



(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 I700634 B

(45)公告日：中華民國 109 (2020) 年 08 月 01 日

(21)申請案號：107136463

(22)申請日：中華民國 107 (2018) 年 10 月 17 日

(51)Int. Cl. : G06F9/30 (2018.01)

G06F15/16 (2006.01)

(30)優先權：2017/10/20 英國

1717291.7

(71)申請人：英商葛夫科有限公司 (英國) GRAPHCORE LIMITED (GB)

英國

(72)發明人：諾爾斯 賽門 克里斯汀 KNOWLES, SIMON CHRISTIAN (GB)；亞歷山大 艾

倫 葛拉漢 ALEXANDER, ALAN GRAHAM (GB)

(74)代理人：陳長文

(56)參考文獻：

TW 201728174A

審查人員：黃鴻鈞

申請專利範圍項數：23 項 圖式數：16 共 78 頁

(54)名稱

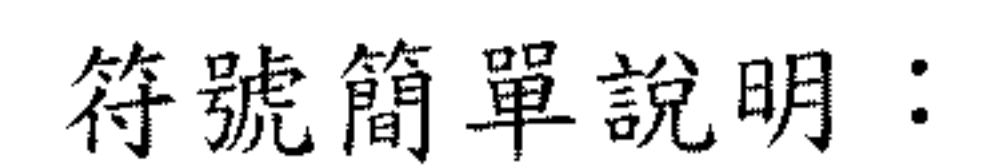
具有在多微磚處理配置中之同步的處理系統、操作處理系統之方法、以及相關電腦程式產品

(57)摘要

本發明揭示一種包括多個微磚及介於該等微磚之間的一互連件之處理系統。該互連件係用於根據一整體同步平行方案在一些或所有該等微磚之一群組之間通信，藉此該群組中之各微磚執行一微磚上運算階段，接著執行一微磚間交換階段，其中阻止該交換階段直至該群組中之所有微磚已完成該運算階段。該群組中之各微磚在完成該運算階段之後具有一局域退出狀態。指令集包括用於藉由各微磚在完成其運算階段之後執行之一同步指令以將一同步請求傳訊至該互連件中之邏輯。回應於自該群組中之所有該等微磚接收該同步請求，該邏輯釋放下一交換階段且亦使該群組中之所有該等微磚之一經彙總狀態可用。

A processing system comprising multiple tiles and an interconnect between the tiles. The interconnect is used to communicate between a group of some or all of the tiles according to a bulk synchronous parallel scheme, whereby each tile in the group performs an on-tile compute phase followed by an inter-tile exchange phase with the exchange phase being held back until all tiles in the group have completed the compute phase. Each tile in the group has a local exit state upon completion of the compute phase. The instruction set comprises a synchronization instruction for execution by each tile upon completion of its compute phase to signal a sync request to logic in the interconnect. In response to receiving the sync request from all the tiles in the group, the logic releases the next exchange phase and also makes available an aggregated state of all the tiles in the group.

指定代表圖：



36:同步控制器/同步邏輯/同步控制器邏輯

39:專用同步邏輯/微磚 上同步邏輯/同步邏輯

40i 至 40N:雙輸入
AND 閘

42i 至 42N:全域共識
暫存器(\$GC)/退出狀
態暫存器/暫存器



I700634

【發明摘要】

【中文發明名稱】

具有在多微磚處理配置中之同步的處理系統、操作處理系統之方法、以及相關電腦程式產品

【英文發明名稱】

PROCESSING SYSTEM WITH SYNCHRONIZATION IN A MULTI-TILE PROCESSING ARRANGEMENT, METHOD OF OPERATING A PROCESSING SYSTEM, AND RELATED COMPUTER PROGRAM PRODUCT

【中文】

本發明揭示一種包括多個微磚及介於該等微磚之間的一互連件之處理系統。該互連件係用於根據一整體同步平行方案在一些或所有該等微磚之一群組之間通信，藉此該群組中之各微磚執行一微磚上運算階段，接著執行一微磚間交換階段，其中阻止該交換階段直至該群組中之所有微磚已完成該運算階段。該群組中之各微磚在完成該運算階段之後具有一局域退出狀態。指令集包括用於藉由各微磚在完成其運算階段之後執行之一同步指令以將一同步請求傳訊至該互連件中之邏輯。回應於自該群組中之所有該等微磚接收該同步請求，該邏輯釋放下一交換階段且亦使該群組中之所有該等微磚之一經彙總狀態可用。

【英文】

A processing system comprising multiple tiles and an interconnect between the tiles. The interconnect is used to communicate between a group of some or all of the tiles according to a bulk synchronous parallel scheme, whereby each tile in the group performs an on-tile compute phase followed by an inter-tile exchange phase with the exchange phase being held back until all tiles in the group have completed the compute

phase. Each tile in the group has a local exit state upon completion of the compute phase. The instruction set comprises a synchronization instruction for execution by each tile upon completion of its compute phase to signal a sync request to logic in the interconnect. In response to receiving the sync request from all the tiles in the group, the logic releases the next exchange phase and also makes available an aggregated state of all the tiles in the group.

【指定代表圖】

圖11

【代表圖之符號簡單說明】

4i至4N	微磚群組
34	晶片上互連件/交換組構/互連件/硬體互連件/內部互連件
36	同步控制器/同步邏輯/同步控制器邏輯
38i至38N	暫存器/局域共識暫存器\$LC/局域退出狀態暫存器\$LC/退出狀態暫存器/控制暫存器
39	專用同步邏輯/微磚上同步邏輯/同步邏輯
40i至40N	雙輸入AND閘
42i至42N	全域共識暫存器(\$GC)/退出狀態暫存器/暫存器

【發明說明書】

【中文發明名稱】

具有在多微磚處理配置中之同步的處理系統、操作處理系統之方法、以及相關電腦程式產品

【英文發明名稱】

PROCESSING SYSTEM WITH SYNCHRONIZATION IN A MULTI-TILE PROCESSING ARRANGEMENT, METHOD OF OPERATING A PROCESSING SYSTEM, AND RELATED COMPUTER PROGRAM PRODUCT

【技術領域】

【0001】 本發明係關於同步一多微磚處理配置中之多個不同微磚之工作負載，各微磚包括其自身處理單元及記憶體。特定言之，本發明係關於整體同步平行(BSP)通信方案，藉此一群組微磚之各者必須在該群組中之該等微磚之任一者可進行至一交換階段之前完成一運算階段。

【先前技術】

【0002】 一多執行緒處理器係能夠同時執行多個程式執行緒之一處理器。該處理器可包括為多個不同執行緒所共有之某一硬體(例如，一共同指令記憶體、資料記憶體及/或執行單元)；但支援多執行緒，該處理器亦包括特定於各執行緒之某一專用硬體。

【0003】 該專用硬體至少包括用於可同時執行之數個執行緒之各者之一各自內文暫存器檔案。一「內文」在談論多執行緒處理器時係指同時執行之執行緒之一各自者之程式狀態(例如，程式計數器值、狀態及當前運算元值)。該內文暫存器檔案係指用於表示該各自執行緒之此程式狀態之暫存器之各自集合。一暫存器檔案中之暫存器與通用記憶體相異之處在於，暫存器位址係固定為指令字中之位元，而記憶體位址可藉由執行指令而運算。一給定內文之暫存器通常包括用於各自執行緒之一各自程式計數

器，及用於在藉由該執行緒執行之運算期間暫時保存作用於該各自執行緒且藉由該各自執行緒輸出之資料之一各自運算元暫存器組。各內文亦可具有用於儲存各自執行緒之一狀態(例如，該各自執行緒是否暫停或運行)之一各自狀態暫存器。因此，當前運行執行緒之各者具有其自身單獨程式計數器，及視需要運算元暫存器及(若干)狀態暫存器。

【0004】 多執行緒之一個可能形式係平行性。即，除了多個內文以外，亦提供多個執行管線：即，用於將並行執行之指令之各串流之一單獨執行管線。然而，此需要在硬體方面進行大量複製。

【0005】 因此代替性地，多執行緒處理器之另一形式採用並行性而非平行性，藉此執行緒共用一共同執行管線(或一管線之至少一共同部分)且不同執行緒透過此相同、共用執行管線交錯。由於增加隱藏管線延時之機會，所以相較於無並行性或平行性，仍可改良多執行緒處理器之效能。又，此方法並不需要與具有多個執行管線之一完全平行處理器一樣多的專用於各執行緒之額外硬體，且因此不會引發如此多的額外矽。

【0006】 平行性之一個形式可藉助於一處理器達成，該處理器包括相同晶片(即，相同晶粒)上之多個微磚之一配置，各微磚包括其自身單獨各自處理單元及記憶體(包含程式記憶體及資料記憶體)。因此，程式碼之單獨部分可在該等微磚之不同者上並行運行。該等微磚經由一晶片上互連件連接在一起，該晶片上互連件使程式碼能夠在不同微磚上運行以在微磚之間通信。在一些情況中，各微磚上之處理單元可自身運行微磚上之多個並行執行緒，各微磚具有如上文所描述之其自身各自組之內文及對應管線以便支援相同微磚上之多個執行緒透過相同管線交錯。

【0007】 一般而言，在不同微磚上運行之一程式之部分之間可存在

相依性。因此需要一種防止一個微磚上之一段程式碼在其所取決於之可藉由另一微磚上之另一段程式碼獲得之資料前面運行之技術。存在用於達成此之數個可能方案，但本文中所關注之方案被稱為「整體同步平行」(Bulk Synchronous Parallel；BSP)。根據BSP，各微磚在一交替循環中執行一運算階段及一交換階段。在運算階段期間，各微磚在微磚上本端執行一或多個運算任務，但不會將其運算之任何結果與微磚之任何其他者通信。在交換階段中，容許各微磚將來自先前運算階段之運算之一或多個結果交換至群組中之微磚之一或多個其他者及/或自群組中之微磚之一或多個其他者交換，但尚未進行至下一運算階段。此外，根據BSP原理，將一屏障同步放置於自運算階段過渡至交換階段之接點，或自交換階段過渡至運算階段之接點，或兩者處。即，以下任一者：(a)需要所有微磚在容許群組中之任一者進行至下一交換階段之前完成其等各自運算階段；或(b)需要群組中之所有微磚在容許群組中之任何微磚進行至下一運算階段之前完成其等各自交換階段；或(c)兩者。在一些案例中，可容許執行運算之一微磚與其他系統資源(諸如一網路卡或儲存磁碟)通信，只要不涉及與群組中之其他微磚之通信。

【0008】 在機器智慧中找到多執行緒及/或多微磚處理之一例示性使用。如熟習機器智慧技術者將熟知，一機器智慧演算法係基於對可藉由多個互連節點之一圖表表示之一「知識模型」執行迭代更新。各節點表示其輸入之一函數。一些節點接收該圖表之輸入且一些自一或多個其他節點接收輸入，而一些節點之輸出形成其他節點之輸入，且一些節點之輸出提供圖表之輸出(且在一些情況中，一給定節點可甚至具有所有以下此等：圖表之輸入、來自圖表之輸出及至其他節點之連接)。此外，各節點處之函

數係藉由一或多個各自參數(例如，權重)參數化。在一學習階段期間，目的係基於一組經驗輸入資料找到用於各種參數之值使得圖表作為一整體將針對一系列可能輸入產生一所要輸出。此項技術中已知用於完成此之各種演算法，諸如基於隨機梯度下降法之一反向傳播演算法。在基於輸入資料之多次迭代中，逐步調諧參數以減少其等錯誤，且因此圖表收斂於一解。在一後續階段中，可接著使用經學習模型以在給定一組指定輸入之情況下對輸出進行預測或在給定一組指定輸出之情況下進行關於輸入(原因)之推斷。

【0009】 各節點之實施方案將涉及資料之處理，且圖表之互連對應於將在節點之間交換之資料。通常，各節點之至少一些處理可獨立於圖表中之節點之一些或所有其他者執行，且因此大圖表為並行性及/或平行性提供巨大機會。

【發明內容】

【0010】 下文描述具有經開發以解決機器智慧應用中所涉及之運算中所出現之問題之一架構之一處理器之組件。本文中所描述之處理器可用作一工作加速器，即，其自運行於一主機電腦上之一應用程式接收一工作負載，該工作負載通常呈待處理之非常大資料集之形式(諸如藉由一機器智慧演算法用於學習一知識模型之大經驗資料集，或使用一先前學習之知識模型自其執行一預測或推斷之資料)。本文中所提出之架構之一目的係高效地處理此等非常大量之資料。該處理器架構已經開發以用於處理機器智慧中所涉及之工作負載。但是，將明白，所揭示架構亦可適用於共用類似特性之其他工作負載。

【0011】 當經由多個微磚執行一程式之不同部分時，可需要執行一

屏障同步以將多個微磚帶至一共同執行點。亦可需要在所有微磚已完成一運算階段之後判定程式作為一整體之一狀態，例如，判定是否應向主機報告一異常，或作出一分支決策以判定是否分支至該程式之下一部分或繼續反覆當前部分。例如，若一微磚群組之各者執行一機器智慧圖表之一各自子圖表之運算，則可需要判定該子圖表之節點是否已全部滿足指示該圖表收斂於一解之一特定條件。使用現有技術作出此一判定需要使用通用指令程式化之數個步驟。

【0012】本文中應認知，將需要針對大規模多執行緒應用(諸如機器學習)定製一處理器之指令集。根據本發明，此係藉由僅在一微磚群組中之所有微磚已完成當前BSP運算階段之後提供用於驗證該群組之一結果之一專用機器碼指令而達成，因此提供同步微磚且同時以減少之延時及較低程式碼密度來判定多個執行緒之一總體結果之能力。

【0013】根據本文中所揭示之一態樣，提供一種包括一微磚配置及用於在該等微磚之間通信之一互連件之處理系統，其中：

各微磚包括用於執行機器碼指令之一執行單元，各機器碼指令為該處理器之一指令集中之一組預定義指令類型之一例項，該指令集中之各指令類型藉由一對應操作碼及用於獲取零個或多個運算元之零個或多個運算元欄位所定義；

該互連件可操作以根據一整體同步平行方案在一些或所有該等微磚之一群組之間進行通信，藉此該群組中之該等微磚之各者執行一微磚上運算階段，接著執行一微磚間交換階段，其中阻止該交換階段直至該群組中之所有該等微磚已完成該運算階段，其中該群組中之各微磚在完成該運算階段之後具有一局域退出狀態；

該指令集包括用於藉由該群組中之各微磚在完成其運算階段之後執行之一同步指令，其中該同步指令之執行引起該執行單元將一同步請求發送至該互連件中之硬體邏輯；及

該互連件中之該邏輯經組態以將該等局域退出狀態彙總成一全域退出狀態，且回應於如藉由自該群組中之所有該等微磚接收該同步請求指示之藉由該群組中之所有該等微磚完成該運算階段，將該全域退出狀態儲存於該群組中之該等微磚之各者上之一全域退出狀態暫存器中，藉此使運行於該群組中之該等微磚之各者上之程式碼之一部分可存取該全域退出狀態。

【0014】 根據本文中所揭示之另一態樣，提供一種包括一微磚配置及用於在該等微磚之間通信之一互連件之處理系統，其中：

各微磚包括用於執行機器碼指令之一各自執行單元，各機器碼指令為該處理器之一指令集中之一組預定義指令類型之一例項，該指令集中之各指令類型藉由一對應操作碼及用於獲取零個或多個運算元之零個或多個運算元欄位所定義；

該互連件包括用於在一些或所有該等微磚之一群組之間協調之呈專用硬體邏輯之形式之同步邏輯；

該指令集包括一同步指令，其中每一各自微磚上之該各自執行單元經組態以便在透過該各自執行單元執行該同步指令之一例項之情況下回應於該同步指令之操作碼引起將一同步請求之一例項自該各自微磚傳輸至該互連件中之同步邏輯且中止該各自微磚上之指令發出直至自該同步邏輯接收回一同步確認；

各微磚具有用於儲存該微磚在完成一各自運算階段之後之一局域退

出狀態之一局域退出狀態暫存器；

該同步邏輯經組態以將該群組中之該等微磚之該等局域退出狀態彙總成一全域退出狀態；且

該同步邏輯進一步經組態以便回應於自該群組之所有該等微磚接收該同步請求之一例項，將該同步確認傳回至該群組中之該等微磚之各者且藉此容許該指令發出繼續，且將該全域退出狀態儲存於該群組中之該等微磚之各者上之一全域退出狀態暫存器中，藉此使運行於該群組中之該等微磚之各者上之程式碼之一部分可存取該全域退出狀態。

【0015】 在實施例中，各微磚上之執行單元可經組態以回應於執行同步指令而暫停指令發出；且互連件中之邏輯可經組態以便回應於自群組中之所有微磚接收同步請求，將一同步確認信號發送回至群組中之微磚之各者以繼續該指令發出。

【0016】 在實施例中，局域退出狀態及全域退出狀態之各者可為一單位元。

【0017】 在實施例中，彙總可由局域退出狀態之一布林(Boolean) AND或局域退出狀態之一布林OR組成。

【0018】 在替代實施例中，經彙總退出狀態可包括表示一個三元值之至少兩個位元，從而指示局域退出狀態是否全部真、全部假或混合。

【0019】 在實施例中，該群組微磚之各者可包括經配置以表示該微磚之局域退出狀態之一局域退出狀態暫存器。

【0020】 在實施例中，群組中之各微磚可包括：

多個內文暫存器組，各內文暫存器組各經配置以儲存多個執行緒之一各自者之一程式狀態；及

一排程器，其經配置以在交錯時槽之一重複序列中之複數個時槽之各者中排程複數個工作者執行緒之一各自者之執行，其中該等工作者執行緒之各者之程式狀態係儲存於該等內文暫存器組之一各自者中；

其中根據該整體同步平行方案，阻止交換階段直至群組中之所有微磚上之所有工作者執行緒已完成運算階段；

其中各微磚上之局域退出狀態可為藉由該微磚上之工作者執行緒之各者輸出之一個別退出狀態之一彙總；且

其中程式碼之該部分可包括該微磚上之多個執行緒之至少一者。

【0021】 在實施例中，群組中之各微磚可包括經組態以執行將該等個別退出狀態彙總成局域退出狀態之硬體邏輯。

【0022】 在實施例中，指令集可包括用於包含於工作者執行緒之各者中之一退出指令，且執行單元可經組態以輸出各自工作者執行緒之個別退出狀態且回應於該退出指令之操作碼終止各自工作者執行緒。

【0023】 在實施例中，個別退出狀態及局域退出狀態之各者可為一單位元，且個別退出狀態之彙總可由個別退出狀態之一布林AND或個別退出狀態之一布林OR組成。

【0024】 在實施例中，局域退出狀態可包括表示一個三元值之至少兩個位元，從而指示該等個別退出狀態是否全部真、全部假或混合。

【0025】 在實施例中，交換階段可經配置以藉由與工作者執行緒分離之一監督器執行緒執行，且該至少一執行緒可包括該監督器執行緒。

【0026】 在實施例中，指令發出之暫停可包括至少暫停自監督器執行緒發出指令直至同步確認。

【0027】 在實施例中，各微磚上之內文暫存器組可包括：多個工作

者內文暫存器組，其等經配置以表示該複數個工作者執行緒之各自者之程式狀態；及一額外監督器內文暫存器組，其包括經配置以表示監督器執行緒之一程式狀態之一組額外暫存器。

【0028】 在實施例中：

監督器執行緒可經配置以藉由在該等時槽之各者中運行而開始；

指令集可進一步包括一轉交指令且執行單元經組態以便回應於該轉交指令之操作碼而將其中執行該轉交指令之該時槽轉交給各自工作者執行緒；及

退出指令可引起將其中執行退出指令之各自時槽交還給監督器執行緒，使得監督器執行緒繼續在該各自時槽中運行。

【0029】 在實施例中，程式碼之該部分可經組態以使用全域退出狀態，一旦有效便可執行取決於該全域退出狀態之一分支決策。

【0030】 在實施例中，處理系統可經程式化以執行一機器智慧演算法，其中一圖表中之各節點具有一或多個各自輸入邊緣及一或多個各自輸出邊緣，其中該等節點中之至少一些節點之輸入邊緣係該等節點之至少一些其他節點之輸出邊緣，各節點包括使其輸出邊緣與其輸入邊緣有關之一各自函數，其中每一各自函數藉由一或多個各自參數參數化，且該等各自參數之各者具有一相關聯錯誤，使得該圖表在一些或所有該等參數中之錯誤減少時收斂於一解；其中該等微磚之各者可模型化包括該圖表中之該等節點之一子集之一各自子圖表，且該等局域退出狀態之各者可用於指示該各自子圖表中之該等節點之一或多個參數中之錯誤是否已滿足一預定條件。

