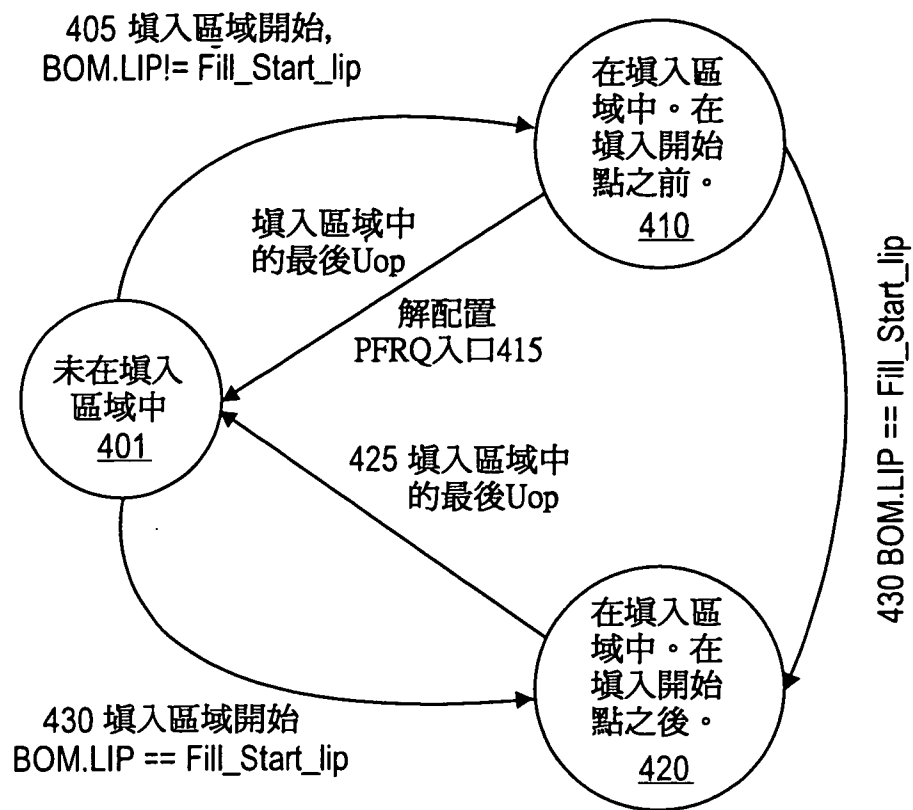


圖4

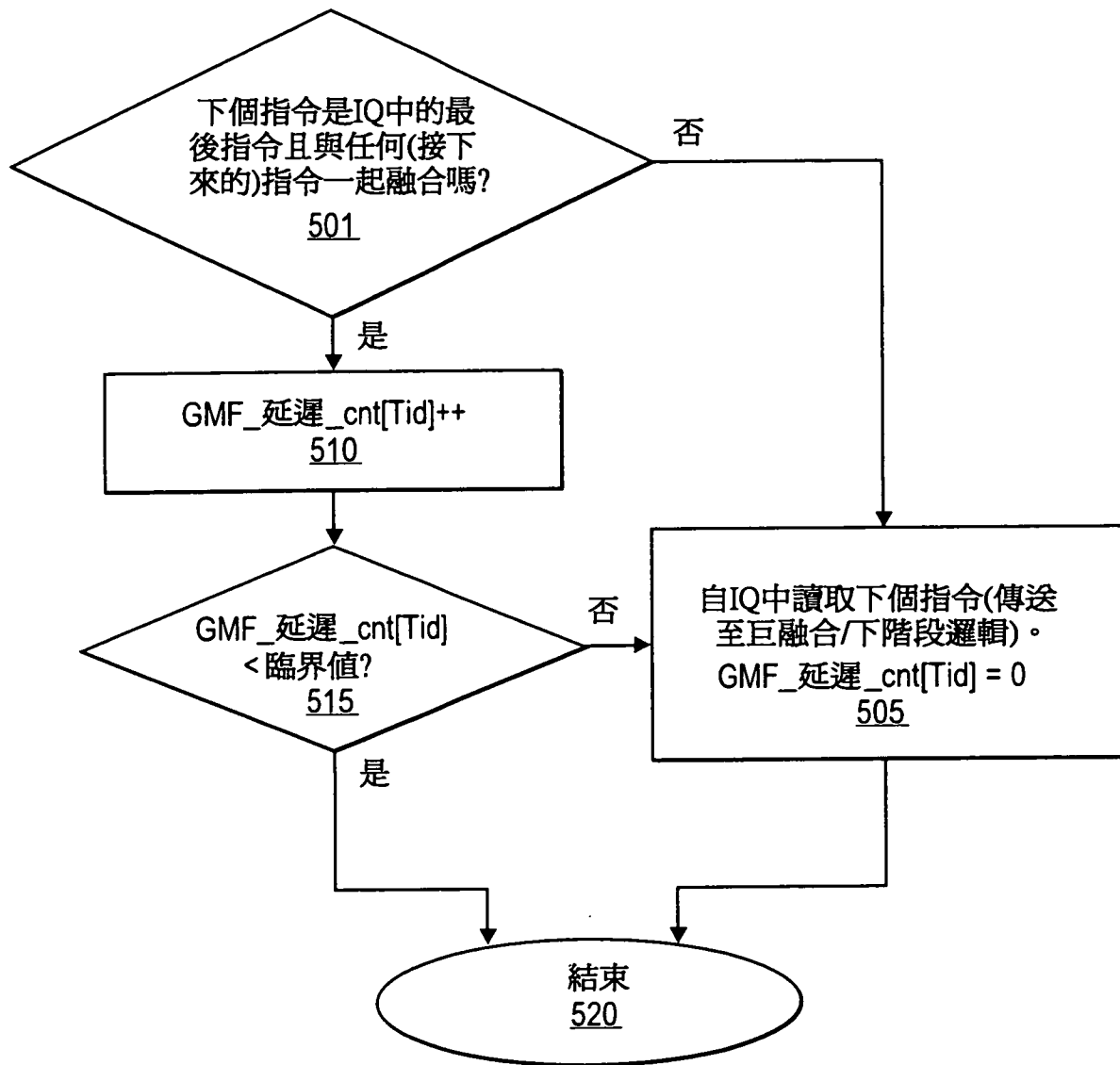


