



(21)申請案號：106140091 (22)申請日：中華民國 106 (2017) 年 11 月 20 日

(51)Int. Cl. : **H03K5/135 (2006.01)**

(30)優先權：2016/11/23 美國 62/425,792
2017/11/17 美國 15/816,235

(71)申請人：美商德吉姆公司 (美國) DEGIRUM CORPORATION (US)
美國

(72)發明人：譚 提特 S. TAM, KIT S. (US)

(74)代理人：蔡清福；蔡馭理

(56)參考文獻：

US 4716575	US 6523155B1
US 7280550B1	US 2009/0100189A1
US 2010/0208742A1	US 2012/0072759A1
US 2012/0287344A1	US 2013/0279525A1
US 2014/0079393A1	US 2014/0266369A1
US 2015/0178831A1	US 2016/0116935A1

審查人員：蕭明椿

申請專利範圍項數：16 項 圖式數：8 共 46 頁

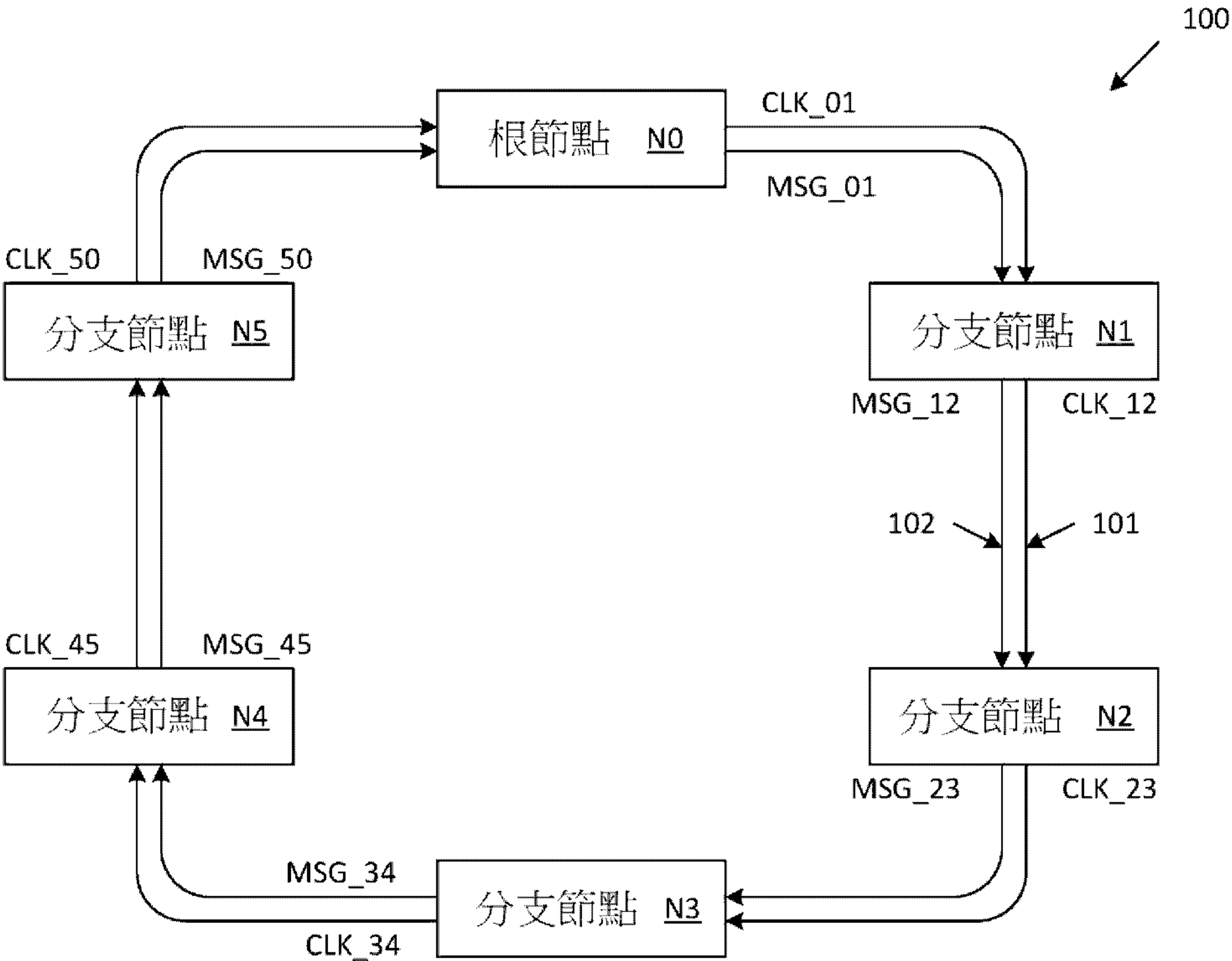
(54)名稱
分散控制同步環狀網路架構

(57)摘要

環狀網路架構包括以環狀配置的多個通訊節點。波狀管線是用以提供高頻寬以及低潛時晶載通訊。每個節點實施源同步計時方案，以至於不需要建立橫跨大晶粒面積的大規模低偏斜時鐘樹。單一參考時鐘訊號是在根節點內產生，並以單向方式經由環狀網路的每個節點路由。每個節點包括時間戳記計數器以及色彩位元暫存器，其儲存能夠使節點以精確的順序解析由其他節點發佈的排序交易訊息的值，即使節點被獨立地操作，並以完全不同的時機順序接收各種交易訊息。因為控制邏輯分散在節點之中，不需要中央控制器。

A ring network architecture includes multiple communication nodes configured in a ring. Wave pipelining is used to provide for high bandwidth and low latency on-chip communications. Each node implements a source-synchronized clocking scheme, such that there is no need to build an extensive low skew clock-tree across a large die area. A single reference clock signal is generated within a root node, and is routed through each of the nodes of the ring network in a unidirectional manner. Each node includes a timestamp counter and a color bit register, which store values that enable the node to resolve ordered transaction messages issued by the other nodes in a precise order, even though the nodes are operating independently, and receive the various transaction messages in totally different timing orders. Because the control logic is distributed among the nodes, no centralized controller is necessary.

指定代表圖：



第 1 圖



I730197

【發明摘要】

【中文發明名稱】 分散控制同步環狀網路架構

【英文發明名稱】 Distributed Control Synchronized Ring Network Architecture

【中文】

環狀網路架構包括以環狀配置的多個通訊節點。波狀管線是用以提供高頻寬以及低潛時晶載通訊。每個節點實施源同步計時方案，以至於不需要建立橫跨大晶粒面積的大規模低偏斜時鐘樹。單一參考時鐘訊號是在根節點內產生，並以單向方式經由環狀網路的每個節點路由。每個節點包括時間戳記計數器以及色彩位元暫存器，其儲存能夠使節點以精確的順序解析由其他節點發佈的排序交易訊息的值，即使節點被獨立地操作，並以完全不同的時機順序接收各種交易訊息。因為控制邏輯分散在節點之中，不需要中央控制器。

【英文】

A ring network architecture includes multiple communication nodes configured in a ring. Wave pipelining is used to provide for high bandwidth and low latency on-chip communications. Each node implements a source-synchronized clocking scheme, such that there is no need to build an extensive low skew clock-tree across a large die area. A single reference clock signal is generated within a root node, and is routed through each of the nodes of the ring network in a unidirectional manner. Each node includes a timestamp counter and a color bit register, which store values that enable the node to resolve ordered transaction messages issued by the other nodes in a precise order, even though the nodes are operating independently, and receive the

various transaction messages in totally different timing orders. Because the control logic is distributed among the nodes, no centralized controller is necessary.

【指定代表圖】 第1圖

【代表圖之符號簡單說明】

100：分散控制同步環狀網路

101：時鐘路徑

102：訊息匯流排

CLK_01、CLK_12、CLK_23、CLK_34、CLK_45、CLK_50：時鐘訊號

MSG_01、MSG_12、MSG_23、MSG_34、MSG_45、MSG_50：輸出訊息

【發明說明書】

【中文發明名稱】 分散控制同步環狀網路架構

【英文發明名稱】 Distributed Control Synchronized Ring Network Architecture

【技術領域】

【0001】 相關申請案

【0002】 此申請案主張標題為「分散控制同步環狀網路架構」於2017年11月17日申請的美國專利申請案第15/816,235號的優先權，其主張於2016年11月23日申請、標題為「分散控制同步環狀網路架構」的美國臨時專利申請案第62/425,792號的優先權，且併入於本文中以為參考。

【0003】 本發明與改進的環狀網路架構有關，用於達到高頻寬以及低潛時晶載(on-chip)通訊。

【先前技術】

【0004】 傳統的半導體晶片可包括大規模低偏斜時鐘樹，時鐘樹將全域時鐘訊號分配至於晶片上製造的多個電路區塊的每一個。時鐘樹被設計以確保多個電路區塊的每一個接收與每個其他電路區塊相同的輸入時鐘訊號。實施這樣的全域時鐘樹需要廣泛的佈局（晶粒）面積以及相當複雜的互連結構（以能夠有點對點通訊、廣播通訊、仲裁以及同步）。沿著互連結構包括了管線階段以維持最小的延遲。此外，需要中央控制器以建立並控制全域時鐘樹。由於這些結構，全域時鐘樹消耗了大量的功率。

【0005】 儘管有上述的控制，時鐘偏斜最後可降低全域時鐘樹的性能。在所有通訊媒介之間的實體距離的變異數使得消除此時鐘偏斜很困難。

【0006】因此會想要具有用於在半導體晶片上實施時鐘同步的改進方法以及結構。期望此改進的方法以及結構能夠在半導體晶片上提供高頻率、低潛時的操作，而不需要中央控制器。更期望此改進的方法以及結構具有簡單的實體實施，其不需要在半導體晶片的大區域上使用低偏斜、高功率消耗的時鐘樹。

【發明內容】

【0007】因此，本發明提供了一種環狀網路架構，其中多個通訊節點以環狀配置。環狀網路架構使用波狀管線以提供高頻寬以及低潛時晶載通訊。使用本發明的環狀網路架構可達到非常高的操作頻率。例如，假設每個通訊節點包括在環狀網路上傳遞訊息的節點邏輯，其中節點邏輯包括少於12級具有四的扇出（FO4）的邏輯，每個邏輯閘具有100皮秒（ps）的閘延遲。在此情況中，甚至考慮了節點之間的線路延遲，本發明的環狀網路架構能夠提供5 GHz或更高的數據傳輸頻率。

【0008】依照一個實施方式，環狀網路的每一個節點實施了源同步計時方案，以至於不需要在遍及大晶粒面積上建立大規模的低偏斜時鐘樹。所需的控制邏輯分散在節點之間，以至於不需要中央控制器。由本發明環狀網路架構實施的計時以及控制方案顯著地簡化了網路的實體實施，同時允許非常高速的操作。

【0009】依照一個實施方式，在環狀網路的根節點內產生參考時鐘訊號。此參考時鐘訊號以單向方式經由環狀網路的每一個節點來按路徑發送。以參考時鐘訊號在節點之間傳輸訊息，其中使用相對應的接收參考時鐘訊號將由節點接收的每個訊息計時至節點中。環狀網路的每個節點包括時間戳記計數器以及色彩位元暫存器。每次在節點產生並傳輸原始排序訊息，以及每次在節點接收其非原本產生的排序訊息時，使用所接收的參考時鐘訊號來更新每個節點的時

間戳記計數器。脈衝訊息不斷地循環通過環狀網路的節點，其中每次由節點接收脈衝訊息時，脈衝訊息雙態觸變節點的色彩位元暫存器。在正常操作前初始化每個時間戳記計數器以及色彩位元暫存器，以至於每個時間戳記計數器儲存相同的時間戳記值，且每個色彩位元暫存器儲存相同的色彩位元值。即使節點被獨立地操作，時間戳記值以及色彩位元值使每個節點能夠以精確的順序解析由其他節點發佈的交易訊息，並以完全不同的計時順序來接收各種交易訊息。此能力提供用於仲裁以及同步的強大器具。