【0031】 在實施例中，該群組可至少部分藉由同步指令之一運算元

選擇。

【0032】 在實施例中，該同步指令之該運算元可選擇是否僅包含該群組中之相同晶片上之微磚或不同晶片上之微磚。

【0033】 在實施例中，該同步指令之該運算元可自分組之不同階層式層級中選擇該群組。

【0034】 在實施例中，指令集可進一步包含一放棄指令，該放棄指令引起其上執行該放棄指令之微磚選擇退出該群組。

【0035】 根據本文中所揭示之另一態樣，提供一種操作一處理系統之方法，該處理系統包括一微磚配置及用於在該等微磚之間通信之一互連件，其中各微磚包括用於執行機器碼指令之一執行單元，各機器碼指令為該處理器之一指令集中之一組預定義指令類型之一例項，該指令集中之各指令類型係由一對應操作碼及用於獲取零個或多個運算元之零個或多個運算元欄位所定義；該方法包括：

經由該互連件根據一整體同步平行方案在一些或所有該等微磚之一群組之間進行通信，藉此該群組中之該等微磚之各者執行一微磚上運算階段，接著執行一微磚間交換階段，其中阻止該交換階段直至該群組中之所有該等微磚已完成該運算階段，其中該群組中之各微磚在完成該運算階段之後具有一局域退出狀態；

其中該指令集包括用於藉由該群組中之各微磚在完成其運算階段之後執行之一同步指令，其中該同步指令之執行引起該執行單元將一同步請求發送至該互連件中之硬體邏輯；且

該方法包括回應於如藉由自該群組中之所有該等微磚接收該同步請求指示之藉由該群組中之所有該等微磚完成該運算階段，觸發該互連件中

之該邏輯以將該等局域退出狀態彙總成一全域退出狀態，且將該全域退出狀態儲存於該群組中之該等微磚之各者上之一全域退出狀態暫存器中，藉此使運行於該群組中之該等微磚之各者上之程式碼之一部分可存取該全域退出狀態。

【0036】 根據本文中所揭示之另一態樣，提供一種電腦程式產品，該電腦程式產品包括體現於電腦可讀儲存器上且經組態以執行於本文中所揭示之任何實施例之處理系統上之程式碼，該程式碼包括用於在群組中之各微磚上執行之在各部分中包含同步指令之一例項之一部分。

【圖式簡單說明】

【0037】

為幫助理解本發明及展示可如何實施實施例，藉由實例參考隨附圖式，其中：

圖1係多執行緒處理單元之一示意性方塊圖，

圖2係複數個執行緒內文之一示意性方塊圖，

圖3示意性地繪示交錯之執行時槽之一方案，

圖4示意性地繪示一監督器執行緒及複數個工作者執行緒，

圖5係用於彙總多個執行緒之退出狀態之邏輯之一示意圖，

圖6示意性地繪示相同微磚上之工作者執行緒中之同步，

圖7係包括多個微磚之一處理器晶片之一示意性方塊圖，

圖8係一整體同步平行(BSP)運算模型之一示意性圖解，

圖9係一BSP模型之另一示意性圖解，

圖10係多執行緒處理單元之間的BSP之一示意性圖解，

圖11係一互連系統之一示意性方塊圖，

圖12係多個互連處理器晶片之系統之一示意性圖解，

圖13係多層BSP方案之一示意性圖解，

圖14係多個處理器晶片之一系統之另一示意性圖解，

圖15係一機器智慧演算法中所使用之一圖表之一示意性圖解，及

圖16繪示用於在晶片之間同步之例示性佈線。

【實施方式】

【0038】 下文描述在其指令集中包含一專用指令之一處理器架構，該專用指令用於執行一屏障同步且同時將跨多個微磚之多個執行緒之退出狀態彙總成一退出狀態暫存器中之一單個經彙總狀態，其中此經彙總退出狀態暫存器存在於每個微磚中且含有用於每個微磚之已經彙總之相同結果。然而，首先參考圖1至圖4描述其中可併入此之一例示性處理器。

【0039】 圖1繪示根據本發明之實施例之一處理器模組4之一實例。例如，處理器模組4可為一相同晶片上之類處理器微磚之一陣列之一個微磚，或可實施為其自身晶片上之一獨立處理器。處理器模組4包括呈一桶形執行緒處理單元之形式之一多執行緒處理單元10，及一本端記憶體11 (即，在一多微磚陣列之情況中在相同微磚上，或在一單處理器晶片之情況中在相同晶片上)。一桶形執行緒處理單元係其中管線之執行時間被劃分成交錯時槽之一重複序列(各交錯時槽可由一給定執行緒所擁有)之多執行緒處理單元之一類型。稍後將對此進行更詳細論述。記憶體11包括一指令記憶體12及一資料記憶體22 (其等可實施於不同可定址記憶體單元或相同可定址記憶體單元之不同區域中)。指令記憶體12儲存待藉由處理單元10執行之機器碼，而資料記憶體22儲存待藉由該經執行程式碼操作之資料及(例如，作為此等操作之結果)藉由該經執行程式碼輸出之資料兩者。

【0040】 記憶體12儲存一程式之各種不同執行緒，各執行緒包括用於執行一或若干特定任務之指令之一各自序列。應注意，如本文中引用之一指令意謂一機器碼指令，即，由一單個操作碼及零個或多個運算元組成之處理器之指令集之基本指令之一者之一例項。

【0041】 本文中所描述之程式包括複數個工作者執行緒，及可結構化為一或多個監督器執行緒之一監督器子程式。稍後將對此等進行更詳細論述。在實施例中，一些或所有該等工作者執行緒之各者採用一各自「標碼串(codelet)」之形式。標碼串係一特定類型之執行緒，有時亦被稱為「原子」執行緒。其具有其從執行緒開始(從經啟動之時間)執行所需之所有輸入資訊，即，其在經啟動之後不會自程式之任何其他部分或自記憶體獲取任何輸入。此外，程式之其他部分不會使用執行緒之任何輸出(結果)直至該執行緒已終止(完成)。除非其遇到一錯誤，否則將保證完成。應注意，一些文獻亦將一標碼串定義為無狀態的，即，若運行兩次，則其不可自其第一次運行繼承任何資訊，但在此處不採用該額外定義。亦應注意，並非所有工作者執行緒需要為標碼串(原子)，且在實施例中，一些或所有工作者可代替性地能夠彼此通信。

【0042】 在處理單元10內，來自指令記憶體12之執行緒之多個不同者可透過一單個執行管線13交錯(但通常僅儲存於指令記憶體中之總執行緒之一子集可在整體程式中之任何給定點處交錯)。多執行緒處理單元10包括：複數個內文暫存器檔案26，其等各經配置以表示將同時執行之執行緒之一不同各自者之狀態(內文)；一共用執行管線13，其為該等同時執行之執行緒所共有；及一排程器24，其用於排程該等並行執行緒以依一交錯方式(較佳依一循環方式)透過該共用管線執行。處理單元10連接至為複數

個執行緒所共有之一共用指令記憶體12及亦為複數個執行緒所共有之一共用資料記憶體22。

【0043】 執行管線13包括一提取級14、一解碼級16及一執行級18，執行級18包括可執行如藉由指令集架構定義之算術及邏輯運算、位址運算、載入及儲存操作及其他操作之一執行單元。內文暫存器檔案26之各者包括用於表示一各自執行緒之程式狀態之一組各自暫存器。

【0044】 在圖2中示意性地繪示組成內文暫存器檔案26之各者之暫存器之一實例。內文暫存器檔案26之各者包括各自一或多個控制暫存器28，一或多個控制暫存器28包括用於各自執行緒之至少一程式計數器(PC) (用於追蹤當前執行執行緒之指令位址)，且在實施例中亦包括記錄各自執行緒之一當前狀態(諸如該執行緒當前是否正運行或(例如)因為其遇到一錯誤而暫停)之一組一或多個狀態暫存器(SR)。內文暫存器檔案26之各者亦包括用於暫時保存藉由各自執行緒執行之指令之運算元(即，藉由在執行時各自執行緒之指令之操作碼所操作或由該等操作碼所定義之操作所引起之值)之一組各自運算元暫存器(OP) 32。將瞭解，內文暫存器檔案26之各者可視需要包括各自一或多個其他類型之暫存器(未展示)。亦應注意，雖然術語「暫存器檔案」有時用於指代一共同位址空間中之一暫存器群組，然在本發明中未必如此且硬體內文26之各者(暫存器組26之各者表示各內文)可更一般地包括一或多個此等暫存器檔案。

【0045】 如稍後將更詳細論述，所揭示之配置具有用於可同時執行之數目M個(在所繪示之實例中M=3，但此並不具限制性)執行緒之各者之一個工作者內文暫存器檔案CX0、...、CX(M-1)，及一個額外監督器內文暫存器檔案CXS。該等工作者內文暫存器檔案經保留以用於儲存工作者

執行緒之內文，且該監督器內文暫存器檔案經保留以用於儲存一監督器執行緒之內文。應注意，在實施例中，監督器內文係專用的，因為其具有與工作者之各者不同之數目個暫存器。工作者內文之各者較佳具有彼此相同數目個狀態暫存器及運算元暫存器。在實施例中，監督器內文可具有少於工作者之各者之運算元暫存器。工作者內文可具有而監督器不具有之運算元暫存器之實例包含：浮點暫存器、累積暫存器及/或專用權重暫存器(用於保存一神經網路之權重)。在實施例中，監督器亦可具有不同數目個狀態暫存器。此外，在實施例中，處理器模組4之指令集架構可經組態使得工作者執行緒及(若干)監督器執行緒執行一些不同類型之指令但亦共用一些指令類型。

【0046】 提取級14經連接以便在排程器24之控制下自指令記憶體12提取待執行之指令。排程器24經組態以控制提取級14在時槽之一重複序列中自一組同時執行之執行緒之各者依序提取一指令，從而將管線13之資源劃分至複數個時間上交錯之時槽中，如稍後將更詳細論述。例如，排程方案可循環或加權循環的。以此一方式操作之一處理器之另一術語係一桶形執行緒處理器。

【0047】 在一些實施例中，排程器24可存取各執行緒之狀態暫存器SR之一者以指示該執行緒是否暫停，使得排程器24實際上控制提取級14以僅提取當前在作用中之該等執行緒之指令。在實施例中，較佳的是，各時槽(及對應內文暫存器檔案)始終由一執行緒或另一執行緒所擁有，即，各時槽始終由某一執行緒佔用，且各時槽始終包含於排程器24之序列中；但佔用任何給定時槽之執行緒可能恰好在該時間暫停，在此情況中在序列到達該時槽時，忽略針對各自執行緒之指令提取。替代性地，例如，不排

除在替代較不佳實施方案中，一些時槽可暫時空置且自經排程序列排除。在引用執行單元可操作以交錯之時槽之數目或此類似者之情況下，此係指執行單元能夠同時執行之時槽之最大數目，即，執行單元之硬體支援之並行時槽之數目。

【0048】 提取級14存取內文之各者之程式計數器(PC)。對於每一各自執行緒，提取級14自程式記憶體12中之下一位址提取該執行緒之下一指令，如藉由程式計數器指示。程式計數器遞增各執行循環，除非藉由一分支指令分支。提取級14接著將經提取指令傳遞至解碼級16以待解碼，且解碼級16接著將經解碼指令之一指示連同該指令中所指定之任何運算元暫存器32之經解碼位址一起傳遞至執行單元18，以執行該指令。執行單元18存取運算元暫存器32及控制暫存器28，其可在執行指令時基於經解碼之暫存器位址使用運算元暫存器32及控制暫存器28，諸如在一算術指令之情況中(例如，藉由使兩個運算元暫存器中之值相加、相乘、相減或相除且將結果輸出至各自執行緒之另一運算元暫存器)。或若指令定義一記憶體存取(載入或儲存)，則執行單元18之載入/儲存邏輯根據該指令將來自資料記憶體之一值載入至各自執行緒之一運算元暫存器中，或將來自各自執行緒之一運算元暫存器之一值儲存至資料記憶體22中。或若指令定義一分支或一狀態改變，則執行單元相應地改變程式計數器PC或狀態暫存器SR之一者中之值。應注意，在藉由執行單元18執行一個執行緒之指令時，可藉由解碼級16解碼來自交錯序列中之下一時槽中之執行緒之一指令；及/或在藉由解碼級16解碼一個指令時，可藉由提取級14提取來自此後之下一時槽中之執行緒之指令(但一般而言，本發明之範疇並不限於每時槽一個指令，例如，在替代案例中，可每時槽自一給定執行緒發出一批

兩個或兩個以上指令)。因此，根據已知桶形執行緒處理技術，交錯有利地隱藏管線13中之延時。

【0049】在圖3中繪示藉由排程器24實施之交錯方案之一實例。此處並行執行緒係根據一循環方案交錯，藉此在該方案之各遍次內，該遍次被劃分成各用於執行一各自執行緒之一時槽序列S0、S1、S2、…。通常各時槽係一個處理器循環長且不同時槽之大小係均勻的，但在所有可能實施例中不一定如此，例如，一加權循環方案亦可行，藉此一些執行緒在每執行遍次獲得比其他執行緒更多之循環。一般而言，桶形執行緒可採用一均勻循環或一加權循環排程，其中在加權循環排程情況中，加權可為固定或適應性的。

【0050】無論每執行遍次之序列如何，此型樣接著重複，各遍次包括時槽之各者之一各自例項。因此應注意，如本文中所引用之一時槽意謂序列中之重複分配位置，而非序列之一給定重複中之時槽之一特定例項。換言之，排程器24將管線31之執行循環分派至複數個時間上交錯(分時多工化)之執行通道中，其中各執行通道包括時槽之一重複序列中之一各自時槽之一復現。在所繪示之實施例中，存在四個時槽，但此僅用於闡釋性目的且其他數目亦可行。例如，在一項較佳實施例中，實際上存在六個時槽。

【0051】無論循環方案被劃分成之時槽之數目如何，根據本發明，處理單元10包括比所存在之時槽多一個之內文暫存器檔案26，即，其支援比其能夠進行桶形執行緒之交錯時槽之數目多一個之內文。

【0052】此係藉由實例繪示於圖2中：若如圖3中所展示存在四個時槽S0、…、S3，則存在此處標記為CX0、CX1、CX2、CX3及CXS之五個

內文暫存器檔案。即，儘管在桶形執行緒方案中僅存在四個執行時槽 S_0 、 $\dots$ 、 S_3 且因此僅可同時執行四個執行緒，然本文中揭示添加一第五內文暫存器檔案 CXS ，其包括一第五程式計數器(PC)、一第五組運算元暫存器32且在實施例中亦包括一第五組一或多個狀態暫存器(SR)。雖然應注意，如所提及在實施例中監督器內文可不同於其他內文 CX_0 、 $\dots$ 、 CX_3 ，且監督器執行緒可支援用於操作執行管線13之一不同指令集。

【0053】 前四個內文 CX_0 、 $\dots$ 、 CX_3 之各者係用於表示當前指派給四個執行時槽 S_0 、 $\dots$ 、 S_3 之一者以用於執行程式設計員所需之任何特定應用運算任務之複數個「工作者執行緒」(應再次注意，此僅可為如儲存於指令記憶體12中之程式之工作者執行緒之總數目之子集)之一各自者之狀態。然而，第五內文 CXS 經保留以用於一特殊功能，以表示「監督器執行緒」(SV)之狀態，SV之作用係協調工作者執行緒之執行(至少在整個程式中之哪一點處指派將在時槽 S_0 、 S_1 、 S_2 、 $\dots$ 之哪一者中執行工作者執行緒 W 之哪一者之意義上)。視需要，監督器執行緒可具有其他「督導程式(overseer)」或協調責任。例如，監督器執行緒可負責執行屏障同步以確保一特定執行順序。例如，在其中一或多個第二執行緒取決於待藉由運行於相同處理器模組4上之一或多個第一執行緒輸出之資料之情況下，監督器可執行一屏障同步以確保直至第一執行緒已完成才開始第二執行緒。及/或，監督器可執行一屏障同步以確保直至一特定外部資料源(諸如另一微磚或處理器晶片)已完成使該資料可用所需之處理才開始處理器模組4上之一或多個執行緒。監督器執行緒亦可用於執行與多個工作者執行緒有關之其他功能性。例如，監督器執行緒可負責將外部資料傳達至處理器模組4 (以接收待藉由該等執行緒之一或多者所作用之外部資料，及/或傳輸藉

由工作者執行緒之一或多者輸出之資料)。一般而言，監督器執行緒可用於提供程式設計員所需之任何種類之督導或協調功能。例如，作為另一實例，監督器可督導微磚本端記憶體12與(陣列6外部之)更廣泛系統(諸如一儲存磁碟或網路卡)中之一或多個資源之間的傳送。

【0054】 當然應注意，四個時槽僅為一實例，且通常在其他實施例中可存在其他數目，使得若每遍次存在最大M個時槽0、...M-1，則處理器模組4包括M+1個內文CX、...、CX(M-1) & CXS (即，一個內文用於可在任何給定時間交錯之各工作者執行緒)及用於監督器之一額外內文。例如，在一個例示性實施方案中存在六個時槽及七個內文。

【0055】 參考圖4，在經交錯時槽之方案中，監督器執行緒SV本身不具有其自身時槽。工作者亦不具有其自身時槽，因為可靈活定義時槽至工作者執行緒之分配。實情係，各時槽具有用於儲存工作者內文之其自身專用內文暫存器檔案(CX0、...、CXM-1)，該工作者內文在時槽分配給工作者時藉由工作者使用，但在時槽分配給監督器時不使用。當一給定時槽分配給監督器時，該時槽代替性地使用監督器之內文暫存器檔案CXS。應注意，監督器始終存取其自身內文且工作者無法佔用監督器內文暫存器檔案CXS。

【0056】 監督器執行緒SV具有在時槽S0、...、S3 (或更一般而言S0、...、SM-1)之任一者及所有者中運行之能力。排程器24經組態以便在程式作為一整體開始時，藉由將監督器執行緒分配給所有時槽而開始，即，因此監督器SV開始在S0、...、S3之所有者中運行。然而，監督器執行緒具有用於在某一後續時間點(立即或在執行一或多個監督器任務之後)將運行該監督器執行緒之時槽之各者暫時轉交給工作者執行緒(例如，最

初在圖4中所展示之實例中之工作者W0、…、W3)之一各自者之一機制。此係藉由監督器執行緒執行一轉交指令(在本文中藉由實例被稱為「RUN」)而達成。在實施例中，此指令採用兩個運算元：指令記憶體12中之一工作者執行緒之一位址及資料記憶體22中之用於該工作者執行緒之一些資料之一位址：

RUN task_addr、data_addr

【0057】工作者執行緒係可彼此同時運行之程式碼之部分，各部分表示待執行之一或多個各自運算任務。資料位址可指定待藉由工作者執行緒作用之一些資料。替代性地，轉交指令可僅採用指定工作者執行緒之位址之一單個運算元，且資料位址可包含於工作者執行緒之程式碼中；或在另一實例中，該單個運算元可指向指定工作者執行緒及資料之位址之一資料結構。如所提及，在實施例中，至少一些工作者可採用標碼串之形式，即，可同時執行之程式碼之原子單元。替代性地或此外，一些工作者不需要為標碼串且可代替性地能夠彼此通信。

【0058】轉交指令(「RUN」)作用於排程器24以便將其中執行此指令自身之當前時槽轉交給藉由運算元指定之工作者執行緒。應注意，在轉交指令中暗示正轉交其中執行此指令之時槽(在機器碼指令之內文中隱式意謂不需要一運算元來指定此，應自操作碼本身隱式地理解)。因此被分送之時槽係其中監督器執行轉交指令之時槽。或換言之，監督器在其分送之相同空間中執行。監督器說「在此位置處運行此段程式碼」，且接著自該時起由相關工作者執行緒(暫時)擁有復現時槽。

【0059】監督器執行緒SV在時槽之一或多個其他者之各者中執行一類似操作，以將一些或所有其時槽分送給工作者執行緒W0、…、W3 (選

自指令記憶體12中之一較大集合 W_0 、 $\dots$ 、 W_j)之不同各自者。一旦其針對最後時槽如此做，便中止監督器(接著稍後將在一工作者 W 交還時槽之一者時在其中斷之處繼續)。

【0060】監督器執行緒 SV 因此能夠將各執行一或多個任務之不同工作者執行緒分配給經交錯執行時槽 S_0 、 $\dots$ 、 S_3 之不同者。當監督器執行緒判定是時候要運行一工作者執行緒時，其使用轉交指令(「 RUN 」)以將此工作者分配給其中執行該 RUN 指令之時槽。

【0061】在一些實施例中，指令集亦包括運行指令之一變體 $RUNALL$ (「運行全部」)。此指令係用於一起啟動全部執行相同程式碼之一組一個以上工作者。在實施例中，此在處理單元之時槽 S_0 、 $\dots$ 、 S_3 (或更一般而言 S_0 、 $\dots$ 、 $S(M-1)$)之每一者中啟動一工作者。

