



(19) 中華民國智慧財產局

(12) 發明說明書公開本

(11) 公開編號：TW 201606653 A

(43) 公開日：中華民國 105 (2016) 年 02 月 16 日

(21) 申請案號：104119095 (22) 申請日：中華民國 104 (2015) 年 06 月 12 日  
 (51) Int. Cl. : **G06F9/455 (2006.01)** **G06F17/27 (2006.01)**  
 (30) 優先權：2014/06/12 美國 62/011,057  
 (71) 申請人：國立交通大學 (中華民國) NATIONAL CHIAO TUNG UNIVERSITY (TW)  
 新竹市大學路 1001 號  
 (72) 發明人：李鎮宜 LEE, CHEN YI (TW)；張錫嘉 CHANG, HSI CHIA (TW)；許書餘 HSU, SHU YU (TW)；陳志龍 CHEN, CHIH LUNG (TW)；蔡長宏 TSAI, CHANG HUNG (TW)；王 永雄 WONG, WING-HUNG (US)；吳東祐 WU, TUNG YU (TW)；廖英秀 LIAO, YING SIOU (TW)；朱家慶 CHU, CHIA CHING (TW)；古方如 KU, FANG JU (TW)  
 (74) 代理人：祁明輝；葉明源  
 申請實體審查：有 申請專利範圍項數：10 項 圖式數：14 共 55 頁

## (54) 名稱

多維資料空間的貝氏循序切割系統及其計數引擎

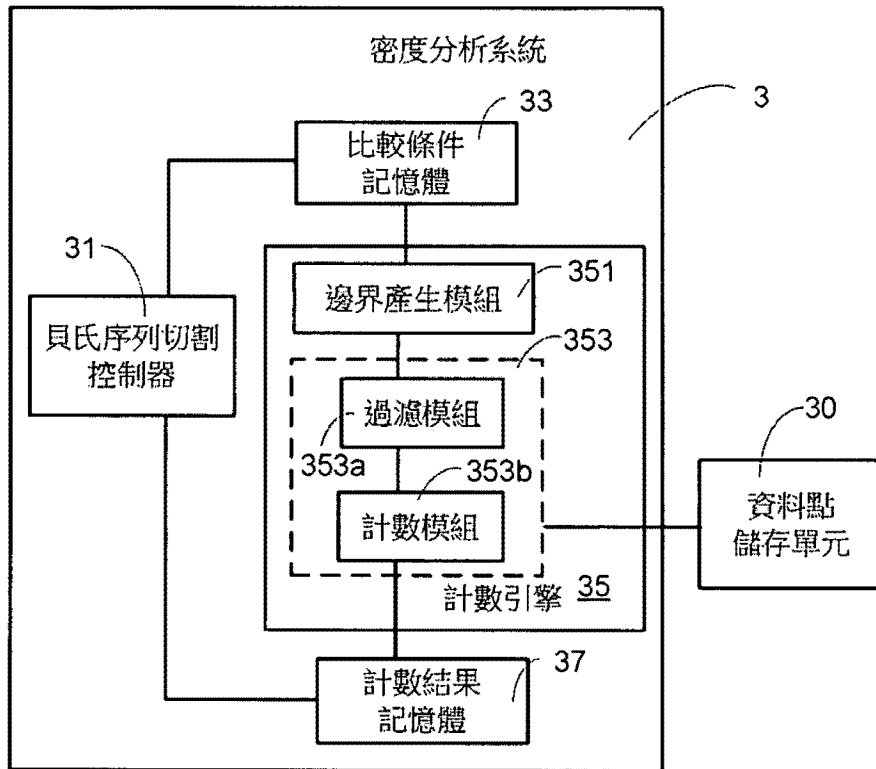
BAYESIAN SEQUENTIAL PARTITION SYSTEM IN MULTI-DIMENSIONAL DATA SPACE AND COUNTING ENGINE THEREOF

## (57) 摘要

本發明係為一計數引擎，應用於 D 維度資料空間的一貝氏循序切割系統中。計數引擎包括：一過濾模組，比較至少一受測資料點及與一子分區所對應的 D 個邊界資訊產生 D 個旗標組；以及一計數模組，連接至該過濾模組，根據 D 個旗標組判斷該至少一受測資料點是否位於該子分區而產生一結果信號，並根據該結果信號而選擇性遞增與該子分區對應之一累計值。

A counting engine for a Bayesian sequential partition system in a D-dimensional data space is provided. The counting engine includes a filtering module and a counting module. The filtering module is used for comparing at least one under-test data point with D boundary information corresponding to a sub-region, and consequently generating D flag sets. The counting module is connected with the filtering module. The counting module determines whether the at least one under-test data point lies in the sub-region, and consequently generates a result signal. A counting value corresponding to the sub-region is selectively accumulated by the counting module according to the result signal.

指定代表圖：



符號簡單說明：

3 . . . 密度分析系統

31 . . . 貝氏序列切割  
控制器

33 . . . 比較條件記  
憶體

35 . . . 計數引擎

37 . . . 計數結果記  
憶體

351 . . . 邊界產生模  
組

353a . . . 過濾模組

353b . . . 計數模組

353 . . . 過濾與計數  
模組

30 . . . 資料點儲存  
單元

第4圖

## 發明摘要

※ 申請案號：104119095

※ 申請日：104. 6. 12

※ IPC 分類：

G06F 9/455 (2006.01)

G06F 17/27 (2006.01)

## 【發明名稱】(中文/英文)

多維資料空間的貝氏循序切割系統及其計數引擎/  
BAYESIAN SEQUENTIAL PARTITION SYSTEM IN  
MULTI-DIMENSIONAL DATA SPACE AND COUNTING ENGINE  
THEREOF

## 【中文】

本發明係為一計數引擎，應用於 D 維度資料空間的一貝氏循序切割系統中。計數引擎包括：一過濾模組，比較至少一受測資料點及與一子分區所對應的 D 個邊界資訊產生 D 個旗標組；以及一計數模組，連接至該過濾模組，根據 D 個旗標組判斷該至少一受測資料點是否位於該子分區而產生一結果信號，並根據該結果信號而選擇性遞增與該子分區對應之一累計值。

## 【英文】

A counting engine for a Bayesian sequential partition system in a D-dimensional data space is provided. The counting engine includes a filtering module and a counting module. The filtering module is used for comparing at least one under-test data point with D boundary information corresponding to a sub-region, and consequently generating D flag sets. The counting module is connected with the filtering module. The counting module determines whether the at least one under-test data point lies in the sub-region, and consequently generates a result signal. A counting value corresponding to the sub-region is selectively accumulated by

the counting module according to the result signal.