【0010】在一個實施方式中，每個節點將來自其時間戳記計數器的時間戳記值附加至所產生的每個排序訊息。每個節點更包括儲存在環狀網路上接收的排序訊息的訊息竊取程式。當在節點內雙態觸變色彩位元時，節點分析儲存在其訊息竊取程式中的排序訊息，並將優先權授予至具有最久時間戳記值的排序訊息。

【0011】本發明環狀網路架構可用以，例如，在巨大的並聯處理器上實施 MOSEI（修改、自有、共享、排除、無效）快取一致性協定記憶體匯流排或通訊網路。

【0012】鑑於下述描述以及圖式，將更完全了解本發明。

【圖式簡單說明】

【0013】

第1圖是依照本發明的一個實施方式的分散控制同步環狀網路的方塊圖。

第2圖是依照本發明的一個實施方式第1圖的分散控制同步環狀網路的根節點的方塊圖。

第3圖是依照本發明的一個實施方式第1圖的分散控制同步環狀網路的分支節點的方塊圖。

第4圖是依照本發明的一個實施方式示出第2圖的根節點部分以及第3圖的分支節點、用於以時鐘訊號來源同步傳輸訊息的方塊圖。

第5圖是依照本發明的一個實施方式示出第4圖的各種訊號的源同步計時的波形圖。

第6圖是依照本發明的一個實施方式示出第1圖的初始化環狀網路的方法的流程圖。

第7圖是依照本發明的一個實施方式示出實施第6圖的節點初始化過程方法的方法的流程圖。

第8A圖、第8B圖、第8C圖、第8D圖、第8E圖、第8F圖、第8G圖、第8H圖以及第8I圖是依照本發明的一個實施方式示出在點對點排序訊息的傳輸期間的環狀網路狀態。

【實施方式】

【0014】第1圖是依照本發明的一個實施方式的分散控制同步環狀網路100的方塊圖。環狀網路100包括根節點N0以及分支節點N1-N5，其以環狀連接。雖然在所描述的範例中包括了六個通訊節點N0-N5，要了解的是，在其他實施方式中可包括其他數目的節點。然而，每個環狀網路將包括一個根節點以及一個或更多個分支節點。每個節點N0-N5接收相對應的時鐘輸入訊號以及相對應的訊息輸入訊號，並傳輸相對應的時鐘輸出訊號以及相對應的訊息輸出訊號。環狀網路100的訊號是單方向的。在所示出的實施方式中，環狀網路100的訊號以順時鐘的方式路由（雖然這不是必要的）。每個節點N0-N5具有傳輸相鄰者以及接收相鄰者。例如，分支節點N3包括傳輸相鄰者N2以及接收相鄰者N4。節點N0、N1、N2、N3、N4以及N5將時鐘訊號CLK_01、CLK_12、CLK_23、CLK_34、CLK_45以及CLK_50以及相對應的輸出訊息MSG_01、MSG_12、MSG_23、

MSG_34、MSG_45以及MSG_50分別傳輸至接收相鄰者N1、N2、N3、N4、N5以及N0。時鐘訊號CLK_01、CLK_12、CLK_23、CLK_34、CLK_45以及CLK_50在環狀網路100的時鐘路徑101上傳輸，且訊息MSG_01、MSG_12、MSG_23、MSG_34、MSG_45以及MSG_50在環狀網路100的訊息匯流排102上傳輸。

【0015】第2圖是依照本發明的一個實施方式更詳細地示出根節點N0的方塊圖。根節點N0包括主時鐘產生器201、延遲線202、再同步器205、節點邏輯210、同步器220、訊息竊取程式230、接收先進先出（FIFO）記憶體231、傳輸FIFO記憶體232、客戶邏輯240、節點識別符（ID）記憶體250、色彩位元暫存器251、時間戳記（TS）計數器255、訊息組譯器260以及多工器265。

【0016】一般而言，根節點N0包括在兩個時鐘域中操作的電路系統，兩個時鐘域包括環狀網路時鐘域（由參考時鐘CK_REF指出）以及客戶邏輯時鐘域（由本地時鐘訊號CK0指出）。這些時鐘域之間的分界由第2圖中的虛線270示出。因此，客戶邏輯240以及客戶邏輯240與色彩位元暫存器251之間的轉移、節點ID暫存器250、時間戳記計數器255、訊息竊取程式230、接收FIFO 231以及傳輸FIFO 232是在客戶邏輯時鐘域中執行。時鐘產生器201、延遲線202、再同步器205、節點邏輯210、節點識別符（ID）記憶體250、色彩位元暫存器251、時間戳記（TS）計數器255、訊息組譯器260以及多工器265在環狀網路時鐘域中操作。此外，至訊息竊取程式230以及接收FIFO 231的寫入操作以及從傳輸FIFO 232的讀取操作是在環狀網路時鐘域中執行。依照一個實施方式，同步電路220（其接收參考時鐘訊號CK_REF以及本地時鐘訊號CK0）使用本技術領域已知的再同步方法來執行環狀網路時鐘域以及客戶邏輯時鐘域之間的同步。下面更詳細地描述了環狀網路時鐘域以及客戶邏輯時鐘域的需求。

【0017】根節點N0的主時鐘產生器201產生環狀網路100的參考時鐘訊號CK_REF。傳統的時鐘產生電路系統（例如，鎖相迴路電路）可用以產生參考時

鐘訊號CK_REF。在所描述的實施方式中，參考時鐘訊號CK_REF具有大約5 GHz或更高的頻率。然而，要了解的是，在其他實施方式中，參考時鐘訊號CK_REF可具有其他的頻率。參考時鐘訊號CK_REF的頻率以及電壓可基於環狀網路架構的頻寬需求以及功率最佳化而按比例縮放。注意，可以低於正常操作的參考時鐘頻率來執行設定過程（其描述於下文），以至於節點N0-N5之間的線延遲對於設定過程沒有影響。注意，根節點N0是包括主時鐘產生器201的唯一節點。如下面更詳細所述，每一個分支節點N1、N2、N3、N4以及N5對由它們的傳輸相鄰者N0、N1、N2、N3以及N4所發送的相對應時鐘訊號CLK_01、CLK_12、CLK_23、CLK_34以及CLK_45做出反應而操作。也就是，參考時鐘訊號CK_REF是經由串聯的分支節點N0-N5來有效地傳輸。

【0018】根節點N0使用參考時鐘訊號CK_REF來同步進來的訊息MSG_50（與進來的時鐘訊號CLK_50一起由傳輸相鄰者N5發送）。更具體而言，進來的訊息MSG_50一開始被同步至傳輸相鄰者N5的網路時鐘域。因此，在根節點N0內橫跨時鐘域的同步是必要的（即，從傳輸相鄰者N5接收的訊息MSG_50必須在根節點N0內與參考時鐘訊號CK_REF再同步）。在一個實施方式中，回應於進來的時鐘訊號CLK_50，再同步電路205藉由將進來的訊息MSG_50鎖存至第一正反器中來執行此再同步操作。回應於參考時鐘訊號CK_REF，在第一正反器的輸出所提供的訊息MSG_50然後被鎖存至第二正反器。第二正反器提供被提供至節點邏輯210的同步訊息MSG_0。回應於參考時鐘訊號CK_REF，將此同步訊息MSG_0儲存在節點邏輯210中。

【0019】第3圖是更詳細示出分支節點N1的方塊圖。在所描述的實施方式中，分支節點N2-N5實質上與分支節點N1相同（雖然這些節點可實施不同的客戶邏輯）。分支節點N1包括時鐘延遲線302、節點邏輯310、同步電路320、訊息竊取程式330、接收FIFO記憶體331、傳輸FIFO記憶體332、客戶邏輯340、節點

識別符（ID）記憶體350、色彩位元暫存器351、時間戳記（TS）計數器355、訊息組譯器360以及多工器365。注意，上面所列出的分支節點N1的元件類似於根節點N0的相似命名的元件。節點N1的客戶邏輯340對本地節點時鐘域訊號CK1做出反應而操作。虛線370標出了本地時鐘域（CK1）以及該環狀網路時鐘域（CK_REF）的邊界，其中同步電路320提供跨越這些時鐘域的通訊。注意，分支節點N1不包括與根節點N0的主時鐘產生器201或再同步器205均等的電路。也就是，根節點N0包括環狀網路100的唯一主時鐘產生器201。

【0020】現在將描述從根節點N0至分支節點N1的訊息以及時鐘訊號的源同步傳輸的時機。注意，在其他節點之間的傳輸是以與從節點N0傳輸至節點N1相同的方式執行。

【0021】第4圖是示出節點N0以及N1用於從節點N0傳輸訊息以及時鐘訊號至節點N1的相關部分的方塊圖，包括節點邏輯210、延遲線202、訊息組譯器260、多工器265、環狀網路時鐘訊號路徑101、環狀網路訊息匯流排102以及節點邏輯310。第5圖是依照本發明的一個實施方式示出第4圖的各種訊號的源同步計時的波形圖。

【0022】一般而言，節點N0可在兩個路徑上傳輸輸出訊息。在第一路徑中，由節點邏輯210接收的訊息MSG_0被轉送至節點邏輯N1。在此例子中，節點邏輯210決定所接收的訊息MSG_0應被轉送（例如，決定所接收的訊息MSG_0不定位址至節點N0），並將訊息轉送至多工器265。多工器265被控制以將此訊息路由為輸出訊息MSG_01（其在第4圖以及第5圖中被標示為MSG_01_{IN}）。節點邏輯210以及多工器265引進了關於參考時鐘訊號CK_REF的節點延遲D_{N0}。

【0023】在第二路徑中，由客戶邏輯240提供的訊息MSG_0_{CL}是經由傳輸FIFO 232路由並提供至訊息組譯器260（回應於參考時鐘訊號CK_REF）。訊息組譯器260將所需的資訊（例如，來自時間戳記計數器250的時間戳記值以及來

自節點ID記憶體350的節點ID值)提供至訊息MSG_0_{CL}，並將所產生的訊息提供至多工器265。多工器265被控制以將此訊息路由為輸出訊息MSG_01(其在第4圖以及第5圖中被標示為MSG_01_{IN})。訊息組譯器260以及多工器265將關於參考時鐘訊號CK_REF的節點延遲D_{N0}引進至訊息MSG_0_{CL}。在一個實施方式中，設計訊息組譯器260以及節點邏輯210，以至於這些電路將關於參考時鐘訊號CK_REF的相同(或類似的)延遲分別引進至所接收的訊息MSG_0_{CL}以及MSG_0。第5圖示出了存在於參考時鐘訊號以及傳輸訊息MSG_01_{IN}之間且用於訊息MSG_0或訊息MSG_0_{CL}的節點延遲D_{N0}。

【0024】雖然在第4圖以及第5圖的範例中大體上描述了訊息MSG_0以及MSG_0_{CL}，下面更詳細地描述了這些訊息的細節(以及有關的訊息「類型」，包括廣播訊息、讀取-修飾-寫入訊息、點對點未排序訊息以及點對點排序訊息)。

【0025】因為相對於參考時鐘訊號CK_REF延遲了傳輸訊息MSG_01_{IN}，延遲線202用以將延遲D₂₀₂加至參考時鐘訊號CK_REF，由此產生根節點N0的輸出時鐘訊號CLK_01(在第4圖以及第5圖中示為時鐘訊號CLK_01_{IN})。由延遲線202引進的延遲D₂₀₂經選擇以確保時鐘訊號CLK_01_{IN}關於從節點N0傳輸的訊息MSG_01_{IN}具有適當的設定以及保持時間。在一個實施方式中，使延遲線202的延遲D₂₀₂與上述的節點延遲D_{N0}相等。也就是，使用已知的節點N0的內部延遲將傳輸訊息MSG_01_{IN}同步至輸出時鐘訊號CLK_01_{IN}。依照一個實施方式，延遲D₂₀₂是可調整的(例如，可在節點邏輯210及/或客戶邏輯240的控制下依需求延長或縮短)。