【0062】此外，在一些實施例中， RUN 及/或 $RUNALL$ 指令在執行時亦將來自監督器狀態暫存器 CXS (SR)之一或多者之一些狀態自動複製至藉由 RUN 或 $RUNALL$ 啟動之(若干)工作者執行緒之一或多個對應狀態暫存器。例如，該經複製狀態可包括一或多個模式，諸如一浮點捨入模式(例如，四捨五入為最近或四捨五入為零)及/或一溢出模式(例如，飽和或使用表示無窮大之一單獨值)。經複製狀態或模式接著控制討論中之工作者以根據該經複製狀態或模式操作。在實施例中，工作者可稍後將此覆寫於其自身狀態暫存器中(但不能改變監督器之狀態)。在進一步替代或額外實施例中，工作者可選擇自監督器之一或多個狀態暫存器讀取一些狀態(且又可稍後改變其等自身狀態)。例如，又，此可採用來自監督器狀態暫存器之一模式，諸如一浮點模式或一捨入模式。然而，在實施例中，監督器無法讀取工作者之內文暫存器 CX_0 、 $\dots$ 之任一者。

【0063】一旦經啟動，當前分配之工作者執行緒W0、…、W3之各者進行以執行藉由各自轉交指令中所指定之程式碼中所定義之一或多個運算任務。在此結束時，各自工作者執行緒接著將其中運行該工作者執行緒之時槽交還給監督器執行緒。此係藉由執行一退出指令(「EXIT」)而達成。

【0064】EXIT指令採用至少一運算元且較佳僅採用一單個運算元exit_state (例如，二元值)，以用於程式設計員所需之任何目的以指示在結束之後各自標碼串之一狀態(例如，指示是否滿足一特定條件)：

EXIT exit_state

【0065】EXIT指令作用於排程器24使得其中執行該指令之時槽返回至監督器執行緒。監督器執行緒可接著執行一或多個後續監督器任務(例如，屏障同步及/或與外部資源(諸如其他微磚)交換資料)，及/或繼續執行另一轉交指令以將一新的工作者執行緒(W4等)分配給討論中之時槽。因此又應注意，指令記憶體12中之執行緒之總數目可大於桶形執行緒處理單元10在任一時間可交錯之數目。監督器執行緒SV之作用係排程來自指令記憶體12之工作者執行緒W0、…、Wj之哪些工作者在整個程式中之哪一階段將被指派給排程器24之循環排程中之交錯時槽S0、…、SM之哪些時槽。

【0066】此外，EXIT指令被賦予一進一步特殊功能，即，引起EXIT指令之運算元中所指定之退出狀態與透過相同處理器模組4 (例如，相同微磚)之相同管線13運行之複數個其他工作者執行緒之退出狀態自動彙總(藉由專用硬體邏輯)。因此，在指令中包含用於終止一工作者執行緒之一額外隱式功能。

【0067】在圖5中展示用於達成此之一例示性電路。在此實例中，個別執行緒之退出狀態及經彙總退出狀態各採用一單位元(即，0或1)之形式。處理器模組4包括用於儲存該處理器模組4之經彙總退出狀態之一暫存器38。此暫存器在本文中可被稱為「局域共識」暫存器\$LC (相對於稍後將更詳細論述之在處理器模組4作為類似處理器微磚之一陣列之一者被包含時之全域共識)。在實施例中，此局域共識暫存器\$LC 38係監督器之內文暫存器檔案CXS中之監督器之狀態暫存器之一者。用於執行彙總之邏輯包括一AND閘37，AND閘37經配置以執行(A) EXIT指令之運算元中所指定之退出狀態及(B)局域共識暫存器(\$LC) 38中之當前值之一邏輯AND，且將結果(Q)作為局域彙總之一新值輸出回至局域共識暫存器\$LC 38中。

【0068】在程式中之一合適同步點處，儲存於局域共識暫存器(\$LC) 38中之值最初經重設至一值1。即，在此點之後退出之任何執行緒將促成局域彙總之退出狀態\$LC直至下一重設。若兩個輸入(A、B)係1，則AND閘37之輸出(Q)係1，但若輸入(A、B)之任一者係0，則輸出Q為0。每當執行一EXIT指令時，其退出狀態即與之前已過去(自上次重設之後)之退出狀態彙總。因此藉由圖5中所展示之配置，邏輯保持自上次重設局域共識暫存器(\$LC) 38起藉由一EXIT指令終止之任何工作者執行緒之退出狀態之一運行彙總。在此實例中，該運行彙總係指到目前為止之所有執行緒是否已退出真：來自工作者執行緒之任一者之任何退出狀態0將意謂暫存器38中之彙總變為鎖存至0直至下一重設。在實施例中，監督器SV可在任何時候藉由自局域共識暫存器(\$LC) 38獲取當前值而讀取運行彙總(不需要等待一微磚上同步來如此做)。

【0069】 局域共識暫存器(\$LC) 38中之彙總之重設可藉由監督器SV使用一或多個通用指令對局域共識暫存器(\$LC) 38之暫存器位址執行一PUT (在此實例中將一值1放置於暫存器38)而執行。替代性地，不排除重設可藉由一自動化機制執行(例如，藉由執行本文中稍後所描述之SYNC指令而觸發)。

【0070】 彙總電路37 (在此情況中AND閘)係使用用於形成一布林AND之功能性之電子組件之任何合適組合實施於執行級18之執行單元中之專用硬體電路中。專用電路或硬體意謂具有一硬接線功能之電路，而非使用通用程式碼程式化於軟體中之電路。局域退出狀態之更新係藉由執行特殊EXIT指令而觸發，此特殊EXIT指令係處理器模組4之指令集中之基本機器碼指令之一者，具有彙總退出狀態之固有功能性。又，局域彙總係儲存於一控制暫存器38中，意謂其值可藉由運行於管線上之程式碼存取但不可藉由載入儲存單元(LSU)用於儲存任何通用資料之一專用儲存件(在實施例中，一單個儲存位元)。代替性地，保存於一控制暫存器中之資料之功能在此情況中固定於儲存局域彙總之退出狀態之功能。較佳地，局域共識暫存器(\$LC) 38形成處理器模組4上(例如，微磚上)之控制暫存器之一者(監督器可藉由執行一GET指令而存取該控制暫存器之值且可藉由執行一PUT指令而設定該值)。

【0071】 應注意，圖5中所展示之電路僅為一實例。一等效電路將用一OR閘取代AND閘37且反轉軟體中之退出狀態0及1之解譯，即，0→真，1→假(其中暫存器38在各同步點重設為0而非1)。等效地，若用一OR閘取代AND閘但不反轉退出狀態之解譯，亦不反轉重設值，則\$LC中之經彙總狀態將記錄任何(而非所有)工作者狀態是否以狀態1退出。在其他實

施例中，退出狀態不需要為單位元。例如，各個別工作者之退出狀態可為一單位元，但經彙總退出狀態\$LC可包括表示如下之一三元狀態之兩個位元：所有工作者以狀態1退出，所有工作者以狀態0退出，或工作者之退出狀態係混合的。作為用於實施此之邏輯之一實例，編碼該三元值之該兩個位元之一者可為個別退出狀態之一布林AND (或OR)，且三元值之另一位元可為個別退出狀態之一布林OR。指示工作者之退出狀態係混合之第三經編碼情況可接著形成為此兩個位元之XOR。

【0072】 退出狀態可用於表示程式設計員想要之任何內容，但一個特別設想之實例係使用一退出狀態1以指示各自工作者執行緒已在一「成功」或「真」狀態中退出，而一退出狀態0指示各自工作者執行緒在一「不成功」或「假」狀態中退出(或若彙總電路37執行一OR而非一AND且暫存器\$LC 38最初重設至0，則反之亦然)。例如，考量其中各工作者執行緒執行具有一相關聯條件(諸如指示一機器智慧演算法之圖表中之一各自節點之一或多個參數中之(若干)錯誤是否落在根據一預定度量之一可接受位準內之一條件)之一運算之一應用。在此情況中，一邏輯位準(例如，1)之一個別退出狀態可用於指示滿足條件(例如，節點之一或多個參數中之錯誤或若干錯誤係在根據某一度量之一可接受位準內)；而相反邏輯位準(例如，0)之一個別退出狀態可用於指示未滿足條件(例如，錯誤或若干錯誤不在根據討論中之度量之一可接受位準內)。例如，條件可為放置於一單個參數或各參數上之一錯誤臨限值，或可為與藉由工作者執行緒執行之各自運算相關聯之複數個參數之一更複雜函數。

【0073】 作為另一更複雜實例，工作者之個別退出狀態及經彙總退出狀態可各包括可用於(例如)表示工作者執行緒之結果中之一置信度之兩

個或兩個以上位元。例如，各個別工作者執行緒之退出狀態可表示在該各自工作者執行緒之一結果中之置信度之一概率量測，且彙總邏輯37可用用於執行硬體中之個別置信度位準之一概率彙總之更複雜電路取代。

【0074】 無論程式設計員賦予退出狀態之含義為何，監督器執行緒SV可接著自局域共識暫存器(\$LC) 38獲取經彙總值以判定自上次重設起(例如，在最後同步點)退出之所有工作者執行緒之經彙總退出狀態，例如，以判定是否所有工作者在一成功或真狀態中退出。取決於此經彙總值，監督器執行緒可接著根據程式設計員之設計作出一決策。程式設計員可選擇任意使用他或她想要之局域彙總之退出狀態。例如，監督器執行緒可諮詢局域彙總之退出狀態以便判定由工作者執行緒之一特定子集組成之程式之一特定部分是否已如預期或視需要完成。若否(例如，工作者執行緒之至少一者在一不成功或假狀態中退出)，則可向一主機處理器報告，或可執行包括相同工作者執行緒之程式之部分之另一反覆；但若是(例如，所有工作者執行緒在一成功或真狀態中退出)，則可代替性地分支至包括一或多個新工作者之程式之另一部分。

【0075】 較佳地，監督器執行緒不應存取局域共識暫存器(\$LC) 38中之值直至討論中之所有工作者執行緒已退出，使得儲存於局域共識暫存器(\$LC) 38中之值表示全部所要執行緒之正確、最新彙總狀態。等待此可藉由憑藉監督器執行緒執行之一屏障同步強制執行以等待所有當前運行之局域工作者執行緒(即，相同處理器模組4上之透過相同管線13運行之局域工作者執行緒)退出。即，監督器執行緒重設局域共識暫存器(\$LC) 38，啟動複數個工作者執行緒，且接著在容許監督器繼續自局域共識暫存器(\$LC) 38獲取經彙總退出狀態之前起始一本端屏障同步(在處理模組4本

端、在一微磚本端)以便等待所有未處理工作者執行緒退出。

【0076】 參考圖6，在實施例中，在處理器之指令集中提供一SYNC (同步)指令。該SYNC指令具有引起監督器執行緒SV等待直至所有當前執行工作者W已藉助於一EXIT指令退出之效應。在實施例中，SYNC指令採用一模式作為一運算元(在實施例中其唯一運算元)，該模式指定SYNC是否僅在本端僅相對於在本端運行於與作為其上執行SYNC之部分之監督器相同之處理器模組4 (例如，相同微磚)上之該等工作者執行緒(即，僅透過相同桶形執行緒處理單元10之相同管線13之執行緒)起作用；或是否代替性地其應跨多個微磚或甚至跨多個晶片應用。

SYNC模式 // mode $\in$ {tile, chip, zone_1, zone_2}

【0077】 此將稍後更詳細論述但出於圖6之目的將假定一本端SYNC (「SYNC微磚」，即，一單個微磚內之一同步)。

【0078】 工作者不需要經識別為SYNC指令之運算元，因為暗示接著引起監督器SV自動等待直至一工作者不佔用桶形執行緒處理單元10之時槽S0、S1…。如圖6中所展示，一旦一當前批次之工作者WLn之各者已全部藉由監督器啟動，監督器便可執行一SYNC指令。若監督器SV在桶形執行緒處理單元10之所有時槽S0、…、S3 (在所繪示之實例中全部四個時槽，但此僅為一個例示性實施方案)中啟動工作者W，則一旦該當前批次之工作者執行緒WLn之第一者退出，SYNC即將藉由監督器執行，因此將對至少一時槽之控制交還給該監督器SV。否則若工作者未佔據所有時槽，則SYNC將僅緊接在已啟動當前批次WLn之最後執行緒之後執行。無論如何，在監督器可繼續進行之前，SYNC引起監督器SV等待當前批次之工作者WLn-1之所有其他者執行一EXIT。僅在此之後監督器執行一GET

指令以獲取局域共識暫存器(\$LC) 38之內容。一旦已執行SYNC，便可在硬體中施加藉由監督器執行緒之此等待。即，回應於SYNC指令之操作碼，執行級18之執行單元(EXU)中之邏輯引起提取級14及排程器24暫停發出監督器執行緒之指令直至所有未處理工作者執行緒已執行一EXIT指令。在獲取局域共識暫存器(\$LC) 38 (視需要在其間具有某一其他監督器程式碼)之值之後之某一點，監督器執行一PUT指令以重設局域共識暫存器(\$LC) 38 (在所繪示之實例中重設至1)。

【0079】 又如圖6中所繪示，SYNC指令亦可用於在工作者執行緒之不同相依層WL1、WL2、WL3、…之間放置同步屏障，其中各連續層中之一或多個執行緒取決於藉由其前一層中之一或多個工作者執行緒輸出之資料。藉由監督器執行緒執行之本端SYNC確保直至前一層WLn中之所有工作者執行緒已退出(藉由執行一EXIT指令)才執行下一層WLn+1中之工作者執行緒。

【0080】 如所提及，在實施例中，處理器模組4可實施為形成一個多微磚處理器之一互連微磚陣列之一者，其中微磚之各者可如上文關於圖1至圖6所描述般組態。

【0081】 此在展示一單晶片處理器2 (即，一單晶粒)之圖7中進一步繪示，單晶片處理器2包括多個處理器微磚4之一陣列6及在微磚4之間連接之一晶片上互連件34。晶片2可單獨實施於其自身單晶片積體電路封裝上，或實施為封裝於相同IC封裝中之多個晶粒之一者。該晶片上互連件在本文中亦可被稱為「交換組構(fabric)」34，因為其使微磚4能夠彼此交換資料。各微磚4包括各如上文關於圖1至圖6所描述般配置之桶形執行緒處理單元10及記憶體11之一各自例項。例如，藉由圖解，晶片2可包括大約

數百個微磚4或甚至超過一千個。為完整性，亦應注意，如本文中所引用之一「陣列」並不一定意味微磚4之任何特定數目個尺寸或實體佈局。

【0082】 在實施例中，各晶片2亦包括使晶片2能夠連接至不同晶片(例如，相同晶片2之一或多個其他例項)上之一或多個其他、外部處理器之一或多個外部鏈路8。此等外部鏈路8可包括以下之任一或多者：用於將晶片2連接至一主機處理器之一或多個晶片至主機鏈路，及/或用於使相同IC封裝或卡上或不同卡上之晶片2之一或多個其他例項連接在一起之一或多個晶片至晶片鏈路。在一個例示性配置中，晶片2自經由晶片至主機鏈路之一者連接至該晶片之一主機處理器(未展示)接收呈待藉由晶片2處理之輸入資料之形式之工作者。晶片2之多個例項可藉由晶片至晶片鏈路一起連接至卡中。因此，取決於主機應用所需之工作負載，一主機可存取架構為一單晶片處理器2或可能配置於多個互連卡上之多個單晶片處理器2之一電腦。

【0083】 互連件34經組態以使陣列6中之不同處理器微磚4能夠在晶片2上彼此通信。然而，除相同微磚4上之執行緒之間可能存在相依性之外，在運行於陣列6中之不同微磚4上之程式之部分之間亦可存在相依性。因此需要一種防止一個微磚4上之一段程式碼在其所取決於之可藉由另一微磚4上之另一段程式碼獲得之資料前面運行之技術。

【0084】 在實施例中，此係藉由實施一整體同步平行(BSP)交換方案而達成，如圖8及圖9中示意性地繪示。

【0085】 根據BSP之一版本，各微磚4在一交替循環中執行藉由微磚之間的一屏障同步30彼此分離之一運算階段52及一交換階段50。在所繪示之情況中，一屏障同步係放置於各運算階段52與隨後交換階段50之

間。在運算階段52期間，各微磚4在微磚上本端執行一或多個運算任務，但不會將此等運算之任何結果與微磚4之任何其他者通信。在交換階段50中，容許各微磚4將來自先前運算階段之運算之一或多個結果交換至群組中之微磚之一或多個其他者及/或自群組中之微磚之一或多個其他者交換，但並不執行任何新運算直至其已自其他微磚4接收任何資料，其(若干)任務對該資料具有相依性。各微磚4也不會將任何資料(惟先前運算階段中運算之資料除外)發送至任何其他微磚。不排除可在交換階段中執行其他操作(諸如內部控制相關操作)。在實施例中，交換階段50並不包含任何非時間確定性運算，但在交換階段50期間可視需要容許少數時間確定性運算。亦應注意，可在運算階段52期間容許執行運算之一微磚4與在經同步之微磚4之陣列外部之其他外部系統資源(例如，一網路卡、磁碟機或場可程式化閘陣列(FPGA))通信，只要此並不涉及與群組內之經同步之其他微磚4之通信。在微磚群組外部之通信可視需要利用BSP機制，但替代性地可不利用BSP機制且可代替性地使用其自身之一些其他同步機制。

【0086】 根據BSP原理，將一屏障同步30放置於自運算階段52過渡至交換階段50之接點，或自交換階段50過渡至運算階段52之接點，或兩者處。即，以下任一者：(a)需要所有微磚4在容許群組中之任一者進行至下一交換階段50之前完成其等各自運算階段52；或(b)需要群組中之所有微磚4在容許群組中之任何微磚進行至下一運算階段52之前完成其等各自交換階段50；或(c)強制執行此等條件之兩者。在全部三個變體中，個別處理器在階段之間交替，且整個總成同步。交換階段及運算階段之序列可接著在多次重複期間重複。在BSP術語學中，交換階段及運算階段之各重複有時被稱為一「超步驟(superstep)」(但應注意，在文獻中術語學並不

總是一致地使用：有時各個別交換階段及運算階段個別地被稱為一超步驟，而在別處，如本文中所採用之術語學中，交換階段及運算階段一起被稱為一超步驟)。

【0087】亦應注意，不排除相同晶片2或不同晶片上之微磚4之多個不同獨立群組可各形成彼此非同步操作之一單獨各自BSP群組，其中僅在各給定群組內施加運算、同步及交換之BSP循環，但各群組獨立於其他群組而施加。即，一多微磚陣列6可能包含各相對於其他此等群組獨立地且非同步地操作之多個內部同步群組(稍後更詳細論述)。在一些實施例中，存在同步及交換之一階層式分組，如稍後將更詳細論述。

【0088】圖9繪示如在陣列6中之一些或所有微磚之一群組4i、4ii、4iii中實施之BSP原理，在此情況中該BSP原理施加：(a)自運算階段52至交換階段50之一屏障同步(參見上文)。應注意，在此配置中，容許一些微磚4開始運算52而一些其他微磚4仍在交換。

【0089】根據本文中所揭示之實施例，可藉由將額外、特殊、專用功能性併入至一機器碼指令中以用於執行屏障同步(即，SYNC指令)而促進此類型之BSP。

【0090】在實施例中，SYNC函數在藉由作為一運算元之一微磚間模式(例如，晶片上模式：SYNC晶片)限定時採用此功能性。

【0091】此係示意性地繪示於圖10中。假使各微磚4包括一多執行緒處理單元10，則各微磚之運算階段52可實際上包括藉由相同微磚4上之多個工作者執行緒W執行之任務(且一給定微磚4上之一給定運算階段52可包括工作者執行緒之一或多個層WL，在多個層之情況中該一或多個層WL可使用具有作為一運算元之本端微磚上模式之SYNC指令藉由內部屏

障同步而分離，如先前所描述)。一旦一給定微磚4上之監督器執行緒SV在當前BSP超步驟中啟動最後工作者執行緒，該微磚4上之監督器便可執行具有設定為運算元之微磚間模式(SYNC晶片)之一SYNC指令。若監督器將在其各自處理單元10之所有時槽中啟動(RUN)工作者執行緒，則一旦將不再需要在當前BSP超步驟中RUN更多工作者之第一時槽交還給監督器，便執行「SYNC晶片」。例如，此可在第一執行緒在最後層WL中EXIT之後或僅在第一工作者執行緒EXIT之後(若僅存在一單個層)發生。另外若非所有時槽用於在當前BSP超步驟中運行工作者，則一旦已啟動需要在當前BSP超步驟中RUN之最後工作者，便可執行「SYNC晶片」。此可在已RUN最後層中之所有工作者時或僅在已RUN所有工作者執行緒(若僅存在一個層)時發生。

【0092】執行級18之執行單元(EXU)經組態以便回應於SYNC指令之操作碼，在藉由晶片上(微磚間)運算元限定時，引起其中執行「SYNC晶片」之監督器執行緒暫停直至陣列6中之所有微磚4已完成運行工作者。此可用於對下一BSP超步驟實施一屏障。即，在晶片2上之所有微磚4通過已該屏障之後，跨微磚程式作為一整體可進展至下一交換階段50。