**【代表圖】**

**【本案指定代表圖】：**第 4 圖。

**【本代表圖之符號簡單說明】：**

3：密度分析系統

31：貝氏序列切割控制器

33：比較條件記憶體

35：計數引擎

37：計數結果記憶體

351：邊界產生模組

353a：過濾模組

353b：計數模組

353：過濾與計數模組

30：資料點儲存單元

**【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：**

無

# 發明專利說明書

(本說明書格式、順序，請勿任意更動)

## 【發明名稱】(中文/英文)

多維資料空間的貝氏循序切割系統及其計數引擎/  
BAYESIAN SEQUENTIAL PARTITION SYSTEM IN  
MULTI-DIMENSIONAL DATA SPACE AND COUNTING ENGINE  
THEREOF

## 【技術領域】

【0001】 本發明是有關於一種密度分析系統，且特別是有關於一種應用於多維資料空間的貝氏循序切割系統及其計數引擎。

## 【先前技術】

【0002】 隨著科技的發展，無論是田野研究、技術研發、金融交易或網路資訊等領域均以爆炸的速度產生資料量。資料本身在沒有做任何處理以前，也許沒有價值，但經過適當的處理與分析後，才能進一步解讀並彰顯蘊含在資料背後的意涵與價值。當資料量規模達到千兆位元組(Petabyte)或百萬兆位元組(Eexabyte)時，便需要透過自動化的方式對巨量資料進行整理與分析。

【0003】 為此，習用技術使用在數十、數百甚至數千台伺服器上同時平行運行的軟體對巨量資料進行分析。但是，並非所有研究機構或開發人員均能負擔此種開發經費。再者，當資料集過度龐大時，分析巨量資料的效率仍然過慢。

【0004】 由此可知，如何加速巨量資料的分析過程，為一待解決的課題。

## 【發明內容】

【0005】 本發明的一方面係有關於一種一計數引擎，應用於

D 維度資料空間的一貝氏循序切割系統中，包括：一過濾模組，比較至少一受測資料點及與一子分區所對應的 D 個邊界資訊產生 D 個旗標組；以及一計數模組，連接至該過濾模組，根據 D 個旗標組判斷該至少一受測資料點是否位於該子分區而產生一結果信號，並根據該結果信號而選擇性遞增與該子分區對應之一累計值。

**【0006】** 本發明的另一方面係有關於一種多維資料空間的貝氏循序切割系統，連接至一資料點儲存單元，該資料點儲存單元儲存複數個資料點的各維度數值，該貝氏循序切割系統包括：一控制器，產生一分區所對應的一分區資訊；一比較條件記憶體，連接至該控制器，並暫存該分區資訊；一計數引擎，連接於該比較條件記憶體與該資料點儲存單元，該計數引擎根據一第一模擬分割將該分區切割為一第一子分區與一第二子分區，以及該計數引擎根據該分區資訊產生一過濾條件運用於過濾該些資料點，並計數該些資料點位於該第一子分區中的一第一數目；以及一計數結果記憶體，連接於該計數引擎與該控制器，用以暫存該第一數目，並傳遞至該控制器；其中，該控制器中記錄該些資料點中有一第二數目位於該分區中，並將該第二數目減去該第一數目而獲得一第三數目，使得該控制器決定該些資料點中有該第三數目位於該第二子分區中；以及該控制器根據該第一數目與該第三數目而獲得該第一模擬切割的一第一切割權重。

**【0007】** 為了對本發明之上述及其他方面有更佳的瞭解，下文特舉較佳實施例，並配合所附圖式，作詳細說明如下：

### **【圖式簡單說明】**

#### **【0008】**

第 1 圖係對資料空間進行貝氏循序切割演算法之流程圖。

第 2A 圖所繪示為 2 維資料空間進行貝氏循序切割演算法的切割示意圖。

第 2B 圖所繪示為 3 維資料空間進行貝氏循序切割演算法的切割示意圖。

第 2C 圖與第 2D 圖為 2 維資料空間並經過多次切割之示意圖。

第 3 圖所示為運用於 D 維的資料空間中的第 P 次切割流程示意圖。

第 4 圖係本發明之多維資料空間的貝氏循序切割系統。

第 5A 圖與第 5B 圖示為 3 維的原始分區組合及第 A 分區各種模擬切割示意圖。

第 6A 圖為過濾與計數模組的第一實施例。

第 6B 圖為比較器電路與旗標產生電路之運作原理。

第 6C 圖為切割拖曳器示意圖。

第 7A 圖為過濾與計數模組的第二實施例。

第 7B 圖為比較器電路與旗標產生電路之運作原理。

第 7C 圖為切割拖曳器陣列示意圖。

第 8A 圖為過濾與計數模組的第三實施例。

第 8B 圖為比較器電路與旗標產生電路之運作原理。

第 8C 圖為切割拖曳器陣列示意圖。

第 8D 圖為切割拖曳器陣列的詳細電路圖。

第 9 圖係本發明多維資料空間的貝氏循序切割系統的另一實施例的示意圖。

第 10 圖係本發明多維資料空間的貝氏循序切割系統的再一實施例之示意圖。

第 11A、11B、11C、11D 圖係本發明實施例對計數引擎進行設定的示意圖。

第 12 圖係本發明實施例之計數晶片提供多個計數引擎(CCE)之示意圖。

第 13 圖係本發明實施例以序列方式連接多個計數晶片的示意圖。

第 14 圖係本發明實施例以並聯方式連接多個計數晶片的示

意圖。

### 【實施方式】

多維資料空間的貝氏循序切割

【0009】 在多維度的資料空間(data space)中，每一筆資料可視為為資料空間中的一個資料點(data point)。此資料點的位置會依據資料點本身在每個維度的數值而決定。對巨量資料的統計而言，資料密度(data density)代表資料點在資料空間的集中程度。得出資料密度後，便能用於分析資料點分布情形所代表的關鍵資訊。