【0026】注意，在節點N0以及N1之間用以實施時鐘訊號路徑101以及訊息匯流排102的結構101D以及102D可展現顯著的延遲(例如，緩衝延遲)。由時鐘訊號線結構101D以及訊息匯流排結構102D引進的延遲被分別標示為延遲D₁₀₁以及D₁₀₂。因此，由節點N1接收的時鐘訊號CLK_01_{OUT}相對於由節點N0傳輸的

時鐘訊號CLK_01_{IN}被延遲了D₁₀₁。同樣地，由節點N1接收的訊息MSG_01_{OUT}相對於由節點N0傳輸的訊息MSG_01_{IN}被延遲了D₁₀₂。此關係示於第5圖中。為了維持想要的設定以及由節點N0建立的保持時間（當產生CLK_01_{IN}以及MSG_01_{IN}時），延遲D₁₀₁以及D₁₀₂必須相同。因此非常重要，用以實施時鐘訊號線結構101D以及訊息匯流排結構102D的線路以及緩衝是高度相等且平衡的，以最小化設定以及保持時間損失。也就是，時鐘路徑結構101D以及訊息匯流排結構102D應在節點N0以及N1之間提供實質上相同的傳輸特徵，使得所傳輸的時鐘訊號CLK_01_{IN}以及所傳輸的訊息MSG_01_{IN}的相關相(phase)被所接收的時鐘訊號CLK_01_{OUT}以及所接收的訊息MSG_01_{OUT}維持在接收節點N1。

【0027】依照上述範例，時鐘路徑101以及訊息匯流排102作為波狀管線系統來操作，其中在訊息匯流排102上傳輸的訊息以源同步方式使用在時鐘路徑101上傳輸的時鐘訊號被鎖存至接收節點中。以此方式，以參考時鐘訊號CK_REF的頻率將訊息從節點N0傳輸至節點N1，允許了從節點N0至節點N1的快速數據傳輸。在正常操作期間，在每個節點N0-N5內的節點邏輯回應於所接收的時鐘訊號而鎖存了進來的訊息（例如，節點邏輯210以及310分別回應於所接收的時鐘訊號CK_REF以及CLK_01，分別鎖存了進來的訊息MSG_0以及MSG_01）。

【0028】因為實施了波狀管線，實施了線，時鐘訊號線結構101D以及訊息匯流排結構102D的線路以及緩衝延遲D₁₀₁以及D₁₀₂將不會降低環狀網路100的操作頻率。注意，波狀管線可允許傳輸節點將第一訊息發送至接收節點，然後在接收節點已接收第一訊息之前將第二訊息發送至接收節點。也就是，可有多個訊息在訊息匯流排結構102D上同時從節點N0傳播至節點N1。注意，如果參考時鐘訊號CK_REF的週期（例如，對於5 GHz時鐘為200 ps）少於相鄰節點N0以及N1之間的線路/緩衝延遲D₁₀₁/D₁₀₂，這可能發生。依照一個實施方式，在每對相鄰節點之間的線路/緩衝延遲D₁₀₁/D₁₀₂必須足夠長（即，展現最小的延遲），以

確保從第一節點（例如，節點N0）傳輸至相鄰第二節點（例如，節點N1）的第一多個訊息不趕上從第二節點（例如，節點N1）傳輸至相鄰第三節點（例如，節點N2）的第二多個訊息。

【0029】在一個實施方式中，節點邏輯210以及多工器265的閘延遲可決定參考時鐘訊號CK_REF的最大頻率。為了最大效率，應使所產生的參考時鐘訊號CK_REF的週期相等於節點延遲 D_{N0} 。在一個實施方式中，節點邏輯210包括連續的邏輯（例如，10級的扇出4（FO4）邏輯），其允許參考時鐘訊號CK_REF具有5 GHz或更高的頻率。

【0030】因為節點邏輯210具有相對簡單的設計，可在相對高的頻率下執行環狀網路的節點N0-N5之間的通訊。客戶邏輯240典型地包括較複雜的設計，且可在比參考時鐘訊號CK_REF的頻率慢的頻率下操作。也就是，雖然客戶邏輯240可在參考時鐘訊號CK_REF的頻率下接收來自環狀網路的訊息，客戶邏輯240不需要以參考時鐘訊號CK_REF的頻率回應。

【0031】在一個替代的實施方式中，也可對具有相對低頻率的參考時鐘訊號CK_REF做出反應來操作環狀網路100，其中參考時鐘訊號CK_REF的週期大於鄰接節點之間的線路/緩衝延遲 D_{101}/D_{102} 。在此例子中，在傳輸節點發送下一個訊息之前，由節點傳輸的每個訊息由鄰接節點接收（以及鎖存）。在此實施方式中，在鄰接節點之間的線路/緩衝延遲 D_{101}/D_{102} 對於訊號管線不具有影響。在一個實施方式中，參考時鐘訊號CK_REF在設定階段期間（下述）被降低至這樣低的頻率，但在正常操作期間被增加至高頻率（例如，5GHz或更高）。

【0032】現在將描述環狀網路100的節點N0-N5的設定。

【0033】第6圖是依照本發明的一個實施方式示出初始化環狀網路100的方法的流程圖600。如第6圖的步驟601所示，開始了設定過程（例如，藉由啟動提供至根節點N0的客戶邏輯240的重設訊號）。作為回應，客戶邏輯240開始節點

初始化過程（步驟602），其中儲存在節點N0-N5中的節點ID暫存器的節點ID值被初始化，儲存在節點N0-N5中的色彩位元暫存器的色彩位元被重設至預定值（例如，「0」），且由節點N0-N5的每一者內的時間戳記計數器所儲存的時間戳記值被重設至預定值（例如，「0」）。如第6圖的步驟603所示，開始了循環經過節點N0-N5的每一者的脈衝訊息（例如，藉由啟動提供至根節點N0的客戶邏輯240的脈衝訊號）。環狀網路100的正常操作然後開始（步驟604）。

【0034】第7圖是依照本發明的一個實施方式示出實施步驟602的節點初始化過程的方法的流程圖700。根節點N0是硬接線，以至於節點ID暫存器250儲存等於「0」的節點ID值。回應於接收到啟動的重設訊號，客戶邏輯240產生為讀取-修飾-寫入訊息的ID_設定訊息（步驟701）。此ID_設定訊息包括一開始設定為具有「0」的值的節點ID欄位（基於由節點N0的節點ID暫存器250所儲存的節點ID值）。客戶邏輯240也重設其時間戳記計數器255以儲存「0」的時間戳記值，並也重設其色彩位元暫存器251以儲存「0」的色彩位元值（步驟702）。客戶邏輯240將ID_設定訊息（經由傳輸FIFO 232、訊息組譯器260以及多工器265）傳輸至相鄰接收節點N1。此ID_設定訊息以與上述第4圖以及第5圖有關的方式與參考時鐘訊號CK_REF同步。

【0035】節點N1內的節點邏輯310（使用相對應的時鐘訊號CLK_01）接收ID_設定訊息。節點邏輯310（經由接收FIFO 331）將ID_設定訊息發送至客戶節點N1的邏輯340。作為回應，客戶邏輯340將ID_設定訊息的ID欄位中的ID值增加1（步驟703）。客戶邏輯340然後將更新的ID值（例如，「1」）寫入至其節點ID暫存器350（步驟704）。客戶邏輯340也重設其時間戳記計數器355以儲存「0」的時間戳記值，並也重設其色彩位元暫存器351以儲存「0」的色彩位元值（步驟705）。因此，分支節點N1被指定等於「1」的節點ID值、等於「0」的時間戳記值以及等於「0」的色彩位元值。

【0036】客戶邏輯340（經由傳輸FIFO 332、訊息組譯器360以及多工器365）將包括更新節點ID值的修飾ID_設定訊息傳輸至相鄰接收節點N2（步驟706）。

【0037】如果下游接收節點是根節點N0（步驟707，「是」分支），則完成ID_設定操作（步驟708）。如果下游接收節點不是根節點N0（步驟707，「否」分支），則處理回到步驟703，由此在接收分支節點中的客戶邏輯將ID_設定訊息中的節點ID值增加1（步驟703），將增加的節點ID值儲存在其節點ID暫存器中（步驟704），並重設其時間戳記計數器以及色彩位元（步驟705）。此過程繼續，以至於節點N0至N5被分別指定0至5的獨特節點ID值。在ID_設定操作的最後，節點N0-N5的時間戳記計數器全部儲存相同的時間戳記值（例如，「0」）以及相同的色彩位元值（例如，「0」）。

【0038】在完成節點初始化過程之後，在節點N0中的客戶邏輯240在環狀網路100上傳輸脈衝訊息。在一個實施方式中，脈衝訊息由提供至客戶邏輯240的啟動脈衝訊號開始。在一個替代的實施方式中，客戶邏輯240回應於接收到來自傳輸節點N5的修飾ID_設定訊息而起始脈衝訊息。脈衝訊息不斷地橫跨環狀網路100。將脈衝訊息相繼地提供至節點N0-N5內的節點邏輯。每次在節點內的節點邏輯接收脈衝訊息時，節點邏輯雙態觸變儲存在其相對應的色彩位元暫存器的色彩位元值。例如，節點N1中的節點邏輯310第一次接收來自節點N0的脈衝訊息，節點邏輯310將儲存在色彩位元暫存器351中的色彩位元值從「0」改變成「1」。如下面更詳細描述的，在由節點N0-N5接收的排序訊息的仲裁中使用色彩位元。

【0039】在完成上述設定過程之後，節點N0-N5準備好用於正常操作。現在將描述依照本發明的一個實施方式的環狀網路100的正常操作。

【0040】節點N0-N5的客戶邏輯（例如，客戶邏輯240）藉由經由訊息匯流排102交換訊息來通訊。訊息匯流排102的寬度以及格式必須滿足下述需求。訊

息匯流排102必須能夠封裝下面更詳細描述的所有訊息類型。訊息匯流排102也被設計以提供所需的頻寬以及時機，同時最小化所需的線路/緩衝的佈局面積。

【0041】在一個實施方式中，在訊息匯流排102上實施了四種訊息類型，包括：1) 廣播訊息、2) 讀取-修飾-寫入訊息、3) 點對點未排序訊息以及4) 點對點排序訊息。大部分的這些訊息是由節點N0-N5內的客戶邏輯產生，其可包括，例如，中央處理單元（CPU）或記憶體控制器。

【0042】在廣播訊息中，傳輸節點的客戶邏輯（例如，節點N0的客戶邏輯240）將廣播訊息發送至其接收相鄰者（例如，節點N1，經由傳輸FIFO 232、訊息組譯器260以及多工器265）。在接收相鄰者內的節點邏輯（例如，節點邏輯310）接收廣播訊息。作為回應，此節點邏輯將廣播訊息路由至其相對應的客戶邏輯（例如，客戶邏輯340，經由其相對應的接收FIFO 331）。此節點邏輯也將廣播訊息轉送至其接收相鄰者（例如，節點N2，經由多工器365）。當原本的傳輸節點（例如，節點N0）從其傳輸相鄰者（例如，節點N5）接收廣播訊息時停止轉送。例如，快取列無效要求可作為廣播訊息被傳輸。

【0043】在讀取-修飾-寫入訊息中，傳輸節點的客戶邏輯（例如，節點N0的客戶邏輯240）將一個訊息發送至其接收相鄰者（例如，節點N1，經由傳輸FIFO 232、訊息組譯器260以及多工器265）。在接收相鄰者內的節點邏輯（例如，節點邏輯310）接收讀取-修飾-寫入訊息，並將此訊息轉送至其相對應的客戶邏輯（例如，客戶邏輯340，經由其相對應的接收FIFO 331）。如果需要的話，此客戶邏輯可修飾此訊息。客戶邏輯隨後將讀取-修飾-寫入訊息轉送至其接收相鄰者（例如，節點N2，經由傳輸FIFO 332、訊息組譯器360以及多工器365）。注意，客戶邏輯在接收讀取-修飾-寫入訊息之後將以其自己的裁量（例如，使用相對應的流程控制以及頻寬分配邏輯）轉送讀取-修飾-寫入訊息，其為（本地時鐘訊號的）某數目的循環。此轉送過程由每個節點重複，直到讀取-修飾-寫入訊息由原