圖5

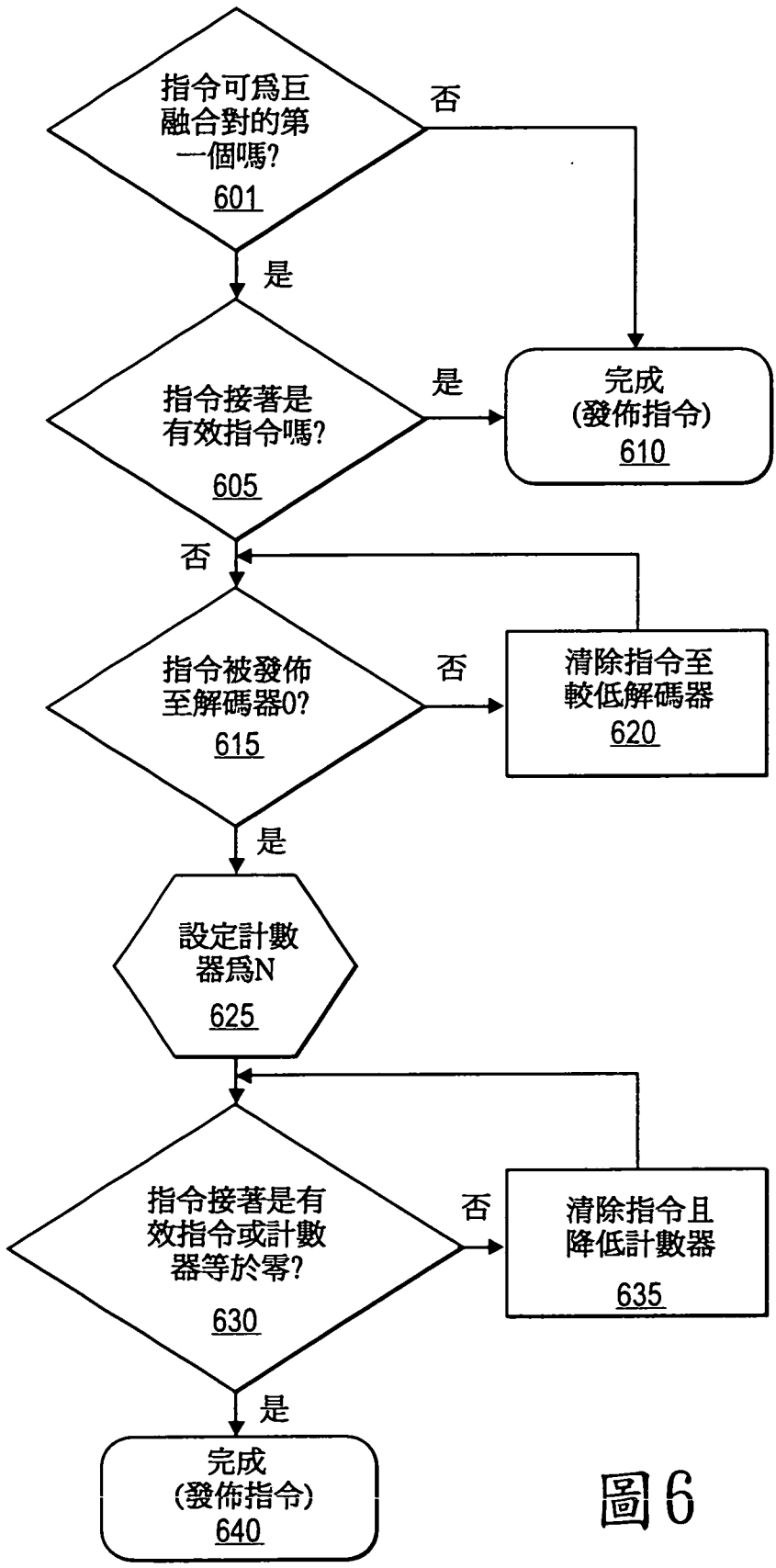


圖6