【0093】圖11給出繪示根據本文中所揭示之實施例之藉由一「SYNC晶片」觸發之邏輯之一示意圖。

【0094】一旦監督器已在當前運算循環52中啟動(RUN)其預期之所有執行緒，其便可執行具有晶片上、微磚間運算元(SYNC晶片)之一SYNC指令。此觸發將在微磚4上之專用同步邏輯39中及在實施於硬體互連件34中之一同步控制器36中觸發之以下功能性。微磚上同步邏輯39及互連件34中之同步控制器36兩者之此功能性經實施於專用硬體電路中，

使得一旦執行SYNC晶片，該功能性之其餘部分便在未執行進一步指令以如此做之情況下繼續進行。

【0095】 首先，微磚上同步邏輯39引起針對討論中之微磚4上之監督器之指令發出以自動暫停(引起提取級14及排程器24中止發出監督器之指令)。一旦本端微磚4上之所有未處理工作者執行緒執行一退出(EXIT)，同步邏輯39便可將一同步請求「sync_req」自動發送至互連件34中之同步控制器36。本端微磚4接著繼續在監督器指令發出暫停之情況下等待。一類似程序亦實施在陣列6中之其他微磚4之各者(各包括其自身之同步邏輯39之例項)上。因此在某一時間點，一旦當前運算階段52中之所有最後工作者已在陣列6中之所有微磚4上退出，同步控制器36便將自陣列6中之所有微磚4接收一各自同步請求(sync_req)。僅在那時，回應於自相同晶片2上之陣列6中之每個微磚4接收sync_req，同步控制器36將一同步確認信號「sync_ack」發送回至微磚4之各者上之同步邏輯39。直至此時，微磚4之各者使其監督器指令發出暫停以等待該同步確認信號(sync_ack)。在接收sync_ack信號之後，微磚4中之同步邏輯39自動取消暫停針對該微磚4上之各自監督器執行緒之監督器指令發出。監督器接著在一後續交換階段50中經由互連件34自由地與其他微磚4繼續交換資料。

【0096】 較佳地，sync_req信號及sync_ack信號係分別經由將各微磚4連接至互連件34中之同步控制器36之一或多個專用同步導線傳輸至同步控制器及自同步控制器接收。

【0097】 此外，根據本文中所揭示之實施例，在SYNC指令中包含一額外功能性。即，至少在一微磚間模式(例如，SYNC晶片)中執行時，SYNC指令亦引起經同步微磚4之各者之局域退出狀態SLC自動彙總於互

連件34中之進一步專用硬體40中。在所展示之實施例中，此邏輯採用(例如)由一串雙輸入AND閘40i、40ii、…形成之一多輸入AND閘(用於陣列6中之各微磚4之一個輸入)之形式，如藉由圖11中之實例所展示。此微磚間彙總邏輯40自陣列中之各微磚4接收局域退出狀態暫存器(局域共識暫存器) \$LC 38中之值，在實施例中，各值為一單位元，且將其等彙總成一單個值(例如，所有局域彙總之退出狀態之一AND)。因此，邏輯跨陣列6中之所有微磚4上之所有執行緒形成一全域彙總之退出狀態。

【0098】 微磚4之各者包括經配置以接收及儲存來自互連件34中之全域彙總邏輯40之全域退出狀態之一全域共識暫存器(\$GC) 42之一各自例項。在實施例中，此係監督器之內文暫存器檔案CXS中之狀態暫存器之另一者。回應於自陣列6中之所有微磚4接收之同步請求(sync_req)，同步控制器36引起彙總邏輯40之輸出(例如，AND之輸出)儲存於各微磚4上之全域共識暫存器(\$GC) 42中(將瞭解，圖11中所展示之「切換」係功能性之一示意性表示且實際上可藉由任何合適數位邏輯實施更新)。一旦繼續監督器指令發出，此暫存器(\$GC) 42便可藉由各自微磚4上之監督器執行緒SV存取。在實施例中，全域共識暫存器\$GC係實施為控制暫存器檔案中之一控制暫存器使得監督器執行緒可藉由一GET指令獲取全域共識暫存器(\$GC) 42中之值。應注意，在更新全域共識暫存器(\$GC) 42之任一者中之值之前，同步邏輯36等待直至自所有微磚4接收sync_req，否則可使尚未完成其運算階段52之部分且因此仍運行之一微磚上之一監督器執行緒存取一錯誤值。

【0099】 全域彙總之退出狀態\$GC使程式能夠判定運行於多個不同微磚4上之程式之部分之一總結果而無需個別地檢查各個別微磚上之各個

別工作者執行緒之狀態。其可出於程式設計員所要之任何目的而使用。例如，在其中全域彙總係一布林AND之圖11中所展示之實例中，此意謂任何輸入為0導致一彙總0，但若所有輸入係1，則彙總係1。即，若使用1來表示一真或成功結果，此意謂若微磚4之任一者之局域退出狀態之任一者為假或不成功，則全域彙總狀態亦將為假或表示一不成功結果。例如，此可用於判定運行於所有微磚上之程式碼之部分是否已全部滿足一預定條件。因此，程式可查詢一單個暫存器(在實施例中，一單位元)以詢問「是否出錯？是或否？」或「圖表中之所有節點是否達到一可接受錯誤位準？是或否？」，而非必須檢查各個別微磚上之個別工作者執行緒之個別狀態(且又，在實施例中，監督器實際上無法查詢工作者之狀態，惟透過退出狀態暫存器38、42除外)。換言之，EXIT及SYNC指令各將多個個別退出狀態縮減為一單個組合狀態。

【0100】 在一個例示性使用情況中，微磚之一或多者上之監督器可向一主機處理器報告全域彙總是否指示一假或不成功結果。作為另一實例，程式可取決於全域退出狀態執行一分支決策。例如，程式檢查全域彙總退出狀態\$GC且基於此判定是否繼續迴圈或是否在別處分支。若全域退出狀態\$GC仍為假或不成功，則程式繼續反覆程式之相同、第一部分，但一旦全域退出狀態\$GC為真或成功，程式便可分支至程式之一第二、不同部分。該分支決策可在各監督器執行緒中個別地執行，或藉由擔任主控監督器之角色且指示其他微磚上之其他從屬監督器之監督器之一者實施(主控監督器角色經組態在軟體中)。

【0101】 應注意，圖11中所展示之彙總邏輯40僅為一實例。在另一等效實例中，AND可用OR取代，且0與1之解譯可反轉(0→真，1→假)。

等效地，若用OR閘取代AND閘但不反轉退出狀態之解譯，亦不反轉重設值，則\$GC中之經彙總狀態將記錄任何(而非所有)微磚是否以局域彙總之狀態1退出。在另一實例中，全域退出狀態\$GC可包括表示如下三元狀態之兩個位元：所有微磚之局域彙總之退出狀態\$LC係狀態1，所有微磚之局域彙總之退出狀態\$LC係狀態0，或微磚之局域彙總之退出狀態\$LC係混合的。作為另一更複雜實例，微磚4之局域退出狀態及全域彙總之退出狀態可各包括可用於(例如)表示微磚4之結果中之一置信度之兩個或兩個以上位元。例如，各個別微磚之局域彙總之退出狀態\$LC可表示在各自微磚4之一結果中之置信度之一統計、概率量測，且全域彙總邏輯40可用用於執行硬體中之個別置信度位準之一統計彙總之更複雜電路取代。

【0102】如先前所提及，在實施例中，晶片2之多個例項可連接在一起以形成跨越多個晶片2之微磚4之一更大陣列。此係繪示於圖12中。一些或所有晶片2可實施於相同IC封裝上或一些或所有晶片2可實施於不同IC封裝上。晶片2藉由一外部互連件72 (經由圖7中所展示之外部鏈路8) 連接在一起。除提供用於在不同晶片上之微磚4之間交換資料之一導管之外，外部交換週邊設備72亦提供用於執行不同晶片2上之微磚4之間之屏障同步及彙總不同晶片2上之微磚4之局域退出狀態之硬體支援。

【0103】在實施例中，SYNC指令可採用其模式運算元之至少一進一步可能值以指定一外部(即，晶片間)同步：SYNC zone_n，其中zone_n表示一外部同步區帶。外部互連件72包括類似於關於圖11所描述之硬體邏輯但在一外部晶片間級之硬體邏輯。當SYNC指令係用其運算元中所指定之兩個或兩個以上晶片2之一外部同步區帶執行時，此引起外部互連件72中之邏輯以類似於關於內部互連件34所描述之方式之一方式但跨該所指定

同步區帶中之多個不同晶片2上之微磚4操作。

【0104】 即，回應於一外部SYNC，暫停監督器指令發出直至外部同步區帶中之所有晶片2上之所有微磚4已完成其等運算階段52且提交一同步請求。此外，外部互連件72中之邏輯彙總跨討論中之區帶中之多個晶片2之所有此等微磚4之局域退出狀態。一旦外部同步區帶中之所有微磚4已作出同步請求，外部互連件72便可將一同步確認傳訊回至微磚4且將跨晶片全域彙總退出狀態儲存至討論中之所有微磚4之全域共識暫存器(\$GC) 42中。回應於該同步確認，區帶中之所有晶片2上之微磚4繼續針對監督器之指令發出。

【0105】 在實施例中，互連件72之功能性可實施於晶片2中，即，邏輯可分佈於晶片2中使得僅需要晶片之間的有線連接(圖11及圖12係示意性的)。

【0106】 所提及之同步區帶內之所有微磚4經由其等各自SYNC指令之模式運算元程式化以指示相同同步區帶。在實施例中，外部互連件72週邊設備中之同步邏輯經組態使得，若歸因於一程式化錯誤或其他錯誤(諸如一記憶體同位錯誤)而使情況並非如此，則一些或所有微磚4不會接收一確認，且因此系統將在下一外部屏障處停止，從而容許一管理外部CPU(例如，主機)干預除錯或系統恢復。在其他實施例中，在其中同步區帶並不匹配之情況中引發一錯誤。然而較佳地，編譯器經組態以確保相同區帶中之微磚在相關時間全部指示相同、正確同步區帶。

【0107】 圖13繪示涉及內部(晶片上)同步及外部(晶片間)同步兩者之一例示性BSP程式流程。如所展示，較佳保持(相同晶片2上之微磚4之間的資料之)內部交換50與(不同晶片2上之微磚4之間的資料之)外部交換

50'分離。此之一原因在於，與僅針對一晶片上同步及交換相比，藉由全域同步進行劃界之跨多個晶片之一全域交換可在延時及負載平衡複雜性方面更「昂貴」。另一可能原因在於，經由內部(晶片上)互連件34交換資料可為時間確定性的，而在實施例中經由外部互連件72交換資料可為非時間確定性的。在此等案例中，使內部交換與外部交換分離可為有用的，使得外部同步及交換程序不會「污染」內部同步及交換。

【0108】 因此，為達成此分離，在實施例中，程式經配置以執行包括以下順序之同步、交換階段及運算階段之一序列：(i)一第一運算階段，接著(ii)一內部屏障同步30，接著(iii)一內部交換階段50，接著(iv)一外部屏障同步80，接著(v)一外部交換階段50'。參見圖13中之晶片2II。在內部交換階段50之後施加外部屏障80，使得程式僅在內部交換50之後進行至外部交換50'。亦應注意，如關於圖12中之晶片2I所展示，視需要可在內部交換(iii)與外部屏障(iv)之間包含一運算階段。整個序列係藉由程式強制執行(例如，藉由編譯器如此產生)，且內部同步及交換並不擴展至另一晶片2上之任何微磚或其他實體。序列(i)至(v) (其中iii與iv之間具有上述選用運算階段)可在一系列整體反覆中重複。每反覆在外部同步及交換之前可存在內部運算、同步及交換(i)至(iii)之多個例項。

【0109】 應注意，在一外部交換50期間，通信並不僅限於外部：一些微磚可僅執行內部交換，一些可僅執行外部交換，且一些可執行一混合交換。亦應注意，如圖13中所展示，一般而言在任何給定BSP超步驟中可具有一空值運算階段52或一空值交換階段50。

【0110】 在一些實施例中，亦如圖13中所展示，一些微磚4可在一運算階段期間執行本端輸入/輸出，例如，其等可與一主機交換資料。

【0111】如圖14中所繪示，在實施例中，SYNC指令之模式可用於指定多個不同可能外部同步區帶之一者(例如，zone_1或zone_2)。在實施例中，此等對應於不同階層層級。即，各較高階層層級92 (例如，區帶2) 包含至少一較低階層層級之兩個或兩個以上區帶91A、91B。在實施例中，僅存在兩個階層層級，但不排除較高數目個巢狀層級。若SYNC指令之運算元係設定至外部同步區帶(SYNC zone_1)之較低階層層級，則相對於僅在與其上執行SYNC之微磚相同之較低層級外部同步區帶中之晶片2上之微磚4執行上述同步及彙總操作。另一方面，若SYNC指令之運算元係設定至外部同步區帶(SYNC zone_2)之較高階層層級，則相對於在與其上執行SYNC之微磚相同之較高層級外部同步區帶中之所有晶片2上之所有微磚自動執行上述同步及彙總操作。在實施例中，同步區帶之最高階層層級包含所有晶片，即，其用於執行一全域同步。當使用多個較低層級區帶時，BSP可在內部施加於各區帶內之(若干)晶片2上之微磚4群組中，但各區帶可相對於彼此非同步操作直至執行一全域同步。

【0112】應注意，在其他實施例中，可藉由SYNC指令之模式指定之同步區帶並不限於在本質上為階層式的。一般而言，一SYNC指令可具有對應於任何種類之分組之模式。例如，該等模式可實現自僅非階層式群組中或階層式分組及一或多個非階層式群組之一混合中選擇(其中至少一群組未完全巢套在另一群組內)。此有利地實現程式設計員或編譯器以最小標碼串密度在相對於彼此非同步之內部同步群組之不同佈局之間選擇之靈活性。

【0113】在圖16中繪示用於在選定同步群組91、92中實施同步之一例示性機制。如所繪示，外部互連件72中之外部同步邏輯76包括與每一

各自晶片2相關聯之一各自同步區塊95。各同步區塊95包括各自閘控邏輯及一各自同步彙總器。該閘控邏輯包括硬體電路，該硬體電路出於同步及退出狀態彙總之目的而以一菊鏈拓撲將晶片2連接在一起，且根據下文傳播同步及退出狀態資訊。該同步彙總器包括經組態以根據下文彙總同步請求(sync_req)及退出狀態之硬體電路。

【0114】 與各晶片2相關聯之各自同步區塊95經連接至其各自晶片2，使得其可偵測藉由該晶片2提出之同步請求(Sync_req)及該晶片2之退出狀態，且使得其可將同步確認(Sync_ack)及全域退出狀態傳回至各自晶片2。與各晶片2相關聯之各自同步區塊95亦經由包括四個同步導線96之一集束（其等之細節稍後將更詳細論述）之一外部同步介面連接至晶片2之至少另一者之同步區塊95。此可為晶片至晶片鏈路8之一者之部分。在不同卡上之晶片2之間的一鏈路之情況下，介面8可(例如)包括一PCI介面且四個同步導線96可藉由重新使用該PCI介面之四個導線而實施。晶片之一些同步區塊95經連接至兩個相鄰晶片2之同步區塊95，各連接經由四個同步導線96之一各自例項。以此方式，晶片2可經由其等同步區塊95連接於一或多個菊鏈中。此使同步請求、同步確認、退出狀態之運行彙總及全域退出狀態能夠沿著該鏈上下傳播。

【0115】 在操作中，對於各同步群組91、92，將與該群組中之晶片2之一者相關聯之同步區塊95設定為用於同步及退出狀態彙總目的之主控裝置，該群組中之其餘部分係用於此目的之從屬裝置。從屬同步區塊95之各者係按其需要傳播用於各同步群組91、92之同步請求、同步確認及退出狀態之方向(例如，左或右)組態(即，該方向朝向主控裝置)。在實施例中，此等設定可藉由軟體(例如)在一初始組態階段(在此之後組態在系統

之整個後續操作期間保持固定)中組態。例如，此可藉由主機處理器組態。替代性地，不排除組態可經硬接線。無論哪種方式，不同同步群組91、92可具有不同主控裝置且一般而言可使一給定晶片2 (或確切地說其同步區塊95)成為一個群組而非其係一成員之另一群組之主控裝置，或成為多個群組之主控裝置。

【0116】 例如，藉由圖解考量圖16之例示性案例。舉例說明，晶片2IV之同步區塊95係設定為一給定同步群組91A之主控裝置。現考量經由其等同步區塊95及導線96最終連接至晶片2IV之晶片2之鏈中之第一晶片2I。當第一晶片2I上之當前運算階段之所有工作者執行緒已執行一EXIT指令且所有(參與)微磚4上之監督器全部已執行指定同步群組91A之一SYNC指令時，第一晶片2I接著將其同步準備就緒傳訊回至其各自相關聯同步區塊95。晶片2I亦將其晶片級經彙總退出狀態(各自晶片2I上之所有參與微磚上之所有退出工作者之彙總)輸出至其各自同步區塊95。作為回應，第一晶片2I之同步區塊95將一同步請求(Sync_req)傳播至鏈中之下一晶片2II之同步區塊95。其亦將第一晶片2I之退出狀態傳播至此下一晶片2II之同步區塊95。此第二晶片2II之同步區塊95等待直至其自身(參與)微磚4之監督器全部已執行指定同步群組91A之一SYNC指令，從而引起第二晶片2II傳訊同步準備就緒。僅在那時，第二晶片之同步區塊95將一同步請求傳播至鏈中之下一(第三)晶片2III之同步區塊95，且亦傳播第一晶片2I之退出狀態與第二2II之退出狀態之一運行彙總。若第二晶片2II在第一晶片2I之前已變為同步準備就緒，則第二晶片2II之同步區塊95應等待第一晶片2I在將一同步請求傳播至第三晶片2III之同步區塊95之前傳訊該同步請求。第三晶片2III之同步區塊95以一類似方式表現，此次彙總來自第

二晶片2II之運行彙總退出狀態以獲得下一運行彙總以向前傳遞等。此繼續針對主控同步區塊(在此實例中晶片2IV之主控同步區塊)。

【0117】 主控裝置之同步區塊95接著基於其接收之運行彙總及其自身晶片2IV之退出狀態判定所有退出狀態之一全域彙總。其沿著鏈將此全域彙總連同同步確認(Sync_ack)傳播回出至所有晶片2。

【0118】 若主控裝置部分沿著一鏈(相對於如上文實例中之一端處)，則同步及退出狀態資訊沿著相反方向在主控裝置之任一側、兩側朝向主控裝置傳播。在此情況中，一旦已接收來自兩側之同步請求，主控裝置僅發出同步確認及全域退出狀態。例如，考量其中晶片2III係群組92之主控裝置之情況。此外，在實施例中，一些晶片2之同步區塊95可連接至三個或三個以上其他晶片2之同步區塊95，從而產生朝向主控裝置之多個鏈分支。各鏈接著如上文所描述般表現，且一旦已接收來自所有鏈之同步請求，主控裝置便僅發出同步確認及全域退出狀態。及/或，晶片2之一或多者可連接至一外部資源(諸如主機處理器、一網路卡、一儲存裝置或一FPGA)。

【0119】 在實施例中，如下般實施同步及退出狀態資訊之傳訊。各對晶片2之間的四個同步導線96之集束包括兩對導線(一第一對96_0及一第二對96_1)。各對包括一同步請求導線之一例項及一同步確認導線之一例項。為傳訊值0之一運行彙總退出狀態，發送晶片2之同步區塊95在傳訊同步請求(sync_req)時使用第一導線對96_0之同步請求導線，或為傳訊值1之一運行彙總，同步區塊95在傳訊同步請求時使用第二導線對96_1之同步請求導線。為傳訊值0之一全域彙總退出狀態，發送晶片2之同步區塊95在傳訊同步確認(sync_ack)時使用第一導線對96_0之同步確認導線，或

為傳訊值1之一全域彙總，同步區塊95在傳訊同步確認時使用第二導線對96_1之同步請求導線。

【0120】 應注意，上文僅為用於傳播同步及退出狀態資訊之機制。實際資料(內容)係藉由(例如)如稍後參考圖16所論述之另一通道傳輸。此外，將瞭解，此僅為一例示性實施方案，且熟習技術者將能夠建置用於實施所揭示之同步及彙總功能性之其他電路(一旦給定本文中所揭示之該功能性之規範)。例如，同步邏輯(圖18中之95)可代替性地使用經由作為專用佈線之替代之互連件34、72攜載之封包。例如，sync_req及/或sync_ack可各以一或多個封包之形式傳輸。

【0121】 不同可能模式中之SYNC指令之功能性係在下文概述。

SYNC微磚(執行本端、微磚上屏障同步)

- 監督器運行模式自執行轉變至等待工作者退出
- 中止用於監督器執行緒之指令發出直至所有工作者執行緒處於非作用中
- 當所有工作者執行緒處於非作用中時，可經由局域共識暫存器(\$LC) 38獲得經彙總工作者退出狀態。

SYNC晶片(執行一內部、晶片上屏障同步)