本的傳輸節點的傳輸相鄰者（例如，節點N5）回歸到原本的傳輸節點（例如，節點N0）。原本的傳輸節點處理所接收的讀取-修飾-寫入，並停止讀取-修飾-寫入訊息的傳播。上述與設定過程有關的ID_設定訊息是讀取-修飾-寫入訊息的一個範例。依照另一個範例，來自CPU的快取列窺探反應可作為讀取-修飾-寫入訊息而被傳輸。

【0044】在點對點未排序訊息中，傳輸節點的客戶邏輯（例如，節點N0的客戶邏輯240）將一個訊息發送至其接收相鄰者（例如，節點N1、經由傳輸FIFO 232、訊息組譯器260以及多工器265）。點對點未排序訊息定址其他節點的特定一個（例如，節點N2），其中所定址的節點的節點ID值被包括在此訊息中。在傳輸節點以及所定址的節點之間的每個中間節點內的節點邏輯將此訊息轉送至其接收相鄰者，直到此訊息到達所定址的節點。例如，在節點N1中的節點邏輯310可從節點N0接收點對點未排序訊息，比較其節點ID值（例如，「1」）與包括在訊息中的節點ID值（例如，「2」），並作為回應，決定此訊息不定址節點N1。作為回應，節點邏輯310（經由多工器365）將點對點未排序訊息轉送至接收節點N2。注意，節點邏輯310不將訊息轉送至非定址的節點N1內的客戶邏輯340。在接收點對點未排序訊息之後，在節點N2內的節點邏輯然後會決定包括在點對點未排序訊息中的節點ID值（例如，「2」）等於其本身的節點ID值（例如，「2」），並作為回應，（經由其相對應的接收FIFO）將點對點未排序訊息提供至其相對應的客戶邏輯。點對點未排序訊息不在環狀網路100上的定址節點之外傳輸（例如，節點N2不將點對點未排序訊息轉送至節點N3）。在一個實施方式中，對於讀取要求的數據反應可作為點對點未排序訊息而被傳輸。

【0045】在點對點排序訊息中，傳輸節點的客戶邏輯（例如，節點N0的客戶邏輯240）將一個訊息發送至其接收相鄰者（例如，節點N1，經由傳輸FIFO 232、訊息組譯器260以及多工器265）。點對點排序訊息定址其他節點中的特定

一個（例如，節點N2），其中在此訊息中包括了所定址節點的節點ID值。傳輸節點也將來自其自身的節點ID暫存器的其自身獨特的節點ID值（例如，來自節點ID暫存器250的節點ID值「0」）以及來自其自身時間戳記計數器（例如，時間戳記計數器255）的目前時間戳記值（TS）附加至此訊息。在傳輸節點（例如節點N0）內，訊息組譯器260在傳輸點對點排序訊息之後增加時間戳記計數器255。

【0046】在傳輸節點以及所定址節點之間的每個中間節點（例如，節點N1）內的節點邏輯比較其節點ID值（例如，「1」）與包括在點對點排序訊息中的所定址節點ID值（例如，「2」），並做出反應，決定此訊息不定址此中間節點。做出反應，此中間節點N1的節點邏輯310（經由多工器365）將點對點排序訊息轉送至其相對應的接收節點N2。在接收點對點排序訊息之後，在每個節點中的節點邏輯增加其相對應的時間戳記計數器，並將排序訊息儲存在其訊息竊取程式中（例如，在節點N1內，節點邏輯310增加其時間戳記計數器355並將排序訊息儲存在其訊息竊取程式330中）。

【0047】在接收點對點排序訊息之後，在所定址節點（例如，節點N2）內的節點邏輯增加其時間戳記計數器，並決定包括在點對點排序訊息中的所定址的節點ID值等於其自身的節點ID值（例如，「2」）。作為回應，在所定址節點N2內的節點邏輯造成儲存在其色彩位元暫存器中的色彩位元值與點對點排序訊息一起被儲存在其訊息竊取程式中。在所定址節點N2內的節點邏輯然後將排序訊息轉送至其相對應的接收節點（例如，節點N3）。此轉送繼續直到排序訊息到達原本的傳輸節點（例如，節點N0），其中每個中間節點增加其時間戳記計數器並將排序訊息擷取至其訊息竊取程式中。在接收排序訊息之後，原本的傳輸節點（例如，節點N0）不增加其時間戳記計數器。在此時，藉由檢驗其自身的訊息竊取程式的內容，原本的傳輸節點可決定其排序訊息是否將由所定址的

節點授予優先權。所定址節點等待直到脈衝訊息改變儲存在其相對應的色彩位元暫存器中的色彩位元的狀態，以解析點對點排序訊息的優先權。以此方式，即使訊息以不同的順序由每個節點接收，每個節點中的訊息竊取程式中的資訊提供了發出訊息的順序的一致觀點。在訊息竊取程式中的資訊將以下面更詳細描述的方式用於仲裁以及衝突解決。

【0048】如上所述，每個節點N0-N5將每一個所接收的點對點排序訊息暫時儲存至其訊息竊取程式中（例如，節點N0以及N1的訊息竊取程式230以及330分別儲存由節點邏輯210以及310接收的排序訊息）。在所描述的實施方式中，每一個訊息竊取程式是FIFO，其中訊息竊取程式FIFO的深度是環狀網路100的設計參數。在一個實施方式中，選擇每個訊息竊取程式的深度以提供佈局面積、操作速度、功率消耗以及所需頻寬的適當取捨。如下所述，訊息竊取程式的深度限制了點對點排序訊息要求的數目，其在任何給定時間在環狀網路100上都可能重要的。

【0049】依照一個實施方式，環狀網路100的每個節點被指定了點對點排序訊息要求的配額，其中此配額範圍從0至等於訊息竊取程式深度減去1的值。每個節點不能具有超過其配額的重要點對點排序訊息的數目，其中點對點排序訊息被視為是重要的，直到其循環回其起源節點。指定至每個節點的配額不需要相同。例如，不能夠發送點對點排序訊息的節點的配額可被設定為零，而性能重要的節點（例如，CPU）可被指定較大配額的點對點排序訊息。然而，可能重要點對點排序訊息要求的總數（例如，指定至每個節點N0-N5的配額的總和）必須少於訊息竊取程式的深度。

【0050】每個訊息竊取程式擷取穿過相對應節點的點對點排序訊息。如下面更詳細描述的，這些所擷取的排序訊息用以識別在環狀網路100上發佈訊息的順序。

【0051】如上所述，節點N0-N5中的每一個時間戳記計數器被初始化至「0」值，且每一者當相對應的節點傳輸原本的點對點排序訊息、或接收在另一個節點起源的點對點排序訊息時增加。每一個時間戳記計數器計數上至MAX_COUNT值。在到達MAX_COUNT值之後，時間戳記計數器概括至「0」的值。在一個實施方式中，將MAX_COUNT值選擇成是每個訊息竊取程式記項數目的至少兩倍，以避免計數器混淆。

【0052】在一個實施方式中，初始化（或概括）時間戳記計數器也反轉時間戳記計數器值的符號位元。如下面更詳細描述的，節點N0-N5使用時間戳記值（以及符號位元）以決定在訊息匯流排102上接收的點對點排序訊息的順序。

【0053】一般而言，如果與兩個點對點排序訊息相關聯的時間戳記值的符號位元相等，則具有較低時間戳記值的點對點排序訊息是較舊的（且因此具有優先權）。如果與兩個點對點排序訊息相關聯的時間戳記值的符號位元不同，則具有較高時間戳記值的點對點排序訊息是較舊的（且因此具有優先權）。如果符號位元以及與兩個點對點排序訊息相關聯的時間戳記值相等，則兩個點對點排序訊息被同時發送。在這些條件下，「連結斷路器」是必要的，以提供解析。例如，可將位址雜湊函數應用至每個點對點排序訊息以產生隨機值。與較大值相關聯的點對點排序訊息可被標示為較舊的訊息並授予優先權。其他方法也可用在這些條件下提供優先權解析。

【0054】使用特定的範例，現在將更詳細地描述依照本發明的一個實施方式決定點對點排序訊息的優先權的方式。在此範例中，根節點N0的客戶邏輯240包括第一CPU（例如，CPU_0），分支節點N3的客戶邏輯包括第二CPU（例如，CPU_1），以及分支節點N5的客戶邏輯包括系統記憶體。兩個CPU_0（節點N0）以及CPU_1（節點N3）企圖對於節點N5的系統記憶體中的相同快取列「A」有

專用的存取。在所描述的範例中，假設節點N0-N5每一者的時間戳記計數器一開始具有「0」的值。

【0055】第8A圖是示出在時間T0的環狀網路100的狀態的表800，包括：由節點N0-N5執行的動作、由節點N0-N5在訊息匯流排102上提供的輸出訊息、以及節點N0-N5的訊息竊取程式的內容、節點N0-N5的時間戳記計數器的值、節點N0-N5的色彩位元的值、以及在環狀網路100中不斷地移動的脈衝訊息的位置。在時間T0，根節點N0的CPU_0在訊息匯流排102上傳輸訊息M*0_5(0)（例如，為MSG_01）。在本文中所描述的範例中，使用了M*X_Y(Z)的訊息匯流排記號，其中M*具體指明了點對點排序訊息，X具體指明了來源節點ID（即，產生訊息的節點N0的節點ID暫存器250的內容），Y與目的地節點ID相對應（即，節點N5的節點ID暫存器的內容以接收點對點排序訊息）、以及Z與起源節點N0的時間戳記值相對應（在訊息一開始被發送的時候）。因此，訊息M*0_5(0)具體指明了從節點N0傳輸至節點N5的點對點排序訊息當節點N0的時間戳記計數器具有0的值時開始。在所描述的範例中，訊息M*0_5(0)包括額外資訊（未示出），額外資訊指出交易是對於節點N5的系統記憶體中快取列「A」的存取。一旦在訊息匯流排102上傳輸點對點排序訊息M*0_5(0)，節點N0的CPU_0將其時間戳記計數器255從0的值增加至1的值。節點N0的訊息竊取程式230儲存訊息M*0_5(0)。

【0056】也在時間T0，節點N2在訊息匯流排102上傳輸排序點對點訊息M*2_4(0)（例如，為MSG_23）。依照上面定義的命名法，訊息M*2_4(0)具體指明了從節點N2傳輸至節點N4的點對點排序訊息，當節點N2的時間戳記計數器具有0的值時開始。一旦在訊息匯流排102上傳輸點對點排序訊息M*2_4(0)，節點N2將其時間戳記計數器從0的值增加至1的值。節點N2的訊息竊取程式儲存訊息M*2_4(0)。

【0057】注意，在時間T0，節點N1以及N3-N5不開始任何動作（或在訊息匯流排102上提供任何訊息），且節點N1以及N3-N5的訊息竊取程式是空的。因為這些節點N1以及N3-N5沒有一個已接收或傳輸點對點排序訊息，這些節點的時間戳記計數器維持不變。

【0058】在所示出的範例中，脈衝訊息在時間T0到達節點N4，由此造成節點N4的色彩位元從「0」值過渡至「1」值。注意，為了呈現範例，色彩位元值以及脈衝位置是任意選擇的。

【0059】第8B圖是示出在下一個循環T1期間的環狀網路100的狀態的表801。在此時，節點N0-N5沒有一個傳輸任何新的訊息。訊息M*0_5（0）是由節點N1接收，且此節點N1將訊息M*0_5（0）儲存在其訊息竊取程式中。因為節點N1已接收點對點排序訊息，節點N1將其時間戳記計數器從0增加至1。在節點N1內的節點邏輯310然後將點對點排序訊息M*0_5（0）傳輸至節點N2。