【0010】 貝氏循序切割演算法(Bayesian Sequential Partition Algorithm，簡稱 BSP 演算法)是一個資料取向的機率密度(probability density)函數估計的方法。貝氏循序切割演算法係以序列化二元分割(Sequential Binary Partitioning)的方式對資料空間進行有效率的切割(cut)，而經過切割後所產生的分區(region)，會根據資料點的分布情況而異，是一種功能十分強大的機器學習演算法。

【0011】 請參見第 1 圖，其係對資料空間進行貝氏循序切割演算法之流程圖。首先，正規化原始資料(raw data)，並獲得資料空間中的資料點數目(步驟 S11)。接著，以資料空間作為最初的原始分區組合(initial region set)。其次，對原始分區組合中的每個分區進行各維度的模擬切割(simulated cut)，並計算位在模擬切割所對應生成之子分區(sub region)內的資料點數目(步驟 S13)。根據位在每種模擬切割所對應生成之二個子分區內的資料點數目，計算出與每種模擬切割對應的權重(cutting weights)(步驟 S15)。根據所有模擬切割的權重決定一選定切割(selected cut)後，根據選定切割而對原始分區組合中的一個分區進行切割，進而產生更新的原始分區組合(步驟 S17)。接著，判斷切割的停止條件是否



成立。於停止條件不成立時，回到步驟 S13；反之，於停止條件成立時，結束流程。

【0012】 其中，步驟 S13 至步驟 S18 的流程係為貝氏循序切割演算法中的循序重要性取樣流程(sequential important sampling，簡稱 SIS)。以下以 2 維資料空間以及 3 維資料空間為例來說明貝氏循序切割演算法的流程。

【0013】 請參照第 2A 圖，其所繪示為 2 維資料空間進行貝氏循序切割演算法的切割示意圖。假設每一個資料點具有兩個欄位的資料，其中每一個欄位對應於一個資料維度(dimension)。因此，資料空間可表示為一個二維的平面(資料維度  $d=2$ )。為便於說明，此處將 X 軸方向定義為第一資料維度，以及將 Y 軸方向定義為第二資料維度。

【0014】 再者，巨量的原始資料經過正規化後，形成散布在平面(二維)資料空間上的多個資料點(未繪示)。其中，每一個資料維度的範圍均介於 0~1 之間。換言之，所有原始資料經過正規化後，根據所有資料點對應的資料維度之數值，即可得知所有資料點在資料空間中的位置並計算出所有資料點數目。

【0015】 如(1)所示，當原始分區組合為整個資料空間時，該資料空間中的資料點數目為已知。再者，該資料空間僅有一個分區，所以共有二個維度的模擬切割。其中，第一模擬切割(simulated cut)係在第一資料維度將此分區切割成為左右二個子分區，並且於計算切割後左右二子分區內的資料點數目後，即可據以計算出第一模擬切割的權重。第二模擬切割係在第二資料維度將此分區切割成為上下二個子分區，並且於計算切割後上下二子分區內的資料點數目後，即可據以計算出第二模擬切割的權重。再者，根據上述二個模擬切割的權重，即可進一步決定二個模擬切割其中之一為選定切割。假設第一模擬切割為選定切割，則此分區即在第一資料維度進行第一次切割，而更新的原始分區組合即如(2)所示具有左右二個分區。

【0016】 如(2)所示，當原始分區組合具有左右二個分區時，該左右二個分區中的資料點數目為已知。再者，每個分區各自針對二個維度進行模擬切割，因此共有  $4(2 \times 2)$  個模擬切割。其中，針對左分區進行第一模擬切割與第二模擬切割，以及針對右分區進行第三模擬切割與第四模擬切割。

【0017】 第一模擬切割係對(2)的左分區的第一資料維度進行，將(2)的左分區再切割成為左右二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第一模擬切割的權重。第二模擬切割係對(2)的左分區的第二資料維度進行，將(2)的左分區再切割成為上下二個子分區。並且，於計算切割後上下二子分區內的資料點數目後，即可據以計算出第二模擬切割的權重。

【0018】 第三模擬切割係對(2)的右分區的第一資料維度進行，將(2)的右分區再切割成為左右二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第三模擬切割的權重。第四模擬切割係對(2)的右分區的第二資料維度進行，將(2)的右分區再切割成為上下二個子分區，並且於計算切割後左右二子分區內的資料點數目後，即可據以計算出第四模擬切割的權重。

【0019】 再者，假設根據上述四個模擬切割的權重，進一步決定(2)的第二模擬切割為選定切割。因此，資料空間即被進行第二次切割，而更新的原始分區組合即如(3)所示具有左上、左下、右三個分區。

【0020】 如(3)所示，當原始分區組合具有左上、左下、右三個分區時，左上、左下、右三個分區中的資料點數目為已知。再者，每個分區有二個維度的模擬切割，因此共有  $6(3 \times 2)$  個模擬切割。其中，針對左上分區進行第一模擬切割與第二模擬切割；針對左下分區進行第三模擬切割與第四模擬切割；以及針對右分區進行第五模擬切割與第六模擬切割。