- 監督器運行模式自執行轉變至等待工作者退出
- 中止用於監督器執行緒之指令發出直至所有工作者執行緒處於非作用中
- 當所有工作者執行緒處於非作用中時：
 - 可經由局域共識暫存器(\$LC) 38獲得經彙總局域工作者退出狀態
 - 將內部同步參與傳訊至交換結構34

- 監督器保持非作用中直至微磚4自交換組構34接收內部同步確認
- 在全域共識暫存器(\$GC) 42中更新全系統退出狀態。

SYNC zone_n (執行跨區帶n之一外部屏障同步)

- 監督器運行模式自執行轉變至等待工作者退出
- 中止用於監督器執行緒之指令發出直至所有工作者執行緒處於非作用中
 - 當所有工作者執行緒處於非作用中時：
 - 可經由局域共識暫存器(\$LC) 38獲得彙總局域工作者退出狀態
 - 將外部同步參與傳訊至外部系統(例如，上述外部互連件72中之同步邏輯)
 - 監督器保持中止直至微磚4自外部系統72接收外部同步確認
- 在全域共識暫存器(\$GC) 42中更新全系統退出狀態。

【0122】如先前所提及，並非所有微磚4一定需要參與同步。在實施例中，如所論述，參與微磚之群組可藉由同步指令之模式運算元設定。然而，此僅容許選擇微磚之預定義群組。本文中應認知，亦可期望能夠在逐微磚基礎上選擇同步參與。因此在實施例中，提供用於選擇哪些個別微磚4參與屏障同步之一替代或額外機制。

【0123】特定言之，此係藉由在處理器指令集中提供待藉由一個或一些微磚4執行之代替SYNC指令之一額外類型之指令而達成。此指令可被稱為「放棄」指令，或「SANS」指令(開始自動非參與式同步)。在實施例中，SANS經保留以供監督器執行緒使用。在實施例中，其採用一單個立即運算元：

SANS n_barriers

【0124】 SANS指令之行為係引起其上執行該SANS指令之微磚放棄當前屏障同步，但不會阻擋等待經指定同步群組中之所有微磚同步之其他微磚。實際上，其稱「別管我」。當執行SANS指令時，SANS指令之操作碼觸發執行級18之執行單元中之邏輯以將同步請求信號(Sync_req)之一例項發送至內部及/或外部同步控制器36、76 (取決於模式)。在實施例中，藉由SANS產生之同步請求適用於包含執行SANS之微磚4之任何同步群組91、92。即，對於此(或若干)局域晶片中之微磚4 (其等必須對同步群組達成一致)接下來使用之任何同步群組，來自執行SANS之微磚4之sync_req將始終有效。

【0125】 因此從同步控制器邏輯36、76及同步群組中之其他微磚4之角度來看，執行SANS指令之微磚4與執行一SYNC指令之一微磚4看似完全一樣，且並不阻撓同步屏障及自同步邏輯36、76發送同步確認信號(Sync_ack)。即，執行SANS而非SYNC之微磚4不阻撓討論中之微磚係其成員之任何同步群組中所涉及之其他微磚4之任一者或使其停止運作。藉由一SANS執行之任何交握針對所有同步群組91、92有效。

【0126】 然而，不同於SYNC指令，SANS指令並不引起監督器指令發出經暫停以等待來自同步邏輯36、76之同步確認信號(Sync_ack)。代替性地，各自微磚可僅繼續不受在執行SYNC指令之其他微磚4之間進行之當前屏障同步所抑制。因此藉由模仿一同步但不等待，SANS指令容許其微磚4加緊處理一或多個任務，同時仍容許其他微磚4同步。

【0127】 運算元n_barriers指定「經發佈」同步之數目，即，微磚不會參與之未來同步點(屏障)之數目。替代性地，不排除在其他實施例中SANS指令並不採用此運算元，且代替性地SANS指令之各執行僅引起一

次性放棄。

【0128】 藉助於SANS指令，特定微磚4可負責執行超出BSP操作排程之直接範疇之外之任務。例如，可期望分配一晶片2內之少量微磚4以起始(及處理)至及/或自主機記憶體之資料傳送，而大部分微磚4被(若干)主要運算任務佔據。在此等案例中，未直接參與主要運算之該等微磚4可使用自動非參與式同步特徵(SANS)聲明其等自身在一段時間內與同步機制有效地斷開連接。當使用此特徵時，不需要一微磚4主動地(即，經由執行SYNC指令)傳訊其對於同步(對於同步區帶之任一者)之準備就緒且在實施例中對經彙總退出狀態作出零貢獻。

【0129】 SANS指令開始或延長一週期，在該週期期間其上執行SANS指令之微磚4將放棄主動參與微磚間同步(或與其他外部資源之同步，若該等外部資源參與同步)。在此週期期間，此微磚4將自動傳訊其對於所有區帶內之同步之準備就緒，且在實施例中亦對全域彙總共識\$GC作出零貢獻。此時間週期可表示為一無符號立即運算元(n_barriers)以指示此微磚4將自動傳訊之額外未來同步點之數目。在執行SANS之後，將藉由其運算元指定之值n_barriers放置於各自微磚4上之一遞減計數暫存器\$ANS_DCOUNT中。此係用於追蹤應作出之額外未來sync_reqs之數目之一段架構狀態。若自動非參與式同步機制當前不在作用中，則將立即執行準備就緒(同步請求sync_req)之第一確證。一旦已完成先前同步(即，在同步確認sync_ack之確證之後)，後續確證將在後台中發生。若自動非參與式同步機制當前處在作用中，則將以一自動方式更新遞減計數計數器暫存器\$ANS_DCOUNT，使得無未計入之同步確認信號。自動非參與式同步機制係實施於專用硬體邏輯中(較佳地其在各微磚4中之一例項)，但在其

他實施例中不排除其可代替性地針對一微磚群組或所有微磚集中實施。

【0130】 關於退出狀態行為，取決於實施方案，實際上存在數個可能性。在實施例中，為獲得全域彙總之退出狀態，同步邏輯36、76僅彙總來自指定同步群組中執行一SYNC指令之該等微磚4而非執行一SANS指令之微磚4 (放棄微磚或若干放棄微磚)之局域退出狀態。替代性地，全域彙總之退出狀態係藉由彙總來自同步群組中執行一SYNC之所有微磚4及執行一SANS之微磚4 (參與微磚4及放棄微磚4兩者)之局域退出狀態而獲得。在後一情況中，藉由(若干)放棄微磚4針對全域彙總輸出之局域退出狀態可為在執行SANS時該微磚之工作者之實際局域彙總之退出狀態，正如SYNC指令般(參見局域共識暫存器\$LC 38之描述)。替代性地，藉由放棄微磚4輸出之局域「退出狀態」可為一預設值，例如，在其中退出狀態係二元之實施例中可為真值(例如，邏輯1)。在其中任何假局域退出狀態引起全域退出狀態為假之實施例中，此防止放棄微磚4干擾全域退出狀態。

【0131】 關於返回全域退出狀態，針對此存在兩種可能性，而無關於放棄微磚是否提交用於產生全域彙總之一局域退出狀態，且無關於該值是否為一實際值或一預設值。即，在一個實施方案中，藉由互連件34、72中之同步邏輯36、76產生之全域彙總退出狀態僅儲存於執行一SYNC指令之參與微磚4 (而非代替性地執行一SANS指令之放棄微磚4)之全域共識暫存器\$GC 42中。在實施例中，代替性地，一預設值係儲存於執行一SANS之(若干)微磚4 (放棄微磚)之全域共識暫存器\$GX 42中。例如，在二元全域退出狀態之情況中，此預設值可為真(例如，邏輯1)。然而，在一替代實施方案中，藉由同步邏輯36、76產生之實際全域彙總係儲存於

執行SYNC指令之參與微磚4及代替性地執行一SANS指令之放棄微磚4兩者之全域共識暫存器\$GC 42中。因此群組中之所有微磚可仍存取全域彙總之退出狀態。

【0132】 圖15繪示本文中所揭示之處理器架構之一例示性應用，即，機器智慧之一應用。

【0133】 如熟習機器智慧技術者將熟知，機器智慧以其中機器智慧演算法學習一知識模型之一學習階段開始。該模型包括互連節點(即，頂點) 102及邊緣(即，鏈路) 104之一圖表。該圖表中之各節點102具有一或多個輸入邊緣及一或多個輸出邊緣。一些節點102之一些輸入邊緣係一些其他節點之輸出邊緣，藉此將該等節點連接在一起以形成圖表。此外，節點102之一或多者之輸入邊緣之一或多者作為一整體形成圖表之輸入，且節點102之一或多者之輸出邊緣之一或多者作為一整體形成圖表之輸出。有時一給定節點可甚至具有以下所有此等：圖表之輸入、來自圖表之輸出及至其他節點之連接。各邊緣104傳送一值或更通常一張量(n 維矩陣)，此等形成分別在節點102之輸入及輸出邊緣上提供至節點102及自節點102提供之輸入及輸出。

【0134】 各節點102表示如在其(若干)輸入邊緣上接收之其一或多個輸入之一函數，其中此函數之結果係提供於(若干)輸出邊緣上之(若干)輸出。各函數係藉由一或多個各自參數(有時被稱為權重，但其等不一定為乘法權重)參數化。一般而言，藉由不同節點102表示之函數可為不同形式之函數及/或可藉由不同參數參數化。

【0135】 此外，各節點之函數之一或多個參數之各者藉由一各自錯誤值特性化。此外，一各自條件可與各節點102之該(等)參數中之該(等)錯

誤相關聯。對於表示藉由一單個參數參數化之一函數之一節點102，該條件可為一簡單臨限值，即，若錯誤在該指定臨限值內則滿足條件，但若錯誤超出該臨限值則不滿足條件。對於藉由一個以上各自參數參數化之一節點102，節點102已達到一可接受錯誤位準之條件可更複雜。例如，僅在該節點102之參數之各者落在各自臨限值內時可滿足條件。作為另一實例，可定義組合用於相同節點102之不同參數中之錯誤之一組合度量，且在該組合度量之值落在一指定臨限值內之條件下可滿足條件，但在其他方面若該組合度量之值超出該臨限值，則不滿足條件(或反之亦然，取決於該度量之定義)。無論條件如何，此給予節點之(若干)參數中之錯誤是否降至低於一特定可接受位準或度之一度量。一般而言，可使用任何合適度量。條件或度量針對所有節點可相同，或針對節點之不同各自者可不同。

【0136】 在學習階段中，演算法接收經驗資料，即，表示圖表之輸入之不同可能組合之多個資料點。隨著接收愈來愈多之經驗資料，演算法基於經驗資料逐步調諧圖表中之各種節點102之參數以便試圖最小化參數中之錯誤。目標係找出參數之值使得圖表之輸出儘可能接近用於一給定輸入之一所要輸出。在圖表作為一整體傾向於此一狀態時，圖表被稱作收斂。在一合適收斂度之後，圖表可接著用於執行預測或或推斷，即，預測某一給定輸入之一結果或推斷某一給定輸出之一原因。

【0137】 學習階段可採用數個不同可能形式。例如，在一監督方法中，輸入經驗資料採用訓練資料(即，對應於已知輸出之輸入)之形式。運用各資料點，演算法可調諧參數使得輸出緊密匹配用於給定輸入之已知輸出。在後續預測階段中，圖表可接著用於將一輸入查詢映射至一近似經預測輸出(或若進行一推斷則反之亦然)。其他方法亦係可行的。例如，在一

非監督方法中，每輸入基準不存在一參考結果之概念，且代替性地由機器智慧演算法來識別輸出資料中之其自身結構。或在一強化方法中，演算法針對輸入經驗資料中之各資料點嘗試至少一個可能輸出，且被告知此輸出是否為正或負(及可能其為正或負之一程度)，例如，贏或輸、或獎勵或懲罰，或此類似者。在多次試驗中，演算法可逐步調諧圖表之參數以能夠預測將導致一正結果之輸入。熟習機器學習技術者將知道用於學習一圖表之各種方法及演算法。

【0138】 根據本文中所揭示之技術之一例示性應用，各工作者執行緒經程式化以執行與一機器智慧圖表中之節點102之一各自個別者相關聯之運算。在此情況中，節點102之間的至少一些邊緣104對應於執行緒之間的資料交換，且一些邊緣104可涉及微磚之間的交換。此外，工作者執行緒之個別退出狀態係由程式設計員用於表示各自節點102是否已滿足其用於收斂該節點之(若干)參數之各自條件，即，是否具有落在錯誤空間中之可接受位準或區域內之(若干)參數中之錯誤。例如，此係實施例之一個例示性使用，其中個別退出狀態之各者係一個別位元且經彙總退出狀態係該等個別退出狀態之一AND (或等效地若將0視為正，則為一OR)；或其中經彙總退出狀態係表示個別退出狀態是否全部真、全部假或混合之一個三元值。因此，藉由檢查退出狀態暫存器38中之一單個暫存器值，程式可判定圖表整體或圖表之至少一子區域是否已收斂於一可接受程度。

【0139】 作為此之另一變體，可使用其中彙總採用個別置信度值之一統計彙總之形式之實施例。在此情況中，各個別退出狀態表示藉由各自執行緒表示之節點之參數已達到一可接受錯誤程度之一置信度(例如，作為一百分比)。經彙總退出狀態可接著用於判定關於圖表或圖表之一子區

域是否已收斂於一可接受程度之一整體置信度。

【0140】 在一多微磚配置6之情況中，各微磚運行圖表之一子圖表。各子圖表包括一監督器子程式，該監督器子程式包括一或多個監督器執行緒及其中一些或所有工作者可採用標碼串之形式之一組工作者執行緒。

【0141】 在此等應用，或實際上其中各工作者執行緒用於表示一圖表中之各自節點之任何基於圖表之應用中，各工作者所包括之「標碼串」可定義為在持久狀態及一個頂點之輸入及/或輸出上操作之一軟體程序，其中標碼串：

- 在一個工作者執行緒暫存器內文上啟動，以藉由執行一「運行」指令之監督器執行緒而在一桶形時槽中運行；
- 運行以在無需與其他標碼串或監督器通信之情況下完成(惟在標碼串退出時返回至監督器除外)；
- 經由藉由該「運行」指令提供之一記憶體指標存取一頂點之持久狀態，及存取記憶體中為該桶形時槽私有之一非持久性工作區；及
- 執行「EXIT」作為其最後指令，因此將其使用之桶形時槽返回至監督器，且將藉由該退出指令指定之退出狀態與監督器可見之微磚之局域退出狀態彙總。

【0142】 更新一圖表(或子圖表)意謂以與由邊緣定義之因果關係一致之任何順序更新各構成頂點一次。更新一頂點意謂在頂點狀態上運行一標碼串。一標碼串係用於頂點之一更新程序，一個標碼串通常與數個頂點相關聯。監督器每頂點執行一個RUN指令，各此指令指定一頂點狀態位址及一標碼串位址。

【0143】 將瞭解，僅藉由實例描述上文實施例。

【0144】 例如，退出狀態彙總機制之適用性並不限於其中對監督器執行緒提供一獨立內文或其中監督器執行緒在一時槽中運行且接著將其時槽轉交給一工作者之上述架構。例如，在另一配置中，監督器可在其自身專用時槽中運行。

【0145】 此外，術語「監督器」及「工作者」並不暗示任何特定責任(惟另有明確說明除外)，且尤其其等自身並不一定限於其中一監督器執行緒將其時槽轉交給一工作者之上述方案等等。一般而言，工作者執行緒可係指某一運算任務所分配至之任何執行緒。監督器可表示負責諸如以下各者之動作之任何種類之督導或協調執行緒：將工作者指派給桶形時槽，及/或在多個執行緒之間執行屏障同步，及/或取決於一個以上執行緒之結果執行任何控制流操作(諸如一分支)。

【0146】 在其中引用交錯時槽之一序列或類似者之情況下，此並不一定暗示所引用之序列組成任何可能或可用時槽。例如，討論中之序列可為所有可能時槽或僅當前在作用中之時槽。並不一定排除可存在當前不包含於經排程序列中之其他潛在時槽。

【0147】 如本文中所使用之術語微磚並不一定限於任何特定拓撲或類似者，且一般而言可係指在一類似模組陣列(通常至少一些模組在相同晶片(即，相同晶粒)上)中包括一處理單元10及對應記憶體11之處理資源之任何模組化單元。

【0148】 此外，本發明之範疇並不限於一時間確定性內部互連或一非時間確定性外部互連。本文中所揭示之同步及彙總機制亦可用於一完全時間確定性配置或一完全非時間確定性配置中。

【0149】此外，在本文中引用在一微磚群組或複數個微磚或類似者之間執行一同步或一彙總之情況下，此未必係指晶片上之所有微磚或系統中之所有微磚，除非明確說明。例如，SYNC及EXIT指令可經組態以僅相對於一給定晶片上之微磚4之一特定子集及/或僅一給定系統中之晶片2之一子集執行同步及彙總；而一給定晶片上之一些其他微磚4及/或一給定系統中之一些其他晶片可不包含於一給定BSP群組中，且可甚至用於與藉由可用群組執行之運算無關之某組完全獨立任務。

【0150】又，雖然上文已描述SYNC指令之特定模式，但更一般而言本發明之範疇並不限於此等模式。例如，上文給定之模式清單並不一定詳盡。或在其他實施例中，SYNC指令可具有較少模式，例如，SYNC不需要支援外部同步之不同階層層級，或不需要在晶片上同步與晶片間同步之間進行區分(即，在一微磚間模式中，無關於是否晶片上或晶片外始終相對於所有微磚起作用)。在又進一步替代實施例中，SYNC指令完全不需要採用一模式作為一運算元。例如，在實施例中，可對同步及退出狀態彙總之不同層級提供SYNC指令之獨立版本(不同操作碼)(諸如不同SYNC指令用於微磚上同步及微磚間、晶片上同步)。或在其他實施例中，可僅對微磚間同步提供一專用SYNC指令(視需要在通用軟體中執行執行緒之間的微磚上同步)。

【0151】此外，同步區帶並不限於階層式(即，在彼此中進行巢套)，且在其他實施例中可選擇同步區帶可由一或多個非階層式群組組成或包含一或多個非階層式群組(該群組之所有微磚未在一單個其他可選擇群組內進行巢套)。

【0152】此外，上述同步方案並不排除在實施例中涉及除多微磚處

理器以外之外部資源，例如，一CPU處理器(諸如主機處理器)，或甚至並非處理器之一或多個組件(諸如一或多個網路卡、儲存裝置及/或FPGA)。例如，一些微磚可選擇參與與一外部系統之資料傳送，其中此等傳送形成該微磚之運算負擔。在此情況中，該等傳送應在下一屏障之前完成。在一些情況中，微磚之退出狀態可取決於與外部資源之通信之一結果，且此資源可替代地影響退出狀態。替代性地或此外，除多微磚處理器以外之資源(例如，主機或一或多個FPGA)可併入至同步網路自身中。即，需要來自此/此等額外資源之一同步信號(諸如Sync_req)以便滿足屏障同步且使微磚進行至下一交換階段。此外，在實施例中，經彙總之全域退出狀態可在彙總中包含(例如)來自一FPGA之外部資源之一退出狀態。

【0153】 一旦給定本文中之揭示內容，熟習此項技術者便可明白所揭示技術之其他應用及變體。本發明之範疇並不受限於所描述實施例而僅受限於隨附發明申請專利範圍。

【符號說明】

【0154】

2	單晶片處理器/晶片/發送晶片
2I	晶片/第一晶片
2II	晶片/第二晶片
2III	晶片/第三晶片
2IV	晶片
4	處理器模組/處理器微磚/微磚
4i至4N	微磚群組
6	陣列/多微磚陣列/多微磚配置

8	外部鏈路/晶片至晶片鏈路/介面
10	多執行緒處理單元/處理單元/桶形執行緒處理單元
11	本端記憶體/記憶體
12	指令記憶體/記憶體/共用指令記憶體/程式記憶體
13	執行管線/管線
14	提取級
16	解碼級
18	執行級/執行單元
22	資料記憶體/共用資料記憶體
24	排程器
26	內文暫存器檔案/硬體內文/暫存器組
28	控制暫存器
30	屏障同步/內部屏障同步
32	運算元暫存器(OP)
34	晶片上互連件/交換結構/互連件/硬體互連件/內部互連件
36	同步控制器/同步邏輯/同步控制器邏輯
37	AND閘/彙總電路/彙總邏輯
38i至38N	暫存器/局域共識暫存器\$LC/局域退出狀態暫存器\$LC/退出狀態暫存器
39	專用同步邏輯/微磚上同步邏輯/同步邏輯
40i至40N	雙輸入AND閘
42i至42N	退出狀態暫存器/全域共識暫存器(\$GC)/暫存器

50	交換階段/內部交換階段
50'	外部交換/外部交換階段
52	運算階段/運算/運算循環
72	外部互連件/外部交換週邊設備/互連件/外部系統
76	同步邏輯/同步控制邏輯
80	外部屏障/外部屏障同步
91A	區帶/同步群組
91B	區帶/同步群組
92	同步群組/群組
95	同步區塊
96	同步導線/導線
96_0	第一導線對
96_1	第二導線對
102	互連節點/節點
104	邊緣
CX0至CX3	工作者內文暫存器檔案
CXS	監督器內文暫存器檔案