【0060】此外，點對點排序訊息M*2_4（0）是由節點N3接收，且此節點N3將訊息M*2_4（0）儲存在其訊息竊取程式中。因為節點N3已接收點對點排序訊息，節點N3將其時間戳記計數器從0增加至1。在節點N3內的節點邏輯然後將點對點排序訊息M*2_4（0）傳輸至節點N4。

【0061】脈衝訊息在循環T1期間到達節點N5，由此造成節點N5的色彩位元從「0」值雙態觸變至「1」值。

【0062】第8C圖是示出在下一個循環T2期間的環狀網路100的狀態的表802。在此循環T2期間，節點N3的CPU_1在訊息匯流排102上傳輸訊息M*3_5（1）（例如，為MSG_34）並將此訊息M*3_5（1）儲存在其訊息竊取程式中。如上所述，訊息M*3_5（1）具體指明了從節點N3至節點N5的點對點排序訊息。注意，在產生訊息M*3_5（1）時，節點N3的時間戳記值是「1」。在所描述的範例中，訊息M*3_5（1）包括額外資訊（未示出），額外資訊指出交易是對於節點N5的

系統記憶體中的快取列「A」的存取。因為節點N3已傳輸排序訊息M*3_5(1)，節點N3的時間戳記計數器從1增加至2。注意，在循環T2期間，節點N0-N2以及N4-N5不開始任何新的動作。

【0063】同樣地在循環T2期間，點對點排序訊息M*0_5(0)是由節點N2接收，且此節點N2將訊息M*0_5(0)儲存在其訊息竊取程式中。因為節點N2已接收排序訊息，節點N2將其時間戳記計數器從1增加至2。在節點N2內的節點邏輯然後將點對點排序訊息M*0_5(0)傳輸至節點N3。

【0064】此外，點對點排序訊息M*2_4(0)是由節點N4接收，且此節點N4將訊息M*2_4(0)儲存在其訊息竊取程式中。因為節點N4已接收排序訊息，節點N4將其時間戳記計數器從0增加至1。在節點N4內的節點邏輯然後將點對點排序訊息M*2_4(0)傳輸至節點N5。

【0065】在循環T2期間，脈衝訊息到達節點N0，由此造成節點N0的色彩位元從「1」值雙態觸變至「0」值。

【0066】第8D圖是示出在下一個循環T3期間的環狀網路100的狀態的表803。在此時，節點N0-N5沒有一個傳輸任何新的訊息。點對點排序訊息M*0_5(0)是由節點N3接收，且此節點N3將訊息M*0_5(0)儲存在其訊息竊取程式中。因為節點N3已接收排序訊息，節點N3將其時間戳記計數器從2增加至3。在節點N3內的節點邏輯然後將點對點排序訊息M*0_5(0)傳輸至節點N4。

【0067】此外，點對點排序訊息M*2_4(0)是由節點N5接收，且此節點N5將訊息M*2_4(0)儲存在其訊息竊取程式中。因為節點N5已接收排序訊息，節點N5將其時間戳記計數器從0增加至1。在節點N5內的節點邏輯然後將點對點排序訊息M*2_4(0)傳輸至節點N0。

【0068】此外，點對點排序訊息M*3_5(1)是由節點N4接收，且此節點N4將訊息M*3_5(1)儲存在其訊息竊取程式中。因為節點N4已接收排序訊息，

節點N4將其時間戳記計數器從1增加至2。在節點N4內的節點邏輯然後將點對點排序訊息M*3_5（1）傳輸至節點N5。

【0069】在循環T3期間，脈衝訊息到達節點N1，由此造成節點N1的色彩位元從「1」值雙態觸變至「0」值。

【0070】第8E圖是示出在下一個循環T4期間的環狀網路100的狀態的表804。在此時，節點N0-N5沒有一個傳輸任何新的訊息。點對點排序訊息M*0_5（0）是由節點N4接收，且此節點N4將訊息M*0_5（0）儲存在其訊息竊取程式中。因為節點N4已接收排序訊息，節點N4將其時間戳記計數器從2增加至3。在節點N4內的節點邏輯然後將點對點排序訊息M*0_5（0）傳輸至節點N5。

【0071】此外，點對點排序訊息M*2_4（0）是由節點N0接收，且此節點N0將訊息M*2_4（0）儲存在其訊息竊取程式中。因為節點N0已接收排序訊息，節點N0將其時間戳記計數器從1增加至2。在節點N0內的節點邏輯然後將點對點排序訊息M*2_4（0）傳輸至節點N1。

【0072】此外，點對點排序訊息M*3_5（1）是由節點N5接收，且此節點N5將訊息M*3_5（1）儲存在其訊息竊取程式中。因為節點N5已接收排序訊息，節點N5將其時間戳記計數器從1增加至2。因為點對點排序訊息M*3_5（1）定址節點N5，節點N5在接收訊息M*3_5（1）時決定並儲存色彩位元的狀態（例如，色彩位元=1）。在一個實施方式中，此色彩位元與點對點排序訊息M*3_5（1）一起儲存在節點N5的訊息竊取程式中。為了確保節點N5以適當的順序解析點對點排序訊息，節點N5必須等待以解析所接收的點對點排序訊息M*3_5（1），直到色彩位元的值被雙態觸變（例如，色彩位元在節點N5內過渡至0的值）。節點N5的節點邏輯然後將點對點排序訊息M*3_5（1）傳輸至節點N0。

【0073】在循環T4期間，脈衝訊息到達節點N2，由此造成節點N2的色彩位元從「1」值雙態觸變至「0」值。

【0074】第8F圖是示出在下一個循環T5期間的環狀網路100的狀態的表805。在此時，節點N0-N5沒有一個傳輸任何新的訊息。點對點排序訊息M*0_5(0)是由節點N5接收，且此節點N5將訊息M*0_5(0)儲存在其訊息竊取程式中。因為節點N5已接收排序訊息，節點N5將其時間戳記計數器從2增加至3。因為點對點排序訊息M*0_5(0)定址節點N5，在接收訊息M*0_5(0)時節點N5決定並儲存色彩位元的狀態（例如，色彩位元=1）。為了確保節點N5以適當的順序解析排序訊息，節點N5必須等待以解析所接收的點對點排序訊息M*0_5(0)，直到色彩位元的值被雙態觸變（例如，色彩位元在節點N5內過渡至0的值）。節點N5的節點邏輯然後將點對點排序訊息M*0_5(0)傳輸至節點N0。

【0075】此外，點對點排序訊息M*2_4(0)是由節點N1接收，且此節點N1將訊息M*2_4(0)儲存在其訊息竊取程式中。因為節點N1已接收排序訊息，節點N1將其時間戳記計數器從1增加至2。節點N1的節點邏輯然後將訊息M*2_4(0)傳輸至節點N2。

【0076】此外，點對點排序訊息M*3_5(1)是由節點N0接收，且此節點N0將訊息M*3_5(1)儲存在其訊息竊取程式中。因為節點N0已接收排序訊息，節點N0將其時間戳記計數器從2增加至3。節點N0的節點邏輯然後將點對點排序訊息M*3_5(1)傳輸至節點N1。

【0077】在循環T5期間，脈衝訊息到達節點N3，由此造成節點N3的色彩位元從「1」值雙態觸變至「0」值。

【0078】第8G圖是示出在下一個循環T6期間的環狀網路100的狀態的表806。在此時，節點N0-N5沒有一個傳輸任何新的訊息。點對點排序訊息M*0_5(0)循環回起源的節點N0。作為回應，節點N0停止環狀網路200上此訊息M*0_5(0)的傳播。注意，回應於接收到其原本傳輸的排序訊息，節點N0不增加其時間戳記計數器。一旦接收訊息M*0_5(0)，節點N0存取其訊息竊取程式，並比

較所儲存的排序訊息以決定其點對點排序訊息 $M*0_5(0)$ 具有優於定址相同的節點N5的其他點對點排序訊息 $M*3_5(1)$ 的優先權（因為訊息 $M*0_5(0)$ 具有比其他排序訊息 $M*3_5(1)$ 低的時間戳記）。因此，節點N0被通知，其點對點排序訊息 $M*0_5(0)$ 將被節點N5授予優先權，且將由節點N5處理。

【0079】此外，點對點排序訊息 $M*2_4(0)$ 循環回起源的節點N2。作為回應，節點N2停止環狀網路200上此訊息 $M*2_4(0)$ 的傳播。回應於接收到其原本傳輸的排序訊息，節點N2不增加其時間戳記計數器。一旦接收訊息 $M*2_4(0)$ ，節點N2存取其訊息竊取程式，並比較所儲存的排序訊息以決定其訊息 $M*2_4(0)$ 具有優於其訊息竊取程式中所儲存的其他排序訊息的優先權（因為這些其他訊息沒有一個定址節點N4）。因此，節點N2被通知其訊息 $M*2_4(0)$ 將被節點N4授予優先權，且將由節點N4處理。

【0080】此外，點對點排序訊息 $M*3_5(1)$ 是由節點N1接收，且此節點N1將訊息 $M*3_5(1)$ 儲存在其訊息竊取程式中。因為節點N1已接收排序訊息，節點N1將其時間戳記計數器從2增加至3。節點N1的節點邏輯然後將點對點排序訊息 $M*3_5(1)$ 傳輸至節點N2。

【0081】在循環T6期間，脈衝訊息到達節點N4，由此造成節點N4的色彩位元從「1」值雙態觸變至「0」值。注意，回應於節點N4的色彩位元的雙態觸變，節點N4解析了儲存在其訊息竊取程式中的排序訊息的優先權。如上所述，節點N4只將優先權授予至在定址節點N4的其訊息緩衝中的排序訊息 $M*2_4(0)$ 。

【0082】第8H圖是示出在下一個循環T7期間的環狀網路100的狀態的表807。在此時，節點N0-N5沒有一個傳輸任何新的訊息。在循環T7期間，脈衝訊息到達節點N5，由此造成節點N5的色彩位元從「1」值雙態觸變至「0」值。回應於此色彩位元的雙態觸變，節點N5執行儲存在其訊息竊取程式中的排序訊息的仲裁，同時色彩位元具有先前的「1」的值。因此，節點N5決定點對點排序訊

息 $M*0_5(0)$ 具有優於其他點對點排序訊息 $M*3_5(1)$ 的優先權，因為訊息 $M*0_5(0)$ 具有比其他訊息 $M*3_5(1)$ 低的時間戳記值。作為回應，節點N5將存取在系統記憶體的快取列「A」中由訊息 $M*0_5(0)$ 要求的數據。注意，節點N5將產生傳輸此要求的數據至節點N0的分開的訊息（未示出）。進一步注意，節點N5忽略儲存在其訊息竊取程式中的點對點排序訊息 $M*2_4(0)$ ，因為此訊息不定址節點N5。

【0083】此外，點對點排序訊息 $M*3_5(1)$ 是由節點N2接收，且此節點N2將訊息 $M*3_5(1)$ 儲存在其訊息竊取程式中。因為節點N2已接收排序訊息，節點N2將其時間戳記計數器從2增加至3。節點N2的節點邏輯然後將點對點排序訊息 $M*3_5(1)$ 傳輸至節點N3。

【0084】第8I圖是示出在下一個循環T8期間的環狀網路100的狀態的表808。在此時，節點N0-N5沒有一個傳輸任何新的訊息。點對點排序訊息 $M*3_5(1)$ 循環回起源的節點N3。作為回應，節點N3停止環狀網路200上此訊息 $M*3_5(1)$ 的傳播。回應於接收到其原本傳輸的點對點排序訊息，節點N3不增加其時間戳記計數器。一旦接收訊息 $M*3_5(1)$ ，節點N3存取其訊息竊取程式，並比較所儲存的排序訊息以決定由節點N0傳輸的點對點排序訊息 $M*0_5(0)$ 具有優於其自身的點對點排序訊息 $M*3_5(1)$ 的優先權（因為訊息 $M*0_5(0)$ 具有比訊息 $M*3_5(1)$ 低的時間戳記值）。因此，節點N3被通知，其訊息 $M*3_5(1)$ 將不被節點N5授予優先權，且將不由節點N5處理。做出反應，節點N3可決定應重傳訊息 $M*3_5(1)$ 。