【0021】 第一模擬切割係對(3)的左上分區的第一資料維度進行，將(3)的左上分區再切割成為左右二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第一模擬切割的權重。第二模擬切割係對(3)的左上分區的第二資料維度進行，將(3)的左上分區再切割成為上下二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第二模擬切割的權重。

【0022】 第三模擬切割係對(3)的左下分區的第一資料維度進行，將(3)的左下分區再切割成為左右二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第三模擬切割的權重。第四模擬切割係對(3)的左下分區的第二資料維度進行，將(3)的左下分區再切割成為上下二子個分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第四模擬切割的權重。

【0023】 第五模擬切割係對(3)的右分區的第一資料維度進行，將(3)的右分區再切割成為左右二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第五模擬切割的權重。第六模擬切割係對(3)的右分區的第二資料維度進行，將(3)的右分區再切割成為上下二個子分區。並且，於計算切割後二子分區內的資料點數目後，即可據以計算出第六模擬切割的權重。

【0024】 其後，根據上述六個模擬切割的權重，進一步決定其中一個模擬切割為選定切割。此時，資料空間即被進行第三次切割。依此類推，貝氏循序切割演算法會對資料空間進行反覆的切割。因此，資料空間經過第(N-1)次切割之後，資料空間已經被切割成 N 個分區，並作為更新的原始分區組合。

【0025】 如(N)所示，當原始分區組合具 N 個分區時，N 個分區中的資料點數目為已知。再者，每個分區有二個維度的模擬切割，因此共有(N×2)個模擬切割。利用相同的方式，於(N×2)個模

擬切割中決定一選定切割後，即可將資料空間進行第  $N$  次切割，使得資料空間被切割成  $(N+1)$  個分區，並作為更新的原始分區組合。再者，當切割的停止條件成立時，即不再繼續進行切割。

**【0026】** 請參照第 2B 圖，其所繪示為 3 維資料空間進行貝氏循序切割演算法的切割示意圖。假設每一個資料點具有三個欄位的資料，其中每一個欄位對應於一個資料維度(dimension)。因此，資料空間可表示為一個三維的空間(資料維度  $d=3$ )。為便於說明，此處將  $X$  軸方向定義為第一資料維度；將  $Y$  軸方向定義為第二資料維度；以及將  $Z$  軸方向定義為第三資料維度。再者，巨量的原始資料經過正規化後，形成散布在三維資料空間為上的多個資料點(未繪示)。其中，每一個資料維度的範圍均介於  $0\sim 1$  之間。

**【0027】** 相同的原理，如(1)所示，當原始分區組合為整個資料空間時，該分區中的資料點數目為已知。再者，此分區有三個維度的模擬切割，因此資料空間中共有三個模擬切割。而經過計算後，根據三個模擬切割的權重，進一步決定第一模擬切割為選定切割。此時，資料空間即被進行第一次切割，而更新的原始分區組合即如(2)所示。

**【0028】** 如(2)所示，當原始分區組合具有二個分區時，該二個分區中的資料點數目為已知。再者，每個分區有三個維度的模擬切割，因此資料空間共有六個( $2\times 3$ )模擬切割。而經過計算後，根據六個模擬切割的權重，進一步決定第一模擬切割為選定切割。此時，資料空間即被進行第二次切割，而更新的原始分區組合即如(3)所示。

**【0029】** 依此類推，貝氏循序切割演算法會對資料空間進行反覆的切割。因此，資料空間經過第  $(M-1)$  次切割之後，資料空間已經被切割成  $M$  個分區，並作為更新的原始分區組合。

**【0030】** 如(M)所示，當原始分區組合具  $M$  個分區時， $M$  個分區中的資料點數目為已知。再者，每個分區有三個維度的模擬

切割，因此資料空間中共有 $(M \times 3)$ 個模擬切割。利用相同的方式，於 $(M \times 3)$ 個模擬切割中決定一選定切割後，即可將資料空間進行第  $M$  次切割，使得資料空間被切割成 $(M+1)$ 個分區，並作為更新的原始分區組合。再者，當切割的停止條件成立時，即不再繼續進行切割。

**【0031】** 由以上的說明可知，貝氏循序切割的過程會預先對原始分區組合中的各分區進行各維度的模擬切割。接著，根據模擬切割所對應生成的二個子分區，進行資料點數目的計算並決定模擬切割的權重。而獲得所有的模擬切割的權重之後，即可決定一選定切割。基本上，模擬切割的權重越大者，被決定為選定切割的機率越高。

**【0032】** 以下以實際的 2 維資料空間 200 為例來進行說明。如第 2C 圖所示，假設巨量的原始資料經過正規化後，形成散布在第 2C 圖的平面資料空間 200 中的左上方以及右下方的資料點群，其中每一個黑點代表一個資料點。其中，每一個資料維度的範圍均介於 0~1 間。

**【0033】** 如第 2D 圖所示，每一條線條代表一個選定切割。由第 2D 圖可知，貝氏循序切割會對資料空間 200 進行反覆的切割，以產生許多分區。如第 2D 圖所示，資料點越集中的分區，其分區中的模擬切割會有較高的機率被決定為選定切割。因此，第 2D 圖中線條越密集的地方代表資料點越集中分布的地方，代表資料密度越高。換言之，貝氏循序切割可對資料空間 200 進行有效率的切割，用以獲得資料密度並運用於分析資料點分布情形所代表的關鍵資訊。

**【0034】** 由以上的說明可知，隨著切割次數的增加，原始分區組合中的分區數目也會增加。同時，模擬切割的數目也會增加，而計算子分區中資料點數目也會成為非常大的負擔。

**【0035】** 如第 3 圖所示，其為運用於  $D$  維的資料空間中的第  $P$  次切割流程示意圖。當原始分區組合具有  $P$  個分區時， $P$  個分

區中的資料點數目為已知。再者，每個分區有  $D$  個維度的模擬切割，因此資料空間中共有  $(P \times D)$  個模擬切割  $WP_{11} \sim WP_{PD}$ 。再者，於  $(P \times D)$  個模擬切割  $WP_{11} \sim WP_{PD}$  中決定一選定切割後，即可將資料空間進行第  $P$  次切割，使得資料空間被切割成  $(P+1)$  個分區。