【發明申請專利範圍】

【第1項】

一種包括一微磚配置及用於在該等微磚之間通信之一互連件之處理系統，每一微磚包括其自己之處理單元及記憶體，其中：

各微磚之該處理單元包括用於執行機器碼指令之一執行單元，各機器碼指令為該等微磚之一指令集中之一組預定義指令類型之一例項，該指令集中之各指令類型係由一對應操作碼及用於獲取零個或多個運算元之零個或多個運算元欄位定義；

該互連件可操作以根據一整體同步平行方案在一些或所有該等微磚之一群組之間進行通信，藉此該群組中之該等微磚之各者執行一微磚上運算階段，接著執行一微磚間交換階段，其中阻止該交換階段直至該群組中之所有該等微磚已完成該運算階段，其中該群組中之各微磚在完成該運算階段之後具有一局域退出狀態；

該指令集包括用於藉由該群組中之各微磚在完成其運算階段之後執行之一同步指令，其中該同步指令之執行引起該執行單元將一同步請求發送至該互連件中之硬體邏輯；及

該互連件中之該邏輯經組態以將該等局域退出狀態彙總成一全域退出狀態，且回應於如藉由自該群組中之所有該等微磚接收該同步請求指示之藉由該群組中之所有該等微磚完成該運算階段，將該全域退出狀態儲存於該群組中之該等微磚之各者上之一全域退出狀態暫存器中，藉此使運行於該群組中之該等微磚之各者上之程式碼之一部分可存取該全域退出狀態。

【第2項】

如請求項1之處理系統，其中各微磚上之該執行單元經組態以回應於執行該同步指令而暫停指令發出；且其中該互連件中之該邏輯經組態以便回應於自該群組中之所有該等微磚接收該同步請求而將一同步確認信號發送回至該群組中之該等微磚之各者以繼續該指令發出。

【第3項】

如請求項1或2之處理系統，其中該等局域退出狀態及該等全域退出狀態之各者係一單位元。

【第4項】

如請求項3之處理系統，其中該彙總由該等局域退出狀態之一布林AND或該等局域退出狀態之一布林OR組成。

【第5項】

如請求項1或2之處理系統，其中該經彙總退出狀態包括表示一個三元值之至少兩個位元，從而指示該等局域退出狀態是否全部真、全部假或混合。

【第6項】

如請求項1或2之處理系統，其中該群組微磚之各者包括經配置以表示該微磚之該局域退出狀態之一局域退出狀態暫存器。

【第7項】

如請求項1之處理系統，其中該群組中之各微磚包括：

多個內文暫存器組，各內文暫存器組各經配置以儲存多個執行緒之一各自者之一程式狀態；及

一排程器，其經配置以在交錯時槽之一重複序列中之複數個時槽之各者中排程複數個工作者執行緒之一各自者之該執行，其中該等工作者執

行緒之各者之該程式狀態係儲存於該等內文暫存器組之一各自者中；

其中根據該整體同步平行方案，阻止該交換階段直至該群組中之所有該等微磚上之所有該等工作者的執行緒已完成該運算階段；

其中各微磚上之該局域退出狀態係藉由該微磚上之該等工作者的執行緒之各者輸出之一個別退出狀態之一彙總；且

其中程式碼之該部分包括該微磚上之該多個執行緒之至少一者。

【第8項】

如請求項7之處理系統，其中該群組中之各微磚包括經組態以執行將該等個別退出狀態彙總成該局域退出狀態之硬體邏輯。

【第9項】

如請求項8之處理系統，其中該指令集包括用於包含於該等工作者的執行緒之各者中之一退出指令，該執行單元經組態以輸出該各自工作者的執行緒之該個別退出狀態且回應於該退出指令之該操作碼終止該各自工作者的執行緒。

【第10項】

如請求項7、8或9之處理系統，其中該等個別退出狀態及該等局域退出狀態之各者係一單位元，且該等個別退出狀態之該彙總由該等個別退出狀態之一布林AND或該等個別退出狀態之一布林OR組成。

【第11項】

如請求項7、8或9之處理系統，其中該局域退出狀態包括表示一個三元值之至少兩個位元，從而指示該等個別退出狀態是否全部真、全部假或混合。

【第12項】

如請求項7、8或9之處理系統，其中該交換階段經配置以藉由與該等工作者執行緒分離之一監督器執行緒執行，且該至少一執行緒包括該監督器執行緒。

【第13項】

如請求項12之處理系統，其中

該指令發出之該暫停包括至少暫停自該監督器執行緒發出指令直至該同步確認。

【第14項】

如請求項12之處理系統，其中各微磚上之該等內文暫存器組包括：多個工作者內文暫存器組，其等經配置以表示該複數個工作者執行緒之各自者之該程式狀態；及一額外監督器內文暫存器組，其包括經配置以表示該監督器執行緒之一程式狀態之一組額外暫存器。

【第15項】

如請求項14之處理系統，其中：

該監督器執行緒經配置以藉由在該等時槽之各者中運行而開始；

該指令集進一步包括一轉交指令且該執行單元經組態以便回應於該轉交指令之操作碼而將其中執行該轉交指令之時槽轉交給該各自工作者執行緒；及

該退出指令引起將其中執行該退出指令之該各自時槽交還給該監督器執行緒，使得該監督器執行緒繼續在該各自時槽中運行。

【第16項】

如請求項1、2、7、8或9之處理系統，其用該程式碼程式化；其中程式碼之該部分經組態以使用該全域退出狀態，一旦有效便執行取決於該全

域退出狀態之一分支決策。

【第17項】

如請求項1、2、7、8或9之處理系統，其經程式化以執行一機器智慧演算法，其中一圖表中之各節點具有一或多個各自輸入邊緣及一或多個各自輸出邊緣，其中該等節點中之至少一些節點之該等輸入邊緣為該等節點之至少一些其他節點之該等輸出邊緣，各節點包括使其輸出邊緣與其輸入邊緣有關之一各自函數，其中每一各自函數藉由一或多個各自參數參數化，且該等各自參數之各者具有一相關聯錯誤，使得該圖表在一些或所有該等參數中之該等錯誤減少時收斂於一解；

其中該等微磚之各者模型化包括該圖表中之該等節點之一子集之一各自子圖表，且該等局域退出狀態之各者係用於指示該各自子圖表中之該等節點之該一或多個參數中之該等錯誤是否已滿足一預定條件。

【第18項】

如請求項1之處理系統，其中該群組係至少部分藉由該同步指令之一運算元選擇。

【第19項】

如請求項18之處理系統，其中該等同步指令之該運算元選擇是否僅包含該群組中之相同晶片上之微磚或不同晶片上之微磚。

【第20項】

如請求項18或19之處理系統，其中該等同步指令之該運算元自分組之不同階層層級中選擇該群組。

【第21項】

如請求項1、2、7、8、9、18或19之處理系統，其中該指令集進一步

包含一放棄指令，該放棄指令引起其上執行該放棄指令之該微磚選擇退出該群組。

【第22項】

一種操作一處理系統之方法，該處理系統包括一微磚配置及用於在該等微磚之間通信之一互連件，每一微磚包括其自己之處理單元及記憶體，其中各微磚之該處理單元包括用於執行機器碼指令之一執行單元，各機器碼指令為該等微磚之一指令集中之一組預定義指令類型之一例項，該指令集中之各指令類型係由一對應操作碼及用於獲取零個或多個運算元之零個或多個運算元欄位定義；該方法包括：

經由該互連件根據一整體同步平行方案在一些或所有該等微磚之一群組之間進行通信，藉此該群組中之該等微磚之各者執行一微磚上運算階段，接著執行一微磚間交換階段，其中阻止該交換階段直至該群組中之所有該等微磚已完成該運算階段，其中該群組中之各微磚在完成該運算階段之後具有一局域退出狀態；

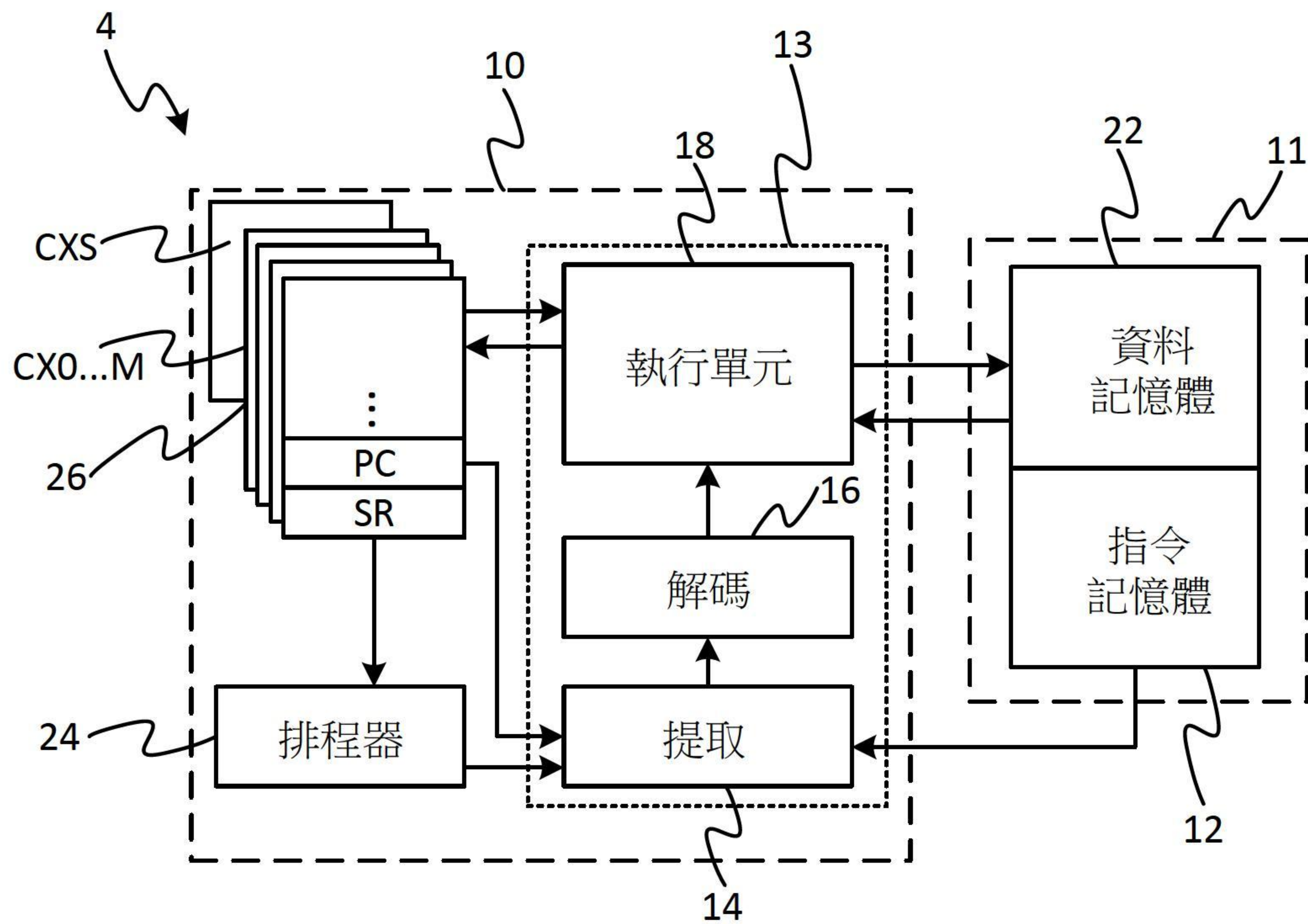
其中該指令集包括用於藉由該群組中之各微磚在完成其運算階段之後執行之一同步指令，其中該同步指令之執行引起該執行單元將一同步請求發送至該互連件中之硬體邏輯；且

該方法包括回應於如藉由自該群組中之所有該等微磚接收該同步請求指示之藉由該群組中之所有該等微磚完成該運算階段而觸發該互連件中之該邏輯以將該等局域退出狀態彙總成一全域退出狀態，且將該全域退出狀態儲存於該群組中之該等微磚之各者上之一全域退出狀態暫存器中，藉此使運行於該群組中之該等微磚之各者上之程式碼之一部分可存取該全域退出狀態。

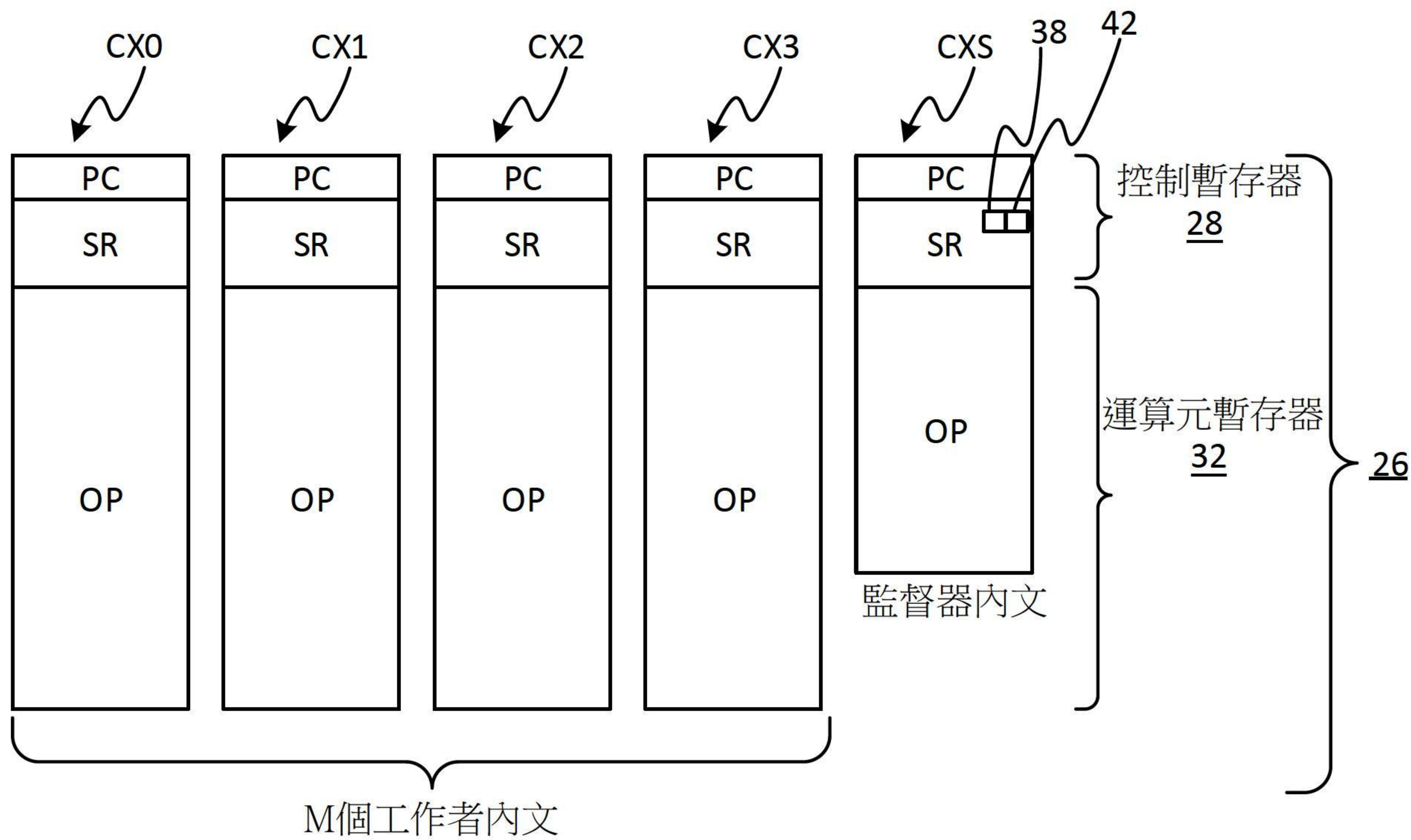
【第23項】

一種電腦程式產品，其包括體現於電腦可讀儲存器上且經組態以執行於如請求項1至21中任一項之處理系統上之程式碼，該程式碼包括用於在群組中之各微磚上執行之一部分，在各部分中包含同步指令之一例項。