【0085】以前述方式，點對點排序訊息可由環狀網路100處理，而不需要複雜的中央控制電路。所需的控制邏輯被有利地分散在節點之中，以至於不需要中央控制器。此外，不需要設計/製造橫跨大晶粒面積的大規模低偏斜時鐘樹，

由此有利地簡化網路的實體實施。此外，可以最小的潛時並以高頻率操作環狀網路100。

【0086】如上所述，在節點N0-N5內的客戶邏輯可控制訊息匯流排102上訊息的流動。因為環狀網路的平均頻寬是2（對剖頻寬），頻寬分配方針以及流程控制邏輯必定被包括在每個節點的客戶邏輯內。數個熟知的技術，例如循環或學分制方案，可由每個節點內的頻寬分配以及流程控制電路使用。依照所提供的描述，每個節點的頻寬分配以及流程控制邏輯將在客戶邏輯時鐘域中操作，且因此可以與環狀網路時鐘域的參考時鐘訊號CK_REF不同的頻率操作。

【0087】雖然已關於數個實施方式描述了本發明，要了解的是，此發明不限於所揭露的實施方式，但能夠有各種修飾，其對於本領域的技術人員而言是顯而易見的。因此，本發明只由下述申請專利範圍所限制。

【符號說明】

【0088】

100：分散控制同步環狀網路

101：時鐘路徑

101D：時鐘路徑結構

102：訊息匯流排

102D：訊息匯流排結構

201：主時鐘產生器

202：延遲線

205：再同步器

210、310：節點邏輯

220：同步器

230、330：訊息竊取程式

231、331：接收先進先出（FIFO）記憶體

232、332：傳輸FIFO記憶體

240、340：客戶邏輯

250、350：節點識別符（ID）記憶體

251、351：色彩位元暫存器

255、355：時間戳記（TS）計數器

260、360：訊息組譯器

265、365：多工器

270：虛線

302：時鐘延遲線

320：同步電路

CK_REF：參考時鐘

CK0：本地時鐘訊號

CK1：本地時鐘域

CLK_01、CLK_12、CLK_23、CLK_34、CLK_45、CLK_50：時鐘訊號

MSG_01、MSG_12、MSG_23、MSG_34、MSG_45、MSG_50：輸出訊息

N0：根節點

N1-N5：分支節點

【發明申請專利範圍】

【第1項】一種環狀網路，包括：

多個節點，該多個節點包括一根節點以及一或更多個分支節點，其中該根節點被耦合以接收一參考時鐘訊號；

一環狀時鐘訊號路徑，其將該參考時鐘訊號從該根節點經由該一或更多個分支節點的每一個分支節點路由回到該根節點；以及

一環狀訊息路徑，其與該環狀時鐘訊號路徑平行延伸，其中該根節點以及該一或更多個分支節點在該訊息路徑上傳輸訊息，

其中回應於在該時鐘訊號路徑上接收的該參考時鐘訊號，該多個節點中的每個節點包括鎖存在該訊息路徑上接收的訊息的一相對應的節點邏輯，

其中當傳輸訊息時，該多個節點中的每個節點引進關於參考時鐘訊號的一相對應的節點延遲，

其中該時鐘訊號路徑包括在該多個節點中的每個節點內的一延遲線，

其中在該相對應的節點邏輯鎖存在該訊息路徑上接收的該訊息以回應於該參考時鐘訊號後，每個延遲線引進一延遲至在該時鐘訊號路徑上的該參考時鐘訊號中，

其中由在該多個節點中的每個節點內的該相對應的延遲線引進的該延遲是基於該相對應的節點延遲而被選擇。

【第2項】如申請專利範圍第1項所述的環狀網路，其中訊息以一單向方式在該訊息路徑上傳輸。

【第3項】如申請專利範圍第1項所述的環狀網路，其中該參考時鐘訊號是用於在該訊息路徑上傳輸的訊息的同步的唯一時鐘訊號。

【第4項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中每個節點包括在由一相對應的本地時鐘訊號定義的一本地時鐘域中操作的一客戶邏輯。

【第5項】如申請專利範圍第4項所述的環狀網路，其中相對於每個本地時鐘訊號，該參考時鐘訊號具有一較高頻率。

【第6項】如申請專利範圍第4項所述的環狀網路，更包括能夠在該多個節點中每個節點內跨越該第一時鐘域以及該本地時鐘域通訊的同步電路系統。

【第7項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中每個節點更包括一相對應的時間戳記計數器。

【第8項】如申請專利範圍第7項所述的環狀網路，更包括用於初始化在該多個節點中每個節點中的該時間戳記計數器的裝置。

【第9項】如申請專利範圍第8項所述的環狀網路，其中回應於產生一排序訊息及回應於從另一個節點接收一排序訊息，該多個節點中的每個節點更包括用於增加該相對應的時間戳記計數器的裝置，其中每一個排序訊息定址該多個節點中的一特定節點。

【第10項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中每個節點包括儲存一色彩位元值的一色彩位元暫存器，該色彩位元值是由不斷地橫越該環狀訊息路徑的一脈衝訊息雙態觸變。

【第11項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中每個節點包括儲存一獨特節點識別符位址的一節點識別暫存器。

【第12項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中每個節點包括儲存在該訊息路徑上接收的多個排序訊息的一相對應的訊息竊取程式。

【第13項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中每個節點包括用於仲裁在該訊息路徑上接收的多個排序訊息的一優先權的裝置。

【第14項】如申請專利範圍第1項所述的環狀網路，其中每個延遲線的該延遲是可調整的。

【第15項】如申請專利範圍第1項所述的環狀網路，其中該根節點包括接收該訊息路徑上的訊息以及將該接收的訊息與該參考時鐘訊號再同步的一再同步電路。

【第16項】如申請專利範圍第1項所述的環狀網路，其中該多個節點中的每個節點更包括：

儲存一相對應的時間戳記值的一相對應的時間戳記計數器；

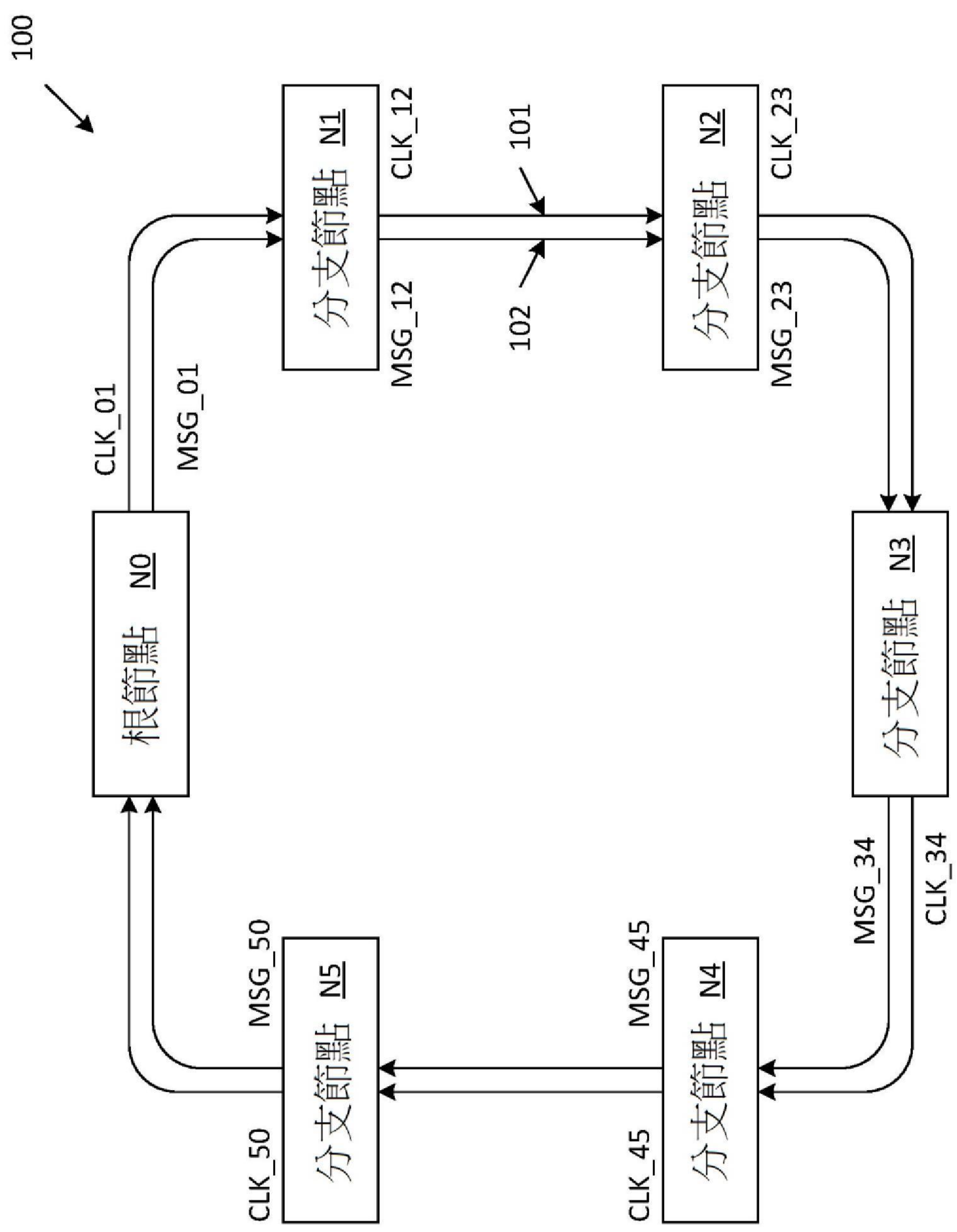
用於回應於產生一排序訊息及回應於從另一個節點接收一排序訊息而增加該相對應的時間戳記計數器的裝置，其中每個排序訊息定址該多個節點中的一特定節點；