【0036】 很明顯地，為了要計算  $(P \times D)$  個模擬切割的權重，共需要獲得  $(2 \times P \times D)$  個子分區中的資料點數目。計算子分區中資料點數目的次數會相關於切割次數  $P$  以及資料維度  $D$ 。再者，當切割次數  $P$  增加時，計算子分區中的資料點數目的次數也會隨之增加。因此，如何有效率地計算子分區中的資料點數目，即可降低多維資料空間的貝氏循序切割系統之負擔。

#### 多維資料空間的貝氏循序切割系統

【0037】 請參見第 4 圖，其係本發明之多維資料空間的貝氏循序切割系統，其為一種多維資料的密度分析系統。此密度分析系統 3 包含貝氏序列切割控制器 31、比較條件記憶體 33、計數引擎(counting engine)35 與計數結果記憶體 37。其中，計數引擎 35 電連接至計數結果記憶體 37。計數引擎 35 包含：邊界產生模組(boundary generating module)351、至少一過濾模組 353a 以及至少一計數模組 353b。其中，過濾模組(filtering module)353a 以及計數模組(counting module)353b 可共同稱為過濾與計數模組 353。

【0038】 比較條件記憶體 33 電連接於貝氏序列切割控制器 31 與邊界產生模組 351。比較條件記憶體 33 用於將貝氏序列切割控制器 31 產生之分區資訊(region information)傳送至邊界產生模組 351。邊界產生模組 351 根據分區資訊產生邊界資訊(boundary information)，提供至過濾模組 353a 用以判斷資料點是否位於子分區內。過濾模組 353a 將判斷的結果輸出至計數模組 353b，由計數模組 353b 計算位於子分區內的資料點個數(即，子

分區計數結果)。

【0039】 再者，計數結果記憶體 37 電連接於貝氏序列切割控制器 31 與計數模組 353b。計數結果記憶體 37 用於儲存計數模組 353b 產生的子分區計數結果，並傳送至貝氏序列切割控制器 31。

【0040】 再者，資料點儲存單元 30 中儲存所有資料點的各維度的數值，而過濾與計數模組 353 可讀取在資料點儲存單元 30 中，所有資料點的各維度數值。

【0041】 請參照第 5A 圖所繪示為 3 維的原始分區組合示意圖。此原始分區組合中共有 A、B、C 三個分區，且每個分區中資料點數目為已知。再者，利用 X、Y、Z 軸的三維座標，每個分區皆可利用八個座標點來表示該分區。

【0042】 再者，本發明更定義新的分區表示法作為分區資訊，以更簡化的方式來表示原始分區組合中的三個分區。如第 5A 圖所示，第 A 分區在第一資料維度(X 軸方向)係由 0 位置開始延伸至 1；第 A 分區在第二資料維度(Y 軸方向)係由 0 位置開始延伸至 1；第 A 分區在第三資料維度(Z 軸方向)係由 0.5 位置開始延伸至 1。

【0043】 因此，第 A 分區以 0 作為第一資料維度的起點，並以 1 作為第一資料維度的總長度。第 A 分區以 0 作為第二資料維度的起點，並以 1 作為第二資料維度的總長度；第 A 分區以 0.5 作為第三資料維度的起點，並以 0.5 作為第三資料維度的總長度。因此，第 A 分區的分區資訊可表示為  $(R1=0, L1=0.5)$ 、 $(R2=0, L2=0.5)$ 、 $(R3=0.5, L3=0.25)$ ；其中 R1、R2、R3 為各維度的起始點，L1、L2、L3 為各資料維度的半長度。

【0044】 同理，第 B 分區的分區資訊可表示為  $(R1=0, L1=0.25)$ 、 $(R2=0, L2=0.5)$ 、 $(R3=0, L3=0.25)$ ；第 C 分區的分區資訊可表示為  $(R1=0.5, L1=0.25)$ 、 $(R2=0, L2=0.5)$ 、 $(R3=0, L3=0.25)$ 。

【0045】 接著，以下以第 5B 圖來介紹對第 A 分區進行的各種模擬切割，及其子分區的資料點數目之計算。當然，第 5B 圖

之方式也可運用於原始分區組合中的第 B 分區與第 C 分區。

【0046】 貝氏循序切割演算法係以序列化二元分割的方式對資料空間進行切割。因此，如第 5B 圖所示，第 A 分區的模擬切割 1 為  $X=0.5$  的平面，亦即 X 軸上  $(R1+L1)$  的位置，並且模擬切割 1 將第 A 分區分成子分區 a1 與子分區 a2。同理，第 A 分區的模擬切割 2 為  $Y=0.5$  的平面，亦即 Y 軸上  $(R2+L2)$  的位置，並且模擬切割 2 將第 A 分區分成子分區 b1 與子分區 b2。第 A 分區的模擬切割 3 為  $Z=0.75$  的平面，亦即 Z 軸上  $(R3+L3)$  的位置，並且模擬切割 3 將第 A 分區分成子分區 c1 與子分區 c2。

【0047】 根據模擬切割 1，可進一步決定子分區 a1 的過濾條件(filtering condition)，用以計算子分區 a1 中的資料點數目。而由於第 A 分區中的資料點數目為已知，所以將第 A 分區中的資料點數目減去子分區 a1 中的資料點數目即為子分區 a2 中的資料點數目。利用相同的方式，模擬切割 2 可進一步決定子分區 b1 的過濾條件，用以獲得子分區 b1 與子分區 b2 中的資料點數目；而根據模擬切割 3 可進一步決定子分區 c1 的過濾條件，用以獲得子分區 c1 與子分區 c2 中的資料點數目。