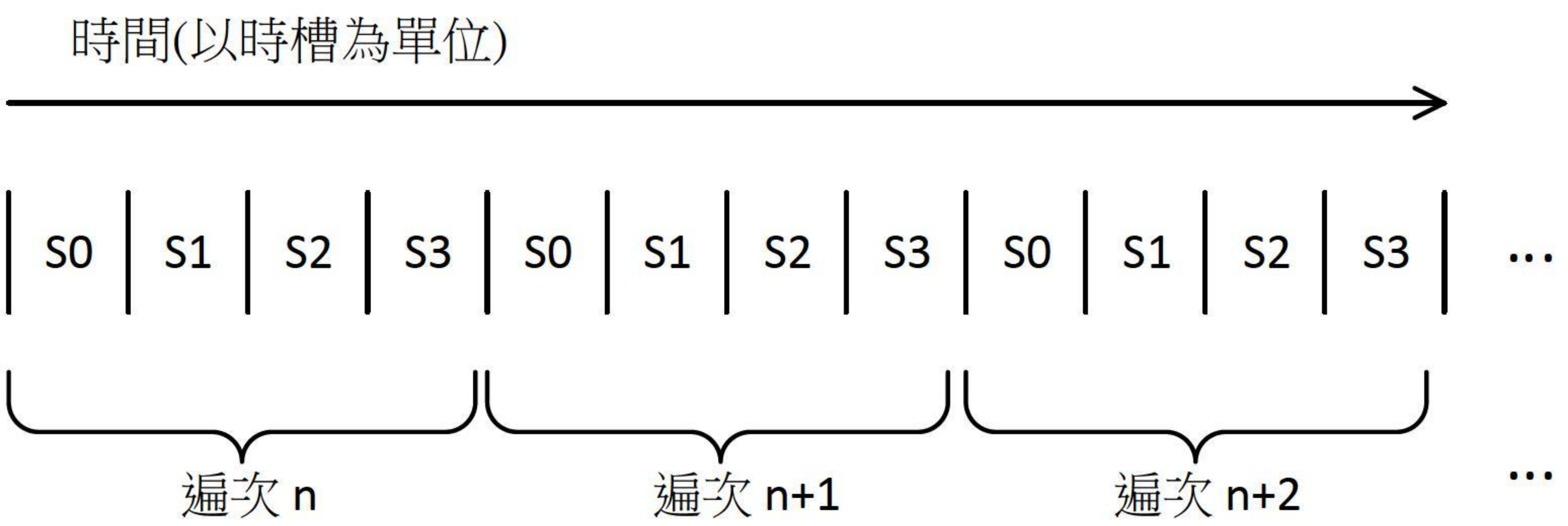
【發明圖式】



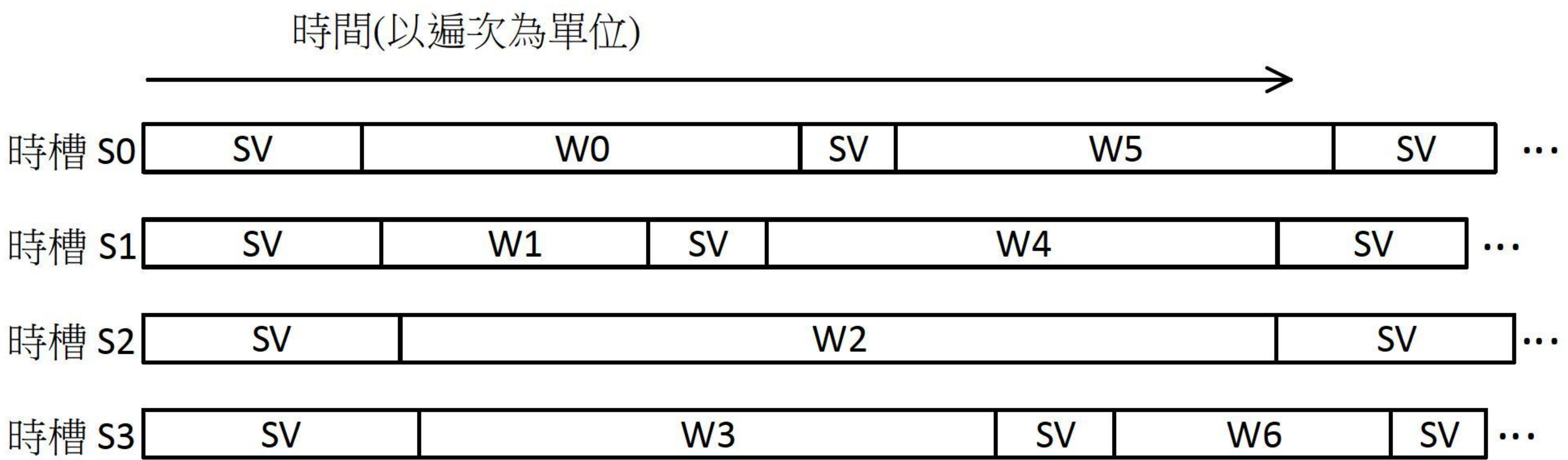
【圖1】



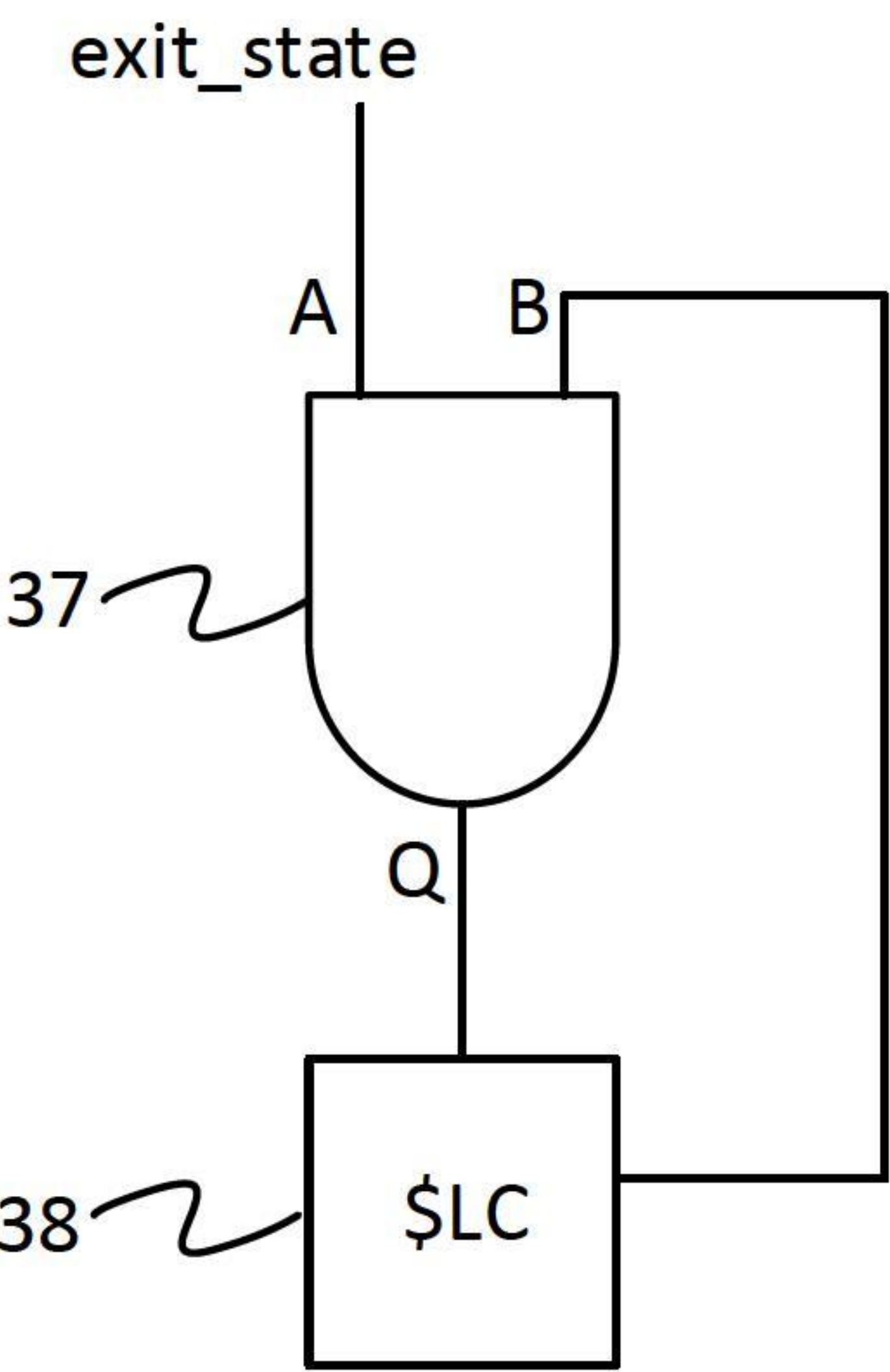
【圖2】



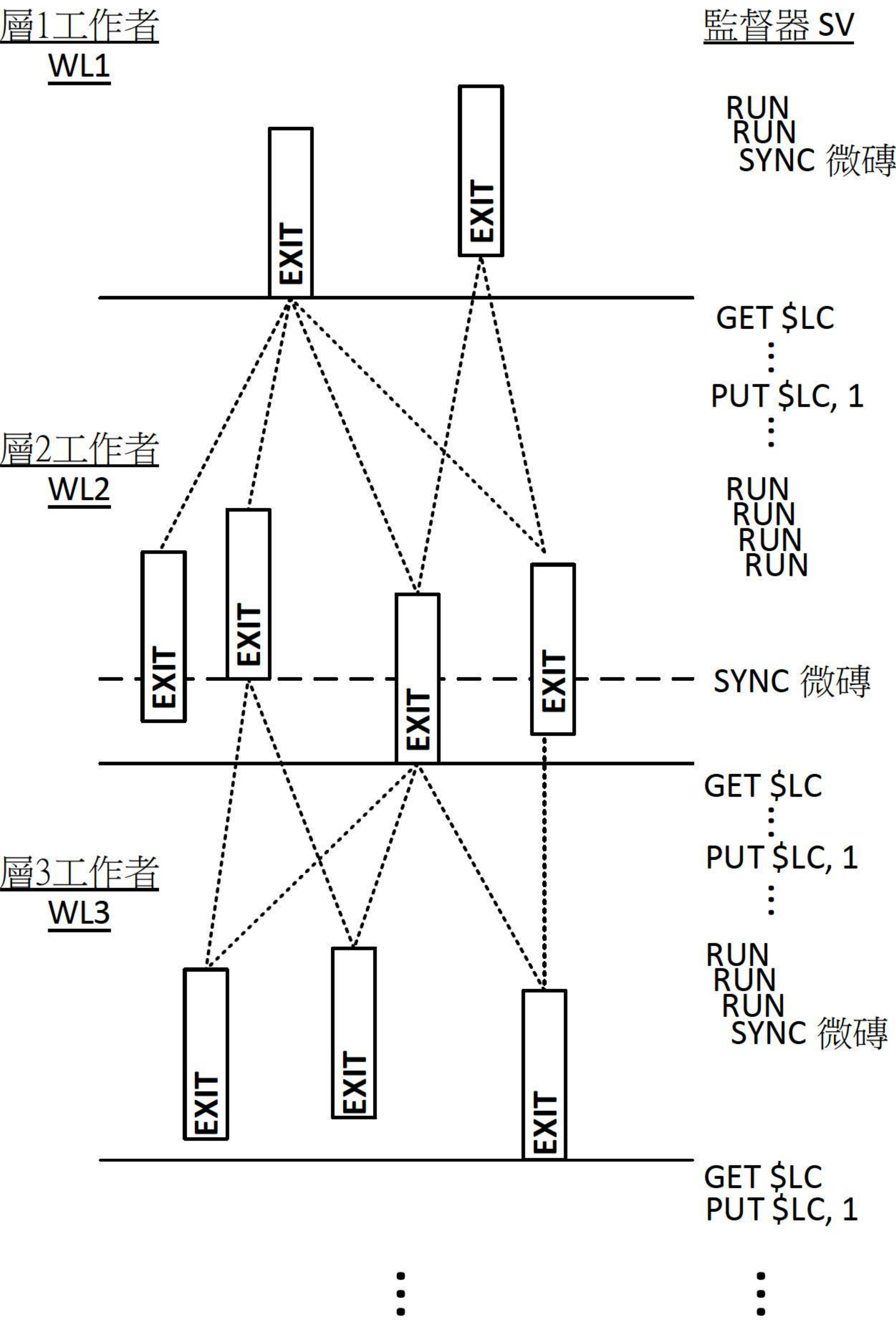
【圖3】



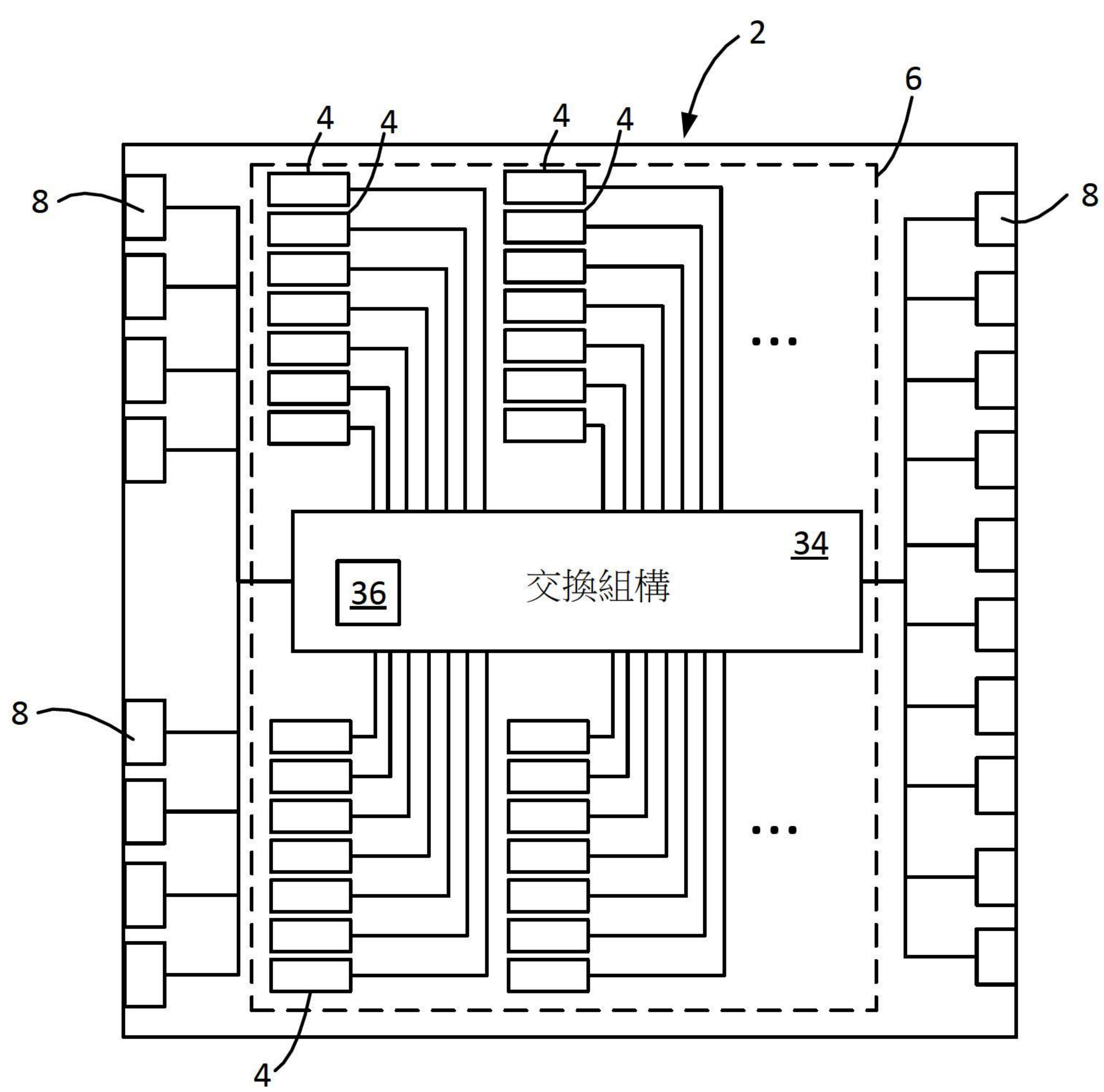
【圖4】



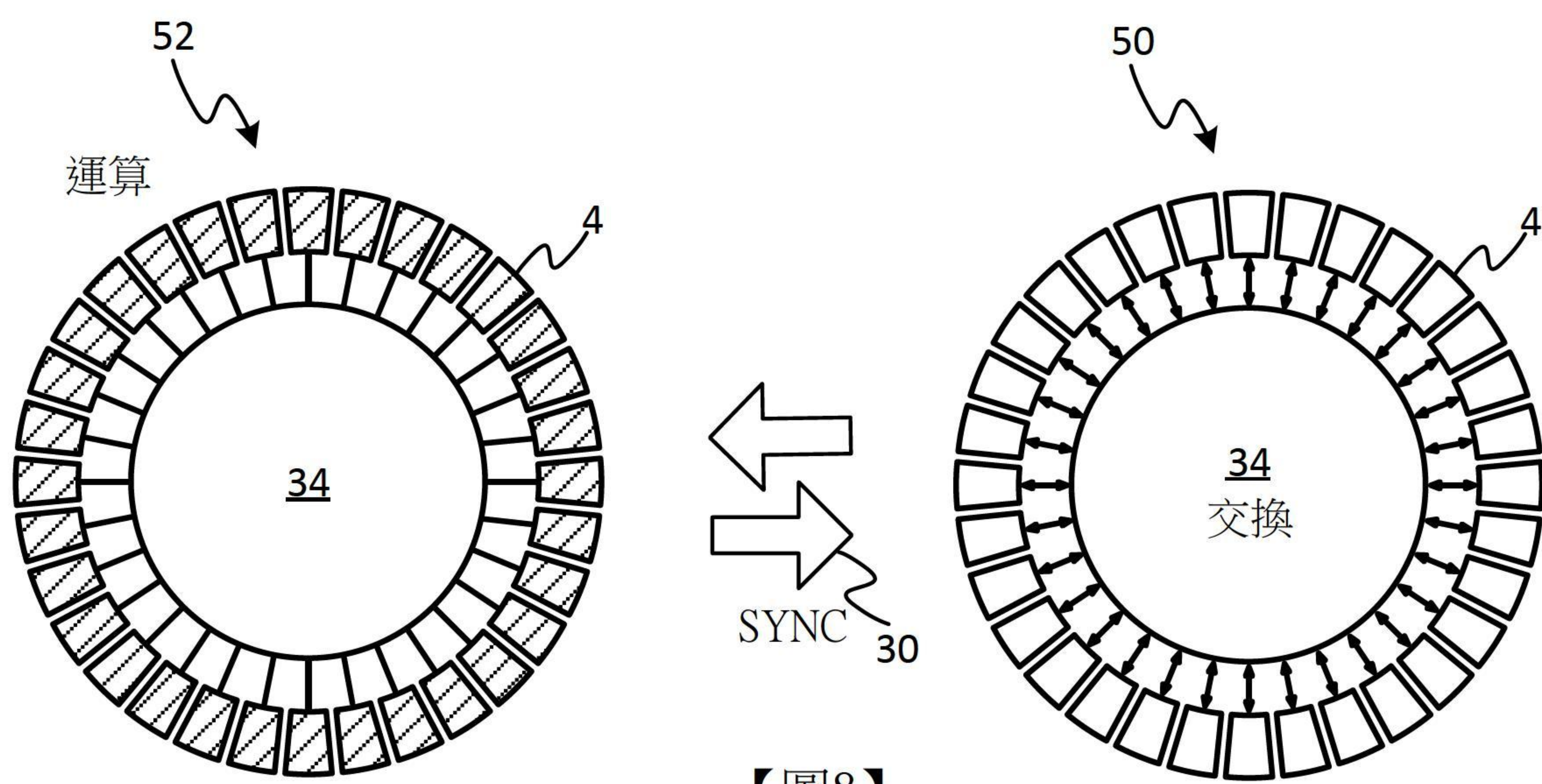
【圖5】



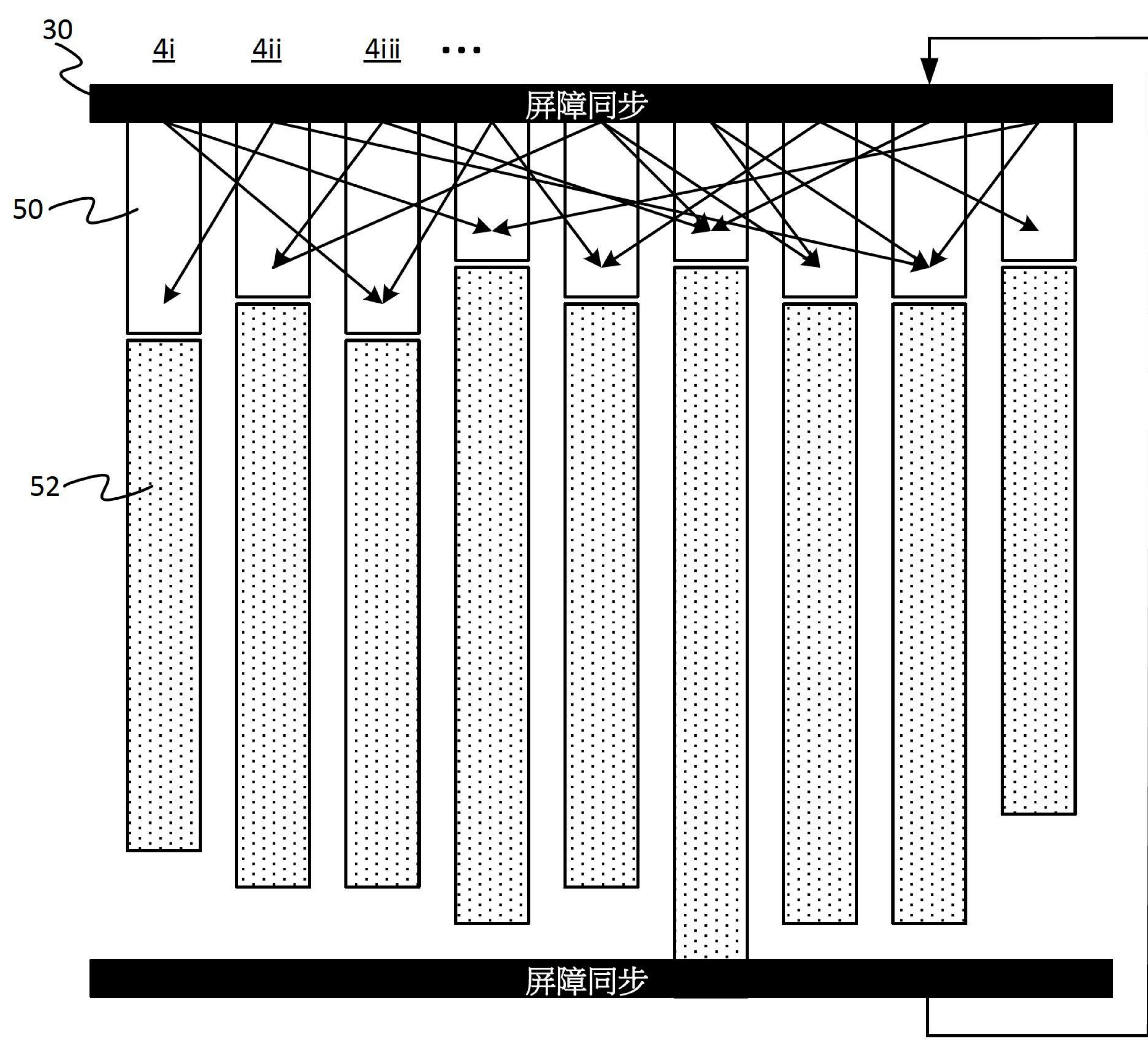