用於將該相對應的時間戳記值附加至由該節點產生的每個排序訊息的裝置；

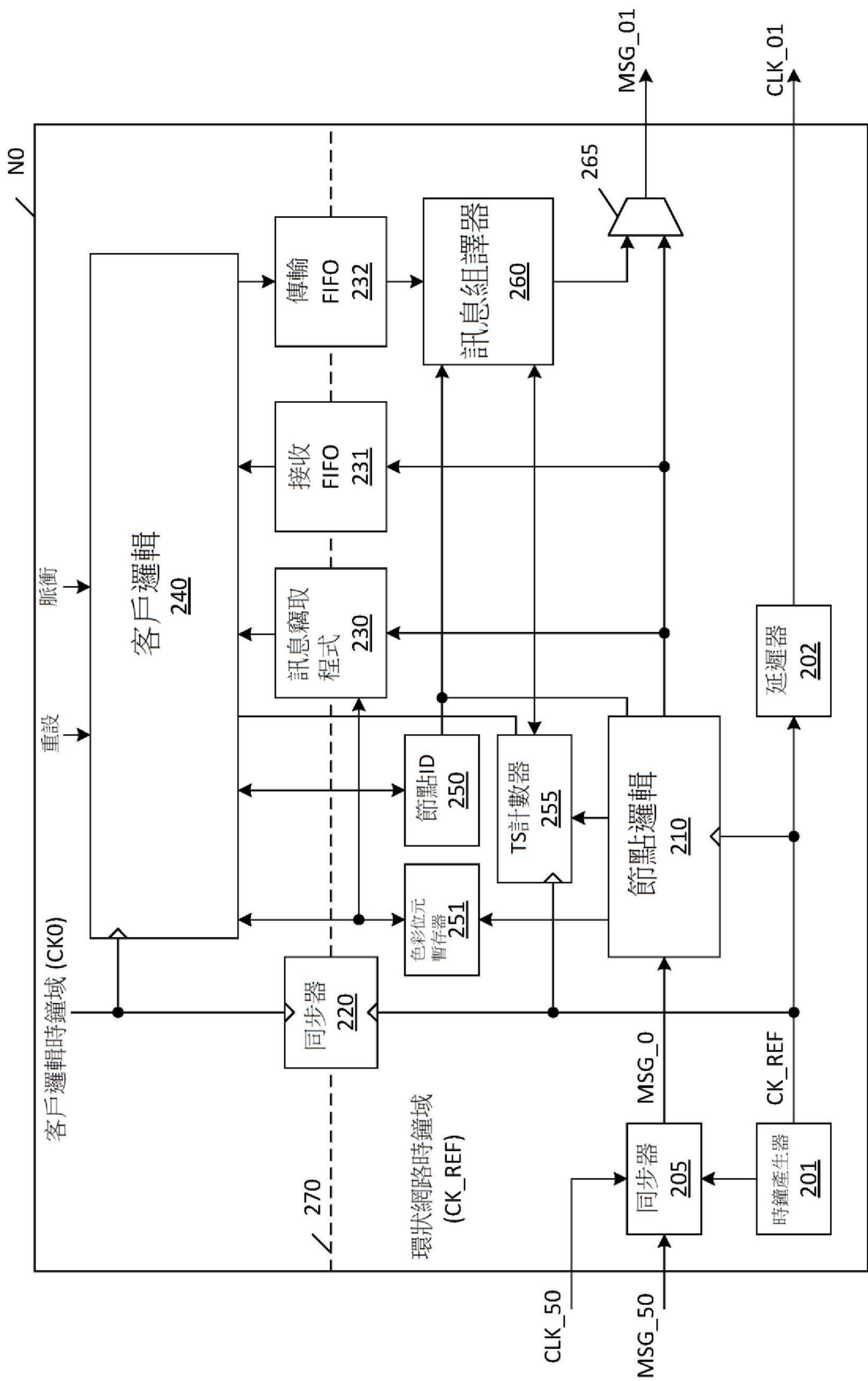
儲存在該訊息路徑上接收的多個排序訊息的一相對應的訊息竊取程式；

儲存一相對應的色彩位元值的一相對應的色彩位元暫存器，其中該相對應的色彩位元值由不斷地橫跨該環狀訊息路徑的一脈衝訊息而被雙態觸變；以及

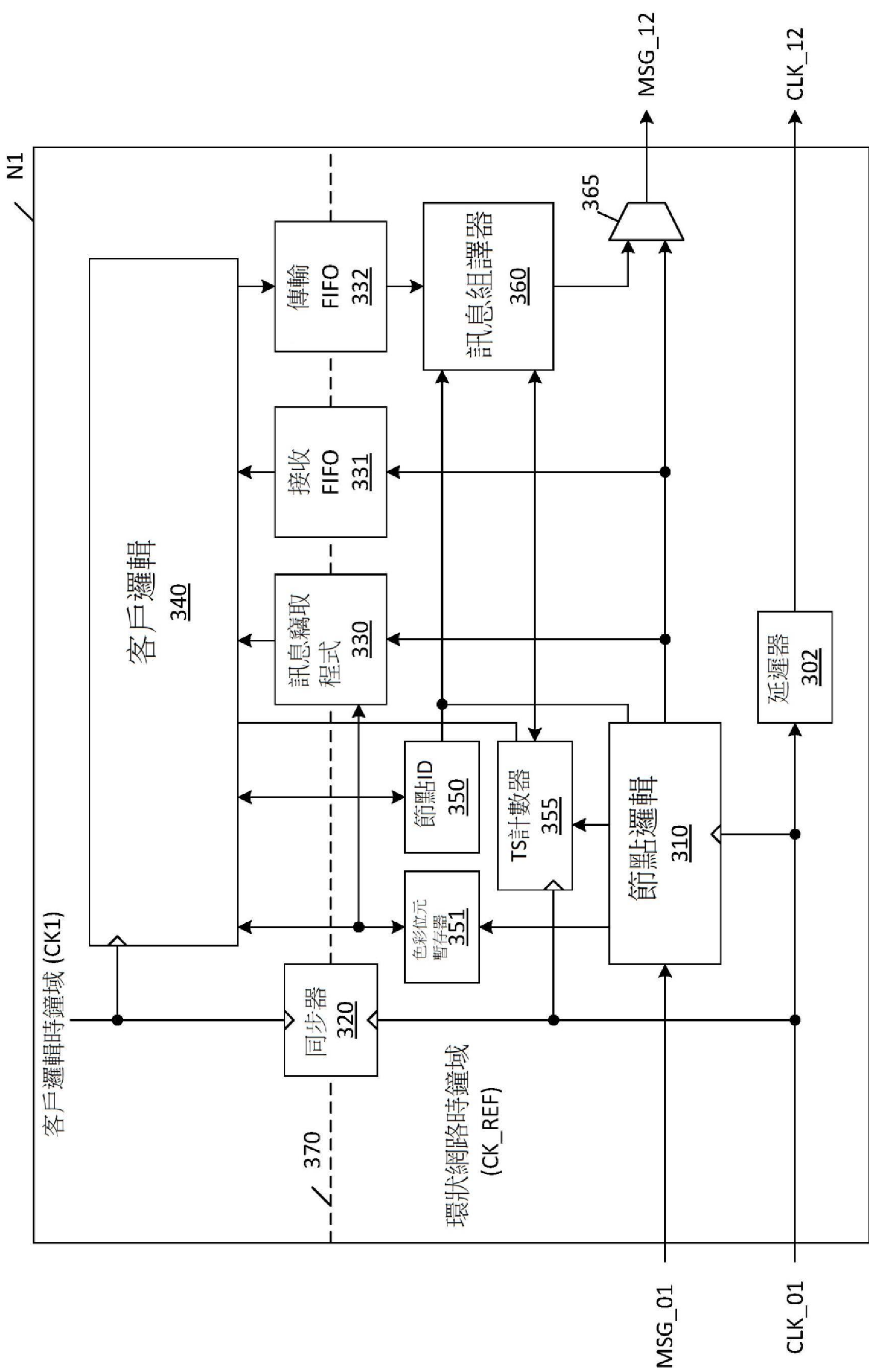
用於回應於該相對應的色彩位元值被雙態觸變而仲裁儲存於該相對應的訊息竊取程式的多個排序訊息的該優先權的裝置，其中該優先權是基於附加至該多個排序訊息的多個時間戳記值而被授予。