【0048】 再者，根據模擬切割 1 所決定的子分區 a1 的過濾條件為：第 1 資料維度的過濾區間(filtering range)  $R1\sim(R1+L1)$ 、第 2 資料維度的過濾區間  $R2\sim(R2+2\times L2)$ 、第 3 資料維度的過濾區間  $R3\sim(R3+2\times L3)$ 。換言之，根據模擬切割 1 的過濾條件，當一受測資料點的第 1 維度數值(value)位在第 1 資料維度的過濾區間  $(R1\sim(R1+L1))$ ；該受測資料點的第 2 維度數值在第 2 資料維度的過濾區間  $(R2\sim(R2+2\times L2))$ ；且該受測資料點的第 3 維度數值在第 3 資料維度的過濾區間  $(R3\sim(R3+2\times L3))$  時，即可確定該受測資料點在子分區 a1 內。反之，即確認該受測資料點並未在子分區 a1 內。

【0049】 再者，根據模擬切割 2 所決定的子分區 b1 的過濾條件為：第 1 資料維度的過濾區間  $R1\sim(R1+2\times L1)$ 、第 2 資料維度的



過濾區間  $R2 \sim (R2+L2)$ 、第 3 資料維度的過濾區間  $R3 \sim (R3+2 \times L3)$ 。同理，根據模擬切割 2 的過濾條件，當一受測資料點的第 1 維度數值位在第 1 資料維度的過濾區間 ( $R1 \sim (R1+2 \times L1)$ )；該受測資料點的第 2 維度數值位在第 2 資料維度的過濾區間 ( $R2 \sim (R2+L2)$ )；且該受測資料點的第 3 維度數值位在第 3 資料維度的過濾區間 ( $R3 \sim (R3+2 \times L3)$ ) 時，即可確定該受測資料點在子分區  $b1$  內。反之，即確認該受測資料點並未在子分區  $b1$  內。

**【0050】** 再者，根據模擬切割 3 所決定的子分區  $c1$  的過濾條件為： $R1 \sim (R1+2 \times L1)$  的第 1 資料維度的過濾區間、 $R2 \sim (R2+2 \times L2)$  的第 2 資料維度的過濾區間、 $R3 \sim (R3+L3)$  的第 3 資料維度的過濾區間。換言之，根據模擬切割 3 的過濾條件，當一受測資料點的第 1 維度數值位在第 1 資料維度的過濾區間 ( $R1 \sim (R1+2 \times L1)$ )；該受測資料點的第 2 維度數值位在第 2 資料維度的過濾區間 ( $R2 \sim (R2+2 \times L2)$ )；且該受測資料點的第 3 維度數值位在第 3 資料維度的過濾區間 ( $R3 \sim (R3+L3)$ ) 時，即可確定該受測資料點在子分區  $c1$  內。反之，即確認該受測資料點並未在子分區  $c1$  內。

**【0051】** 基本上，第 4 圖的密度分析系統 3 中，貝氏循序切割控制器 31 提供一個分區的分區資訊 (region information) 至比較條件記憶體 33。而邊界產生模組 351 可根據分區資訊產生每一維度的邊界資訊 (boundary information) 至過濾與計數模組 353。過濾與計數模組 353 即根據每一維度的邊界資訊，來建立每一模擬切割所對應的過濾條件，用以計算子分區的資料點數目。而過濾與計數模組 353 計算子分區的資料點數後會儲存於計數結果記憶體 37 並提供至貝氏循序切割控制器 31。

**【0052】** 根據以上的說明，本發明提出第 6A 圖之過濾與計數模組的第一實施例，可運用於第 4 圖的密度分析系統 3。過濾與計數模組 453 包含過濾模組 453a 與計數模組 453b。其中，過濾模組 453a 至少包括：一比較器電路 41 與一旗標產生電路 42。計數模組 453b 至少包括：一切割拖曳器 (cutting trailer) 45、與一

累加器(accumulator)46。其中，SC\_dim 為模擬切割的資料維度；Data\_dim 為正在處理的資料維度，R 與 L 為一個分區中，單一維度的起始點與半長度，因此  $R$ 、 $(R+L)$ 、 $(R+2L)$  即為單一維度的邊界資訊。

**【0053】** 如第 6B 圖所示，其為比較器電路與旗標產生電路之運作原理。比較器電路 41 中至少包括三個比較器，用於比較資料點在與邊界資訊之間的關係。其中，第一比較器判斷資料點是否大於或等於  $R$ ，當資料點大於等於  $R$  時，第一比較結果 Cmp\_a="1"；當資料點小於  $R$  時，第一比較結果 Cmp\_a="0"。第二比較器判斷資料點是否小於  $(R+L)$ ，當資料點小於  $(R+L)$  時，第二比較結果 Cmp\_b="1"；當資料點大於或等於  $(R+L)$  時，第二比較結果 Cmp\_b="0"。第三比較器判斷資料點是否小於  $(R+2L)$ ，當資料點小於  $(R+2L)$  時，第三比較結果 Cmp\_c="1"；當資料點大於或等於  $(R+2L)$  時，第三比較結果 Cmp\_c="0"。

**【0054】** 再者，旗標產生電路 42 根據比較器電路 41 產生的三個比較結果來產生二個旗標。當資料點位於  $R \sim (R+L)$  的區間時，Cmp\_a 與 Cmp\_b 皆為"1"且 Cmp\_c 為"0"，此時設定第一旗標 Flag\_a="1"。當資料點位於  $R \sim (R+2L)$  的區間時，Cmp\_a 與 Cmp\_c 皆為"1"且 Cmp\_b 為"0"，此時設定第二旗標 Flag\_b="1"。