【圖6】



【圖7】

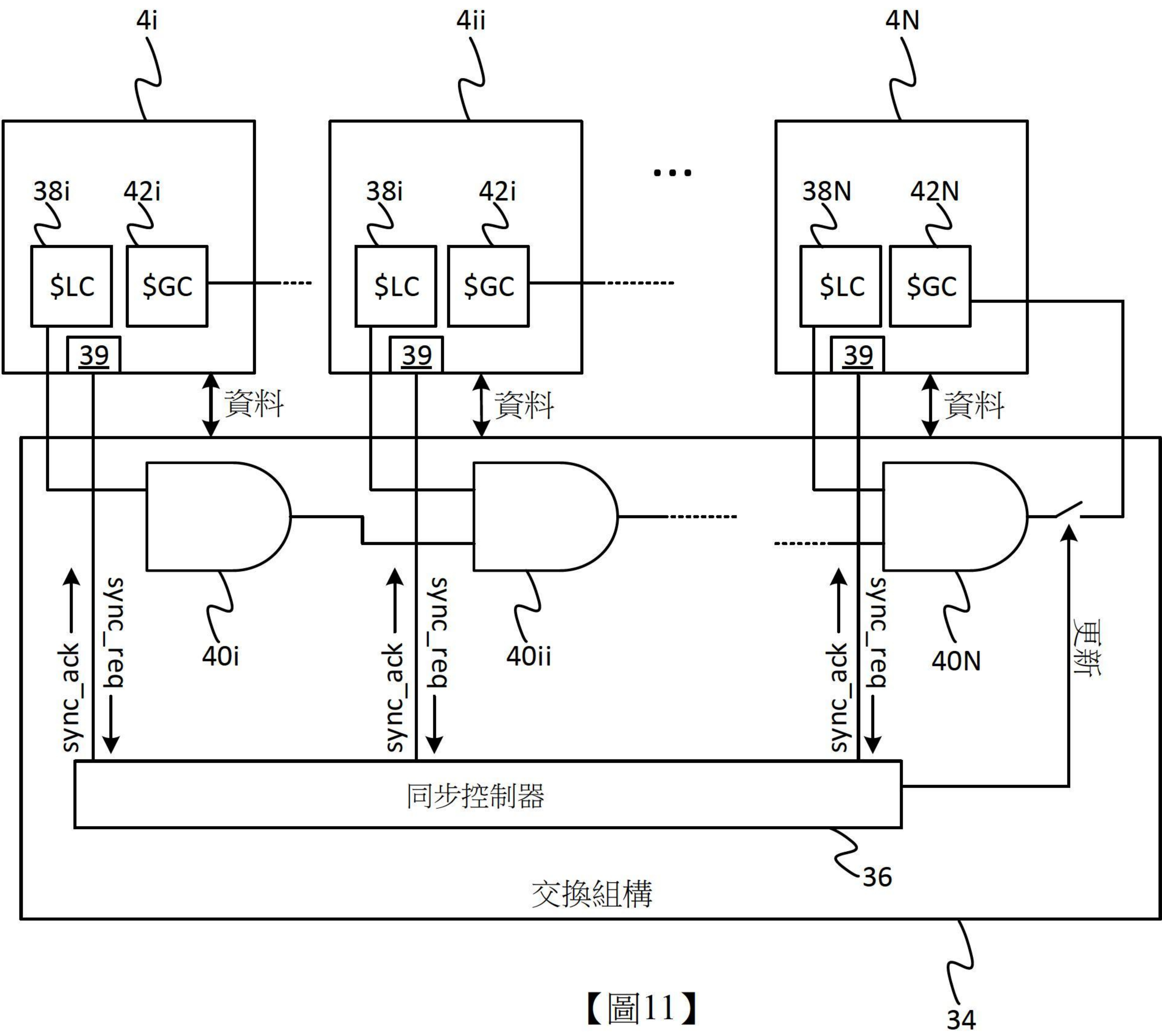


【圖8】

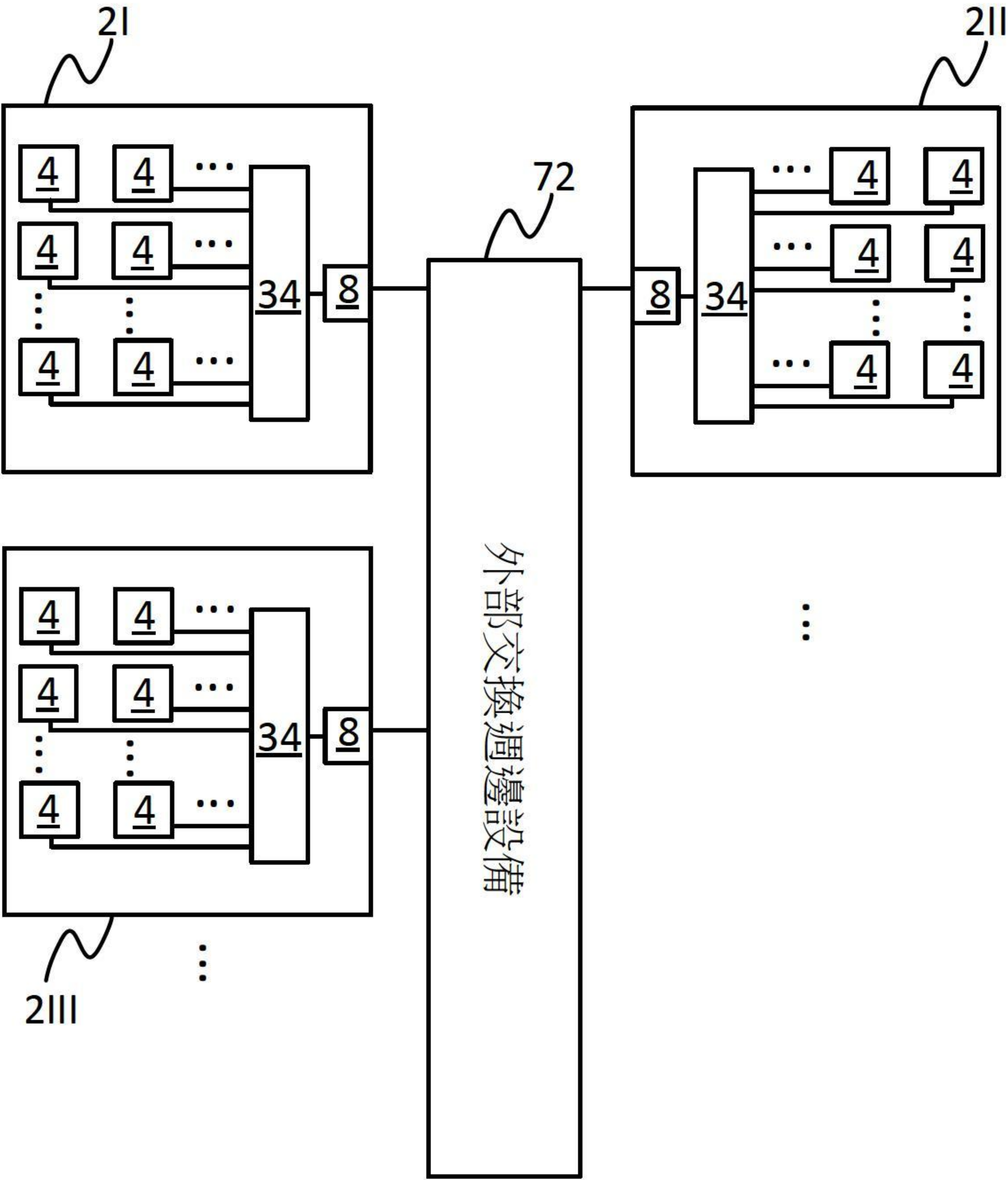


【圖9】

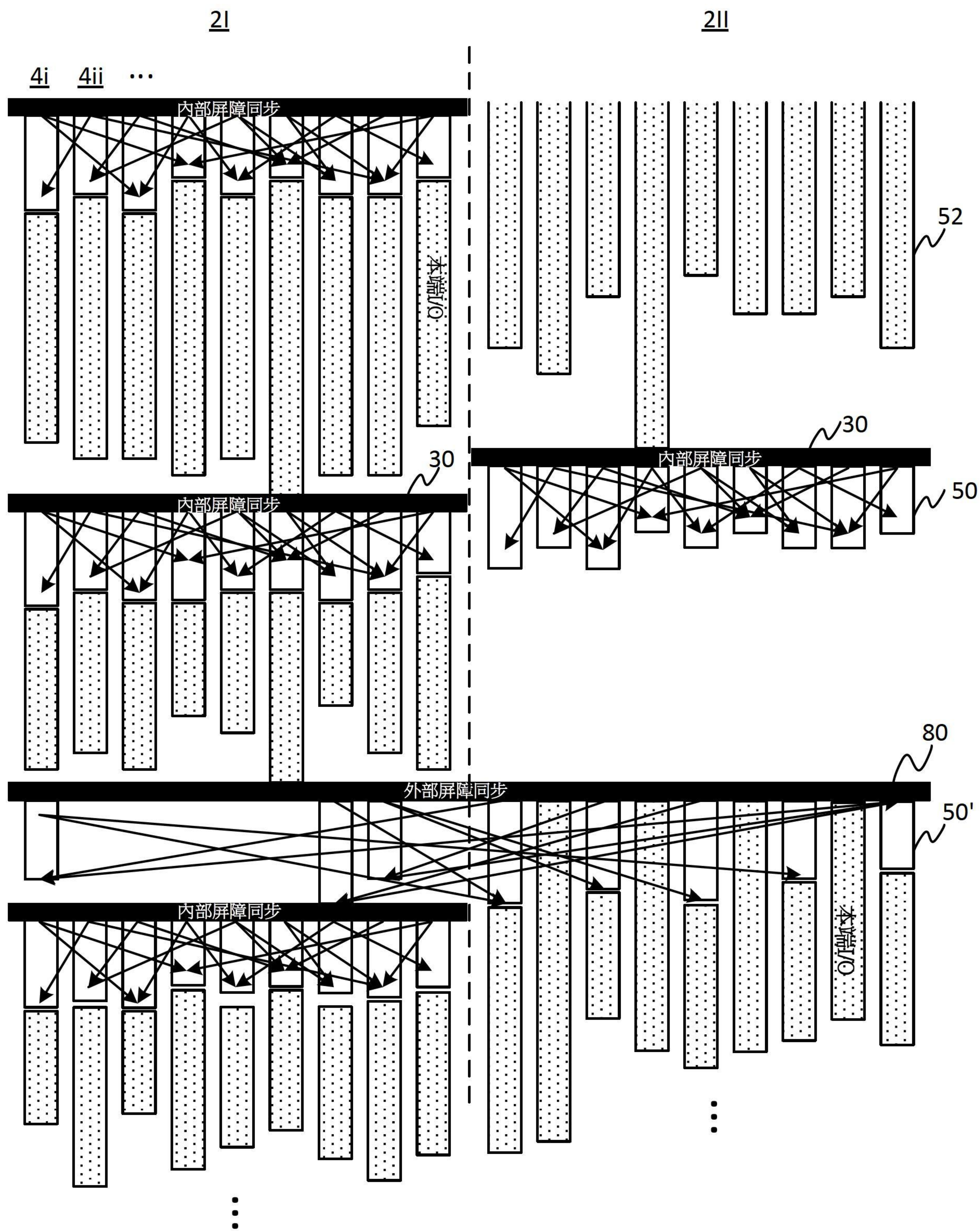




【圖11】

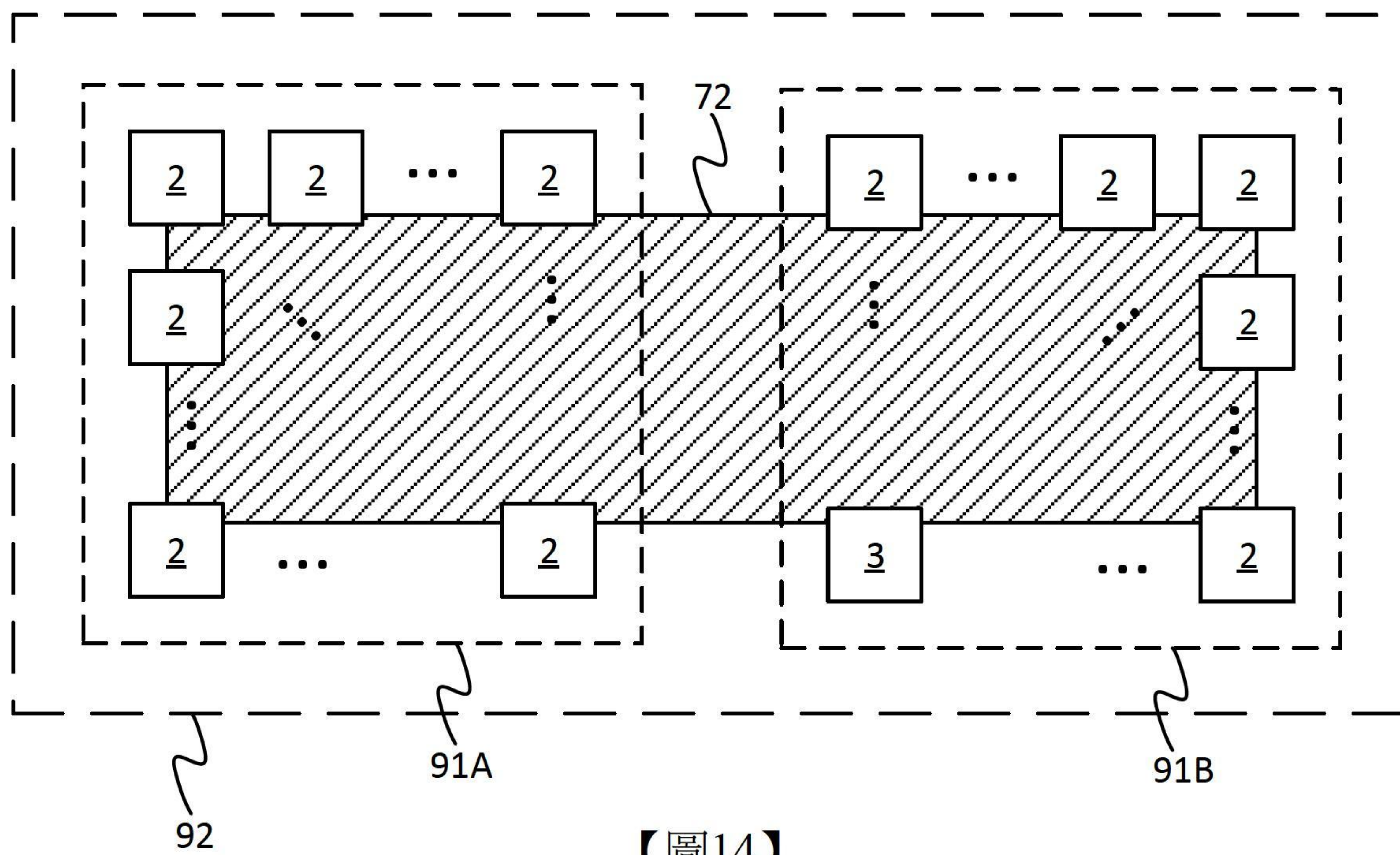


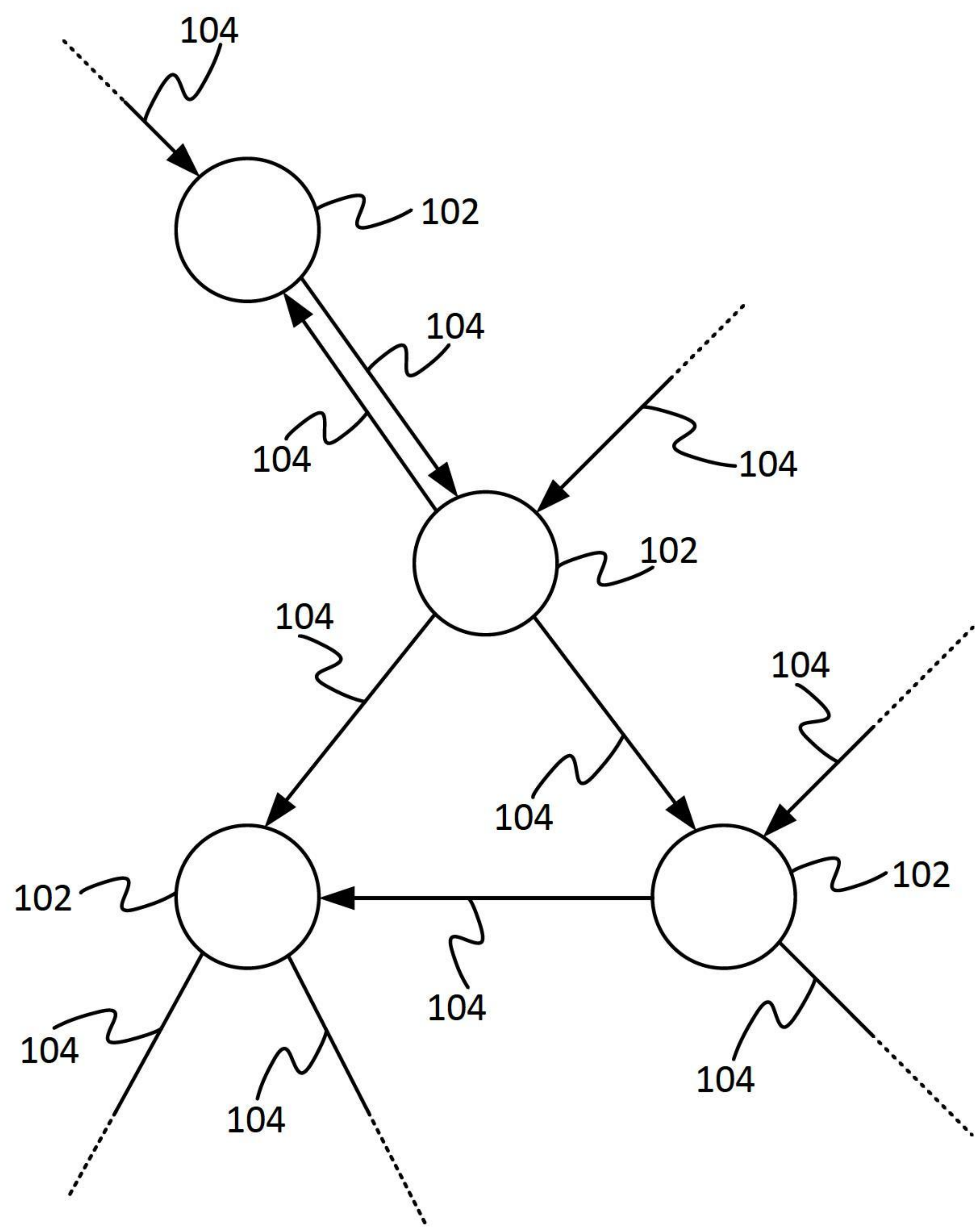
【圖12】



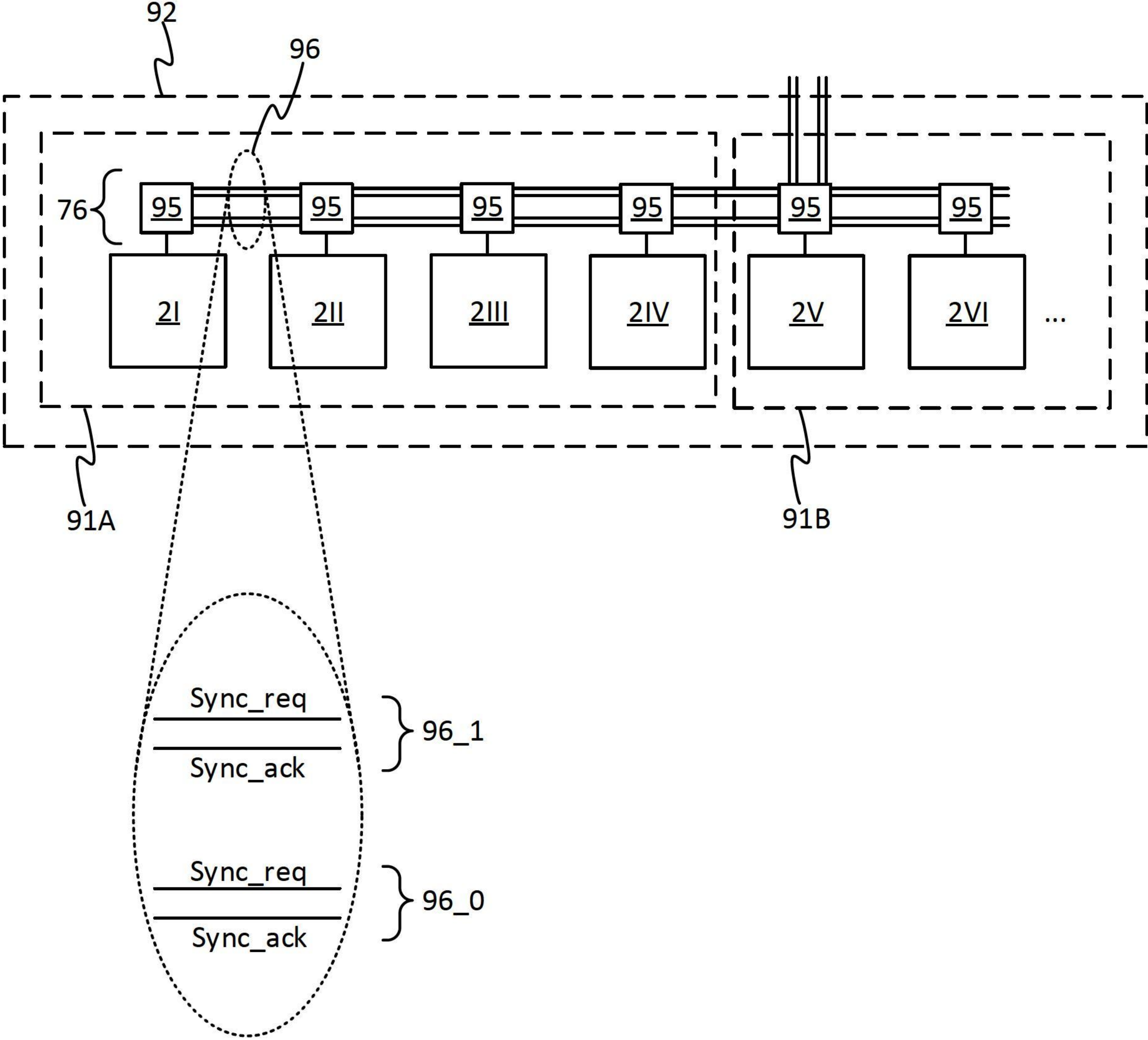
【圖13】

【圖14】





【圖15】



【圖16】