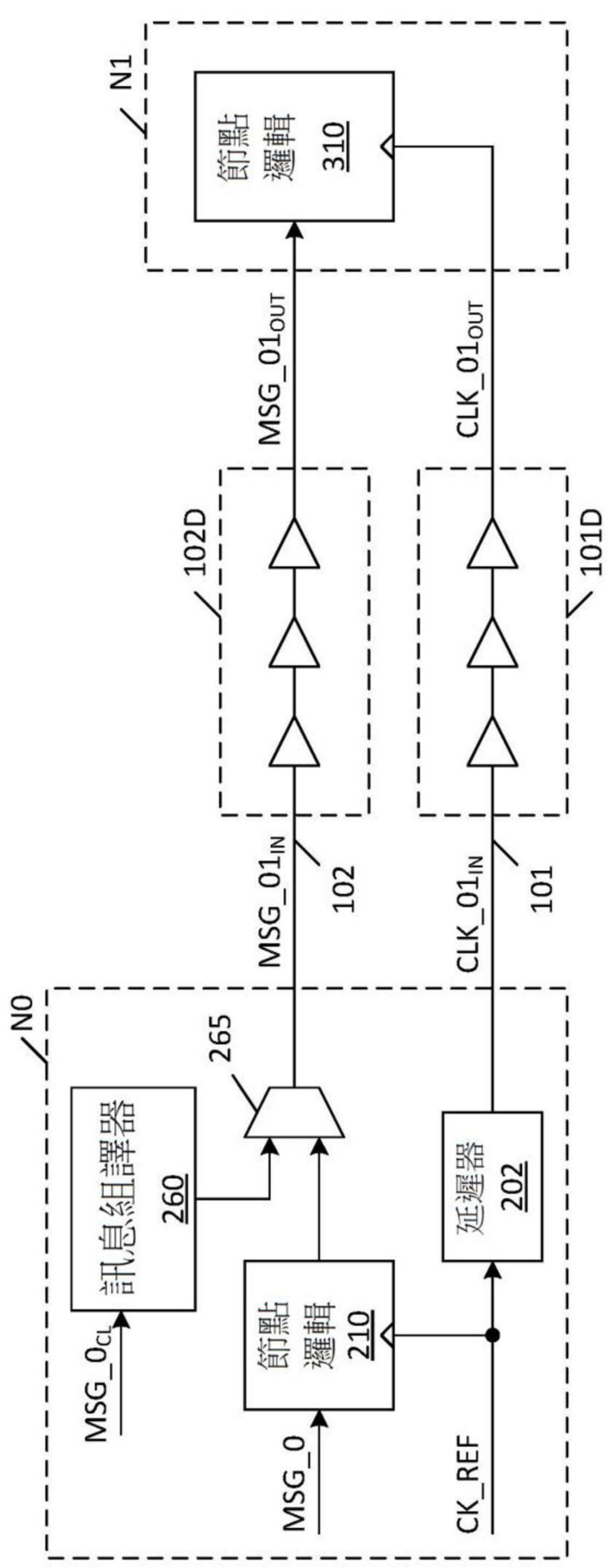
第 1 圖



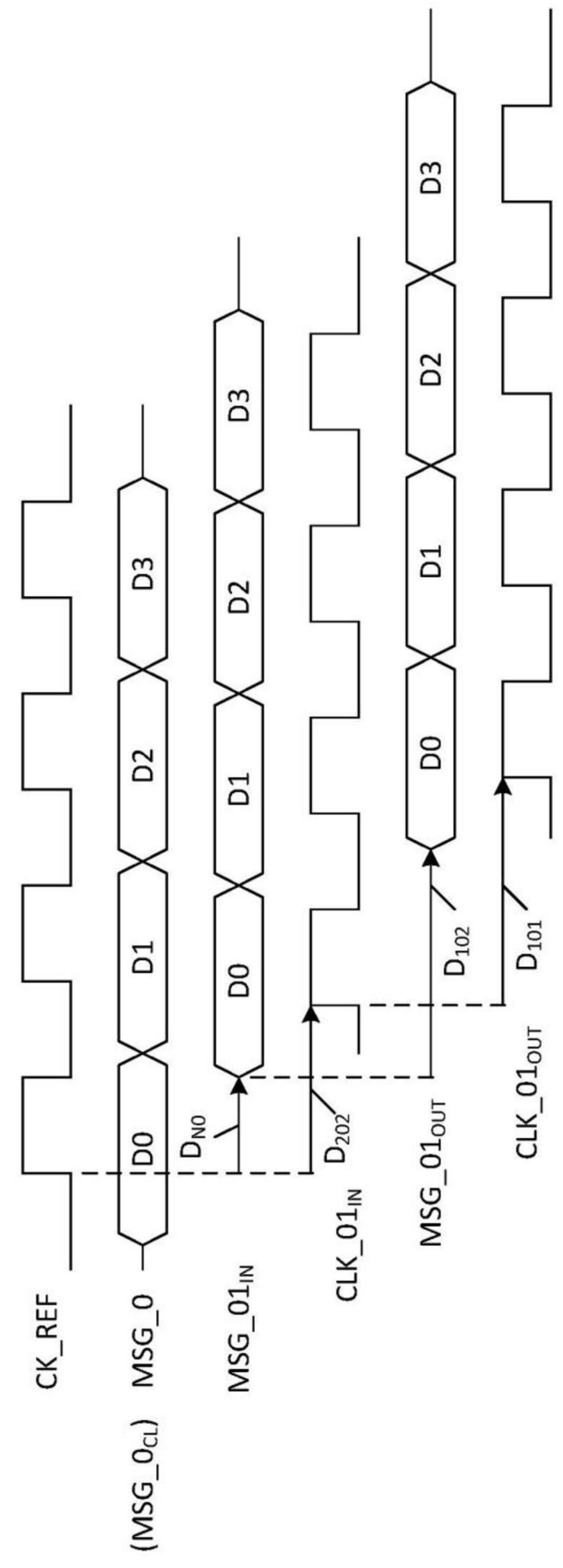
第2圖



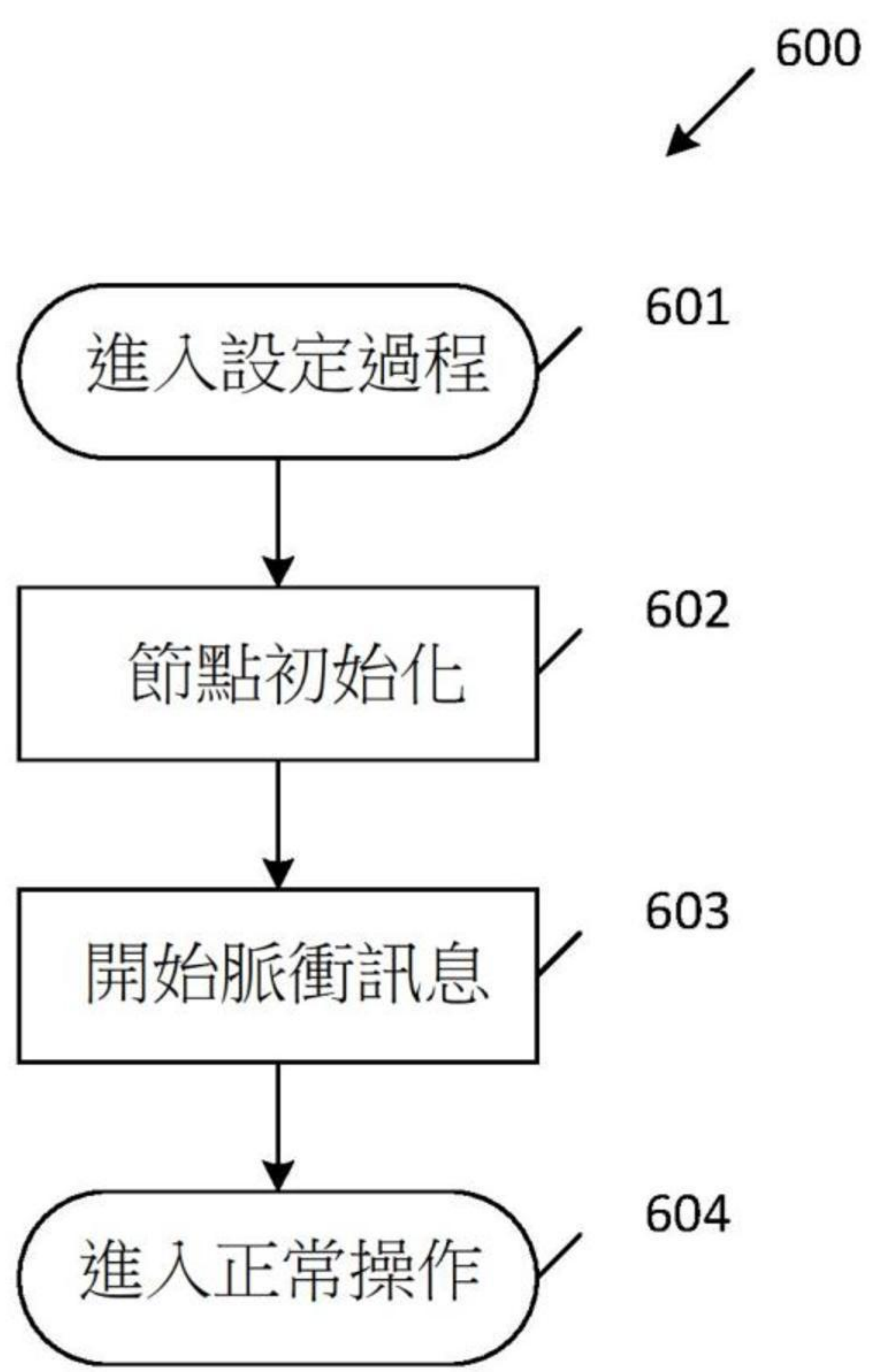
第3圖



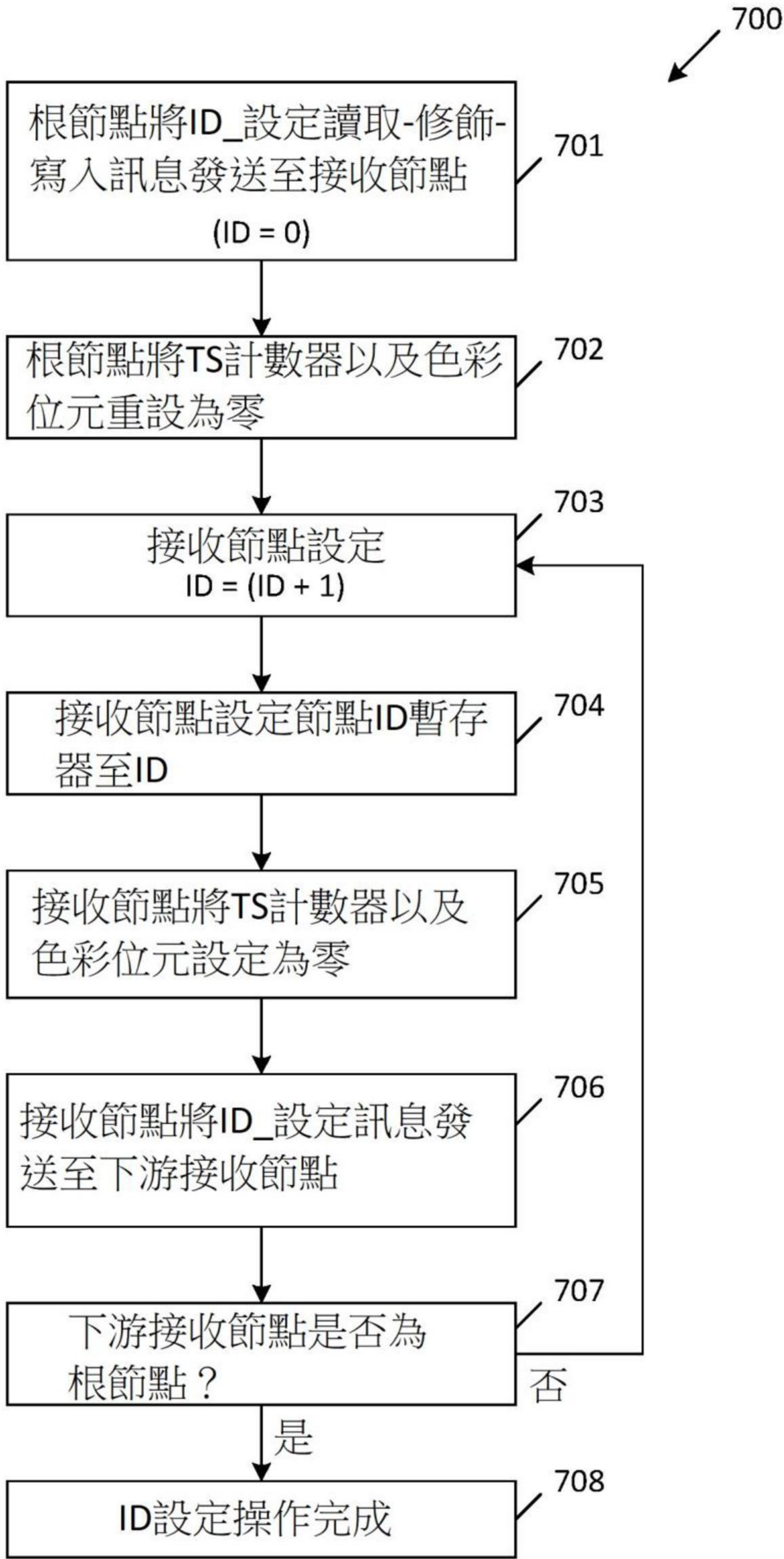
第 4 圖



第 5 圖



第 6 圖



第 7 圖

800
↓

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作	CPU_0將對快取列「A」的專用要求發送至節點N5		節點N2將排序P2P訊息發送至節點N4			
訊息匯流排	M*0_5(0)		M*2_4(0)			
訊息竊取程式	M*0_5(0)		M*2_4(0)			
TS計數器	0 → 1	0	0 → 1	0	0	0
色彩位元	1	1	1	1	0 → 1	0
脈衝位置					X	

第8A圖

801
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作						
訊息匯流排		M*0_5(0)		M*2_4(0)		
訊息竊取程式	M*0_5(0)	M*0_5(0)	M*2_4(0)	M*2_4(0)		
TS計數器	1	0 → 1	1	0 → 1	0	0
色彩位元	1	1	1	1	1	0 → 1
脈衝位置						X

第 8B 圖

802
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作				CPU_1將對快 取列「A」的 專用要求發送 至節點N5		
訊息匯 流排			M*0_5(0)	M*3_5(1)	M*2_4(0)	
訊息竊取 程式	M*0_5(0)	M*0_5(0)	M*2_4(0) M*0_5(0)	M*2_4(0) M*3_5(1)	M*2_4(0)	
TS計數器	1	1	1 → 2	1 → 2	0 → 1	0
色彩位元	1 → 0	1	1	1	1	1
脈衝位置	x					

第 8C 圖

803
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作						
訊息匯 流排				M*0_5(0)	M*3_5(1)	M*2_4(0)
訊息竊取 程式	M*0_5(0)	M*0_5(0)	M*2_4(0) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1)	M*2_4(0)
TS計數器	1	1	2	2→3	1→2	0→1
色彩位元	0	1→0	1	1	1	1
脈衝位置		X				

第8D圖

804
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作						
訊息匯 流排	M*2_4(0)				M*0_5(0)	M*3_5(1)
訊息竊取 程式	M*0_5(0) M*2_4(0)	M*0_5(0)	M*2_4(0) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1)
TS計數器	1 → 2	1	2	3	2 → 3	1 → 2
色彩位元	0	0	1 → 0	1	1	1
脈衝位置			X			

第 8E 圖

805
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作						
訊息匯 流排	M*3_5(1)	M*2_4(0)				M*0_5(0)
訊息竊取 程式	M*0_5(0) M*2_4(0) M*3_5(1)	M*0_5(0) M*2_4(0)	M*2_4(0) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)
TS計數器	2→3	1→2	2	3	3	2→3
色彩位元	0	0	0	1→0	1	1
脈衝位置				X		

第 8F 圖

806



	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作	M*0_5 (0) 循環回N0		M*2_4 (0) 循環回N2		N4解析訊息 竊取程式的 內容	
訊息匯 流排	M*0_5(0)	M*3_5(1)	M*2_4(0)			
訊息竊取 程式	M*0_5(0) M*2_4(0) M*3_5(1)	M*0_5(0) M*2_4(0) M*3_5(1)	M*2_4(0) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)
TS計數器	3	2→3	2	3	3	3
色彩位元	0	0	0	0	1→0	1
脈衝位置					X	

第8G圖

807
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作						N5解析訊息 竊取程式的 內容
訊息匯 流排			M*3_5(1)			
訊息竊取 程式	M*0_5(0) M*2_4(0) M*3_5(1)	M*0_5(0) M*2_4(0) M*3_5(1)	M*2_4(0) M*0_5(0) M*3_5(1)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)
TS計數器	3	3	2→3	3	3	3
色彩位元	0	0	0	0	0	1→0
脈衝位置						X

第 8H圖

808
↙

	N0 (CPU_0)	N1	N2	N3 (CPU_1)	N4	N5 (記憶體)
動作				M*3_5 (1) 循環回N3		
訊息匯 流排				M*3_5(1)		
訊息竊取 程式	M*0_5(0) M*2_4(0) M*3_5(1)	M*0_5(0) M*2_4(0) M*3_5(1)	M*2_4(0) M*0_5(0) M*3_5(1)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)	M*2_4(0) M*3_5(1) M*0_5(0)
TS計數器	3	3	3	3	3	3
色彩位元	0→1	0	0	0	0	0
脈衝位置	x					

第 8I 圖