**【0055】** 如第 6C 圖所示，其為切割拖曳器 45 示意圖。切割拖曳器 45 可建立各維度的過濾區間，並以旗標信號作為判斷的結果。切割拖曳器 45 接收旗標產生電路 42 的旗標信號，根據 SC\_dim 與 Data\_dim 選擇一旗標信號作為判斷的結果。當模擬切割的資料維度 SC\_dim 與正在處理的資料維度 Data\_dim 相同時，第一多工器 mux1 即以第一旗標 Flag\_a 作為區間信號 Pt；當模擬切割的資料維度 SC\_dim 與正在處理的資料維度 Data\_dim 不相同時，第一多工器 mux1 區即以第二旗標 Flag\_b 作為區間信號 Pt。

**【0056】** 再者，暫存器 Reg 儲存的資料經由第二多工器 mux2 輸出為前一級區間信號 Pt-1，而區間信號 Pt 與前一級區間信號

Pt-1 輸入及閘(and gate)而產生的結果再次儲存於暫存器 Reg。再者，由第 6C 圖可知，當正在處理的資料維度 Data\_dim 為第 1 維度時，第二多工器 mux2 會將“1”作為前一級區間信號 Pt-1。

【0057】 根據本發明之實施例，切割拖曳器 45 係根據模擬切割的資料維度 SC\_dim 與正在處理的資料維度 Data\_dim，產生區間信號 Pt 後，將區間信號 Pt 前一級區間信號 Pt-1 經過 AND 運算後的結果儲存在儲存於暫存器 Reg 中。而暫存器 Reg 根據所有資料維度的區間信號，產生一檢查結果信號 Check。當結果信號 Check=“1”時，代表資料點位在模擬切割所生成的一個子區間內；反之，結果信號 Check=“0”時，代表資料點沒有位在模擬切割所生成的一個子區間。以下第 5B 圖的三維分區為例來說明過濾與計數模組 453 的實際動作原理。

【0058】 當過濾與計數模組 453 計算 a1 子分區中的資料點數目時，模擬切割 1 為第 1 維度(SC\_dim=“1”)。首先，令正在處理的資料維度 Data\_dim=“1”，且  $R=R1$ ， $L=L1$ ，而第一多工器 mux1 輸出第一旗標 Flag\_a 作為區間信號 Pt。因此，於<步驟 A1>中，過濾模組 453a 先判斷第一資料點的第 1 維度數值是否位於第 1 資料維度的過濾區間  $R1\sim(R1+L1)$ 。假設第一資料點的第 1 維度數值位於過濾區間  $R1\sim(R1+L1)$ ，則切割拖曳器 45 以第一旗標 Flag\_a=“1”作為區間信號 Pt，並儲存於暫存器 Reg。區間信號 Pt 相當於第一資料點的第 1 維度數值是否位於過濾區間  $R1\sim(R1+L1)$  的判斷結果。

【0059】 接著，令正在處理的資料維度 Data\_dim=“2”，且  $R=R2$ ， $L=L2$ ，而第一多工器 mux1 輸出第二旗標 Flag\_b 作為區間信號 Pt。因此，於<步驟 A2>，過濾模組 453a 判斷第一資料點的第 2 維度數值是否位於第 2 資料維度的過濾區間  $R2\sim(R2+2L2)$ 。假設第一資料點的第 2 維度數值位於過濾區間  $R2\sim(R2+2L2)$ ，則切割拖曳器 45 以第二旗標 Flag\_b=“1”作為區間信號 Pt。因此，區間信號 Pt 與前一級區間信號 Pt-1 經過及閘後儲存於暫存器

Reg。

【0060】 其中，前一級區間信號  $P_{t-1}$  相當於第一資料點的第 1 維度數值是否位於過濾區間  $R_1 \sim (R_1 + 2L_1)$  的判斷結果；且區間信號  $P_t$  相當於第一資料點的第 2 維度數值是否位於過濾區間  $R_2 \sim (R_2 + L_2)$  的判斷結果。因此，區間信號  $P_t$  與前一級區間信號  $P_{t-1}$  經過及閘後的結果，就相當於第一資料點的第 1 維度數值與第 2 維度數值是否皆位於過濾區間  $R_1 \sim (R_1 + L_1)$ 、 $R_2 \sim (R_2 + 2L_2)$  的判斷結果。

【0061】 接著，令正在處理的資料維度  $Data\_dim = "3"$ ，且  $R = R_3$ ， $L = L_3$ ，而第一多工器  $mux1$  輸出第二旗標  $Flag\_b$  作為區間信號  $P_t$ 。因此，於 <步驟 A3>，過濾模組 453a 判斷第一資料點的第 3 維度數值是否位於第 3 資料維度的過濾區間  $R_3 \sim (R_3 + 2L_3)$ 。假設第一資料點的第 3 維度數值位於過濾區間  $R_3 \sim (R_3 + 2L_3)$ ，則切割拖曳器 45 以第二旗標  $Flag\_b = "1"$  作為區間信號  $P_t$ 。因此，區間信號  $P_t$  與前一級區間信號  $P_{t-1}$  經過及閘後儲存於暫存器 Reg。

【0062】 其中，前一級區間信號  $P_{t-1}$  相當於第一資料點的第 1 維度數值與第 2 維度數值是否皆位於過濾區間的判斷結果；且區間信號  $P_t$  相當於第一資料點的第 3 維度數值是否位於過濾區間  $R_3 \sim (R_3 + 2L_3)$  的判斷結果。因此，區間信號  $P_t$  與前一級區間信號  $P_{t-1}$  經過及閘後的結果，就相當於第一資料點的第 1 維度數值、第 2 維度數值與第 3 維度數值是否皆位於過濾區間  $R_1 \sim (R_1 + L_1)$ 、 $R_2 \sim (R_2 + 2L_2)$ 、 $R_3 \sim (R_3 + 2L_3)$  的判斷結果。

【0063】 很明顯地，上述 <步驟 A1>~<步驟 A3> 中，係根據第 1 維度的模擬切割 1 ( $SC\_dim = "1"$ ) 所建立的過濾條件。而根據正在處理的資料維度  $Data\_dim$  的變化，依序產生三個過濾區間。再者，利用三個過濾區間進行判斷所獲得的三個區間信號  $P_t$  必需都要為 "1" 時，才會使最後儲存在暫存器 Reg 的區間信號  $P_t$  為 "1"。此時可確定第一資料點位在  $a1$  子分區中，並使得結果信號

Check="1"，並累計至累加器 46 中，使得累加器 46 中的累計值遞增 1。反之，假設最後儲存在暫存器 Reg 的區間信號 Pt 為"0"時，即確定第一資料點沒有位在 a1 子分區中，使得結果信號 Check="0"，此時將維持累加器 46 的數值。

【0064】 假設資料空間中，資料點數目共有 M\_data 筆，則上述<步驟 A1>至<步驟 A3>必須進行 M\_data 次之後，累加器 46 中的數值即為位於 a1 子分區中資料點數目。

【0065】 再者，當過濾與計數模組 453 計算 b1 子分區中的資料點數目時，模擬切割 2 為第 2 維度(SC\_dim="2")。首先，令正在處理的資料維度 Data\_dim="1"，且 R=R1，L=L1，而第一多工器 mux1 輸出第二旗標 Flag\_b 作為區間信號 Pt。因此，於<步驟 B1>中，過濾模組 453a 先判斷第一資料點的第 1 維度數值是否位於第 1 資料維度的過濾區間  $R1 \sim (R1+2L1)$ 。假設第一資料點的第 1 維度數值位於過濾區間  $R1 \sim (R1+2L1)$ ，則切割拖曳器 45 以第二旗標 Flag\_b="1"作為區間信號 Pt，並儲存於暫存器 Reg。區間信號 Pt 相當於第一資料點的第 1 維度數值是否位於過濾區間  $R1 \sim (R1+2L1)$ 的判斷結果。

【0066】 接著，令正在處理的資料維度 Data\_dim="2"，且 R=R2，L=L2，而第一多工器 mux1 輸出第一旗標 Flag\_a 作為區間信號 Pt。因此，於<步驟 B2>，過濾模組 453a 判斷第一資料點的第 2 維度數值是否位於第 2 資料維度的過濾區間  $R2 \sim (R2+L2)$ 。假設第一資料點的第 2 維度數值位於過濾區間  $R2 \sim (R2+L2)$ ，則切割拖曳器 45 以第一旗標 Flag\_a="1"作為區間信號 Pt。因此，區間信號 Pt 與前一級區間信號 Pt-1 經過及閘後儲存於暫存器 Reg。

【0067】 其中，前一級區間信號 Pt-1 相當於第一資料點的第 1 維度數值是否位於過濾區間  $R1 \sim (R1+L1)$ 的判斷結果；且區間信號 Pt 相當於第一資料點的第 2 維度數值是否位於過濾區間  $R2 \sim (R2+2L2)$ 的判斷結果。因此，區間信號 Pt 與前一級區間信號 Pt-1 經過及閘後的結果，就相當於第一資料點的第 1 維度數值與

第 2 維度數值是否皆位於過濾區間  $R1 \sim (R1+2L1)$ 、 $R2 \sim (R2+L2)$  的判斷結果。

【0068】 接著，令正在處理的資料維度  $Data\_dim="3"$ ，且  $R=R3$ ， $L=L3$ ，而第一多工器  $mux1$  輸出第二旗標  $Flag\_b$  作為區間信號  $Pt$ 。因此，於<步驟 B3>，過濾模組 453a 判斷第一資料點的第 3 維度數值是否位於第 3 資料維度的過濾區間  $R3 \sim (R3+2L3)$ 。假設第一資料點的第 3 維度數值位於過濾區間  $R3 \sim (R3+2L3)$ ，則切割拖曳器 45 以第二旗標  $Flag\_b="1"$  作為區間信號  $Pt$ 。因此，區間信號  $Pt$  與前一級區間信號  $Pt-1$  經過及閘後儲存於暫存器  $Reg$ 。

【0069】 其中，前一級區間信號  $Pt-1$  相當於第一資料點的第 1 維度數值與第 2 維度數值是否接位於過濾區間的判斷結果；且區間信號  $Pt$  相當於第一資料點的第 3 維度數值是否位於過濾區間  $R3 \sim (R3+2L3)$  的判斷結果。因此，區間信號  $Pt$  與前一級區間信號  $Pt-1$  經過及閘後的結果，就相當於第一資料點的第 1 維度數值、第 2 維度數值與第 3 維度數值是否皆位於過濾區間  $R1 \sim (R1+2L1)$ 、 $R2 \sim (R2+L2)$ 、 $R3 \sim (R3+2L3)$  的判斷結果。

【0070】 很明顯地，上述<步驟 B1>~<步驟 B3>中，係根據第 2 維度的模擬切割 2( $SC\_dim="2"$ )所建立的過濾條件。而根據正在處理的資料維度  $Data\_dim$  的變化，依序產生三個過濾區間。再者，利用三個過濾區間進行判斷所獲得的三個區間信號  $Pt$  必需都要為“1”時，才會使最後儲存在暫存器  $Reg$  的區間信號  $Pt$  為“1”。此時可確定第一資料點位在  $b1$  子分區中，並使得結果信號  $Check="1"$ ，並累計至累加器 46 中，使得累加器 46 中的累計值遞增 1。反之，假設最後儲存在暫存器  $Reg$  的區間信號  $Pt$  為“0”時，即確定第一資料點沒有位在  $b1$  子分區中，使得結果信號  $Check="0"$ ，而無法改變累加器 46 的數值。

【0071】 假設資料空間中，資料點數目共有  $M\_data$  筆，則上述<步驟 B1>至<步驟 B3>必須進行  $M\_data$  次之後，累加器 46 中

的數值即為  $b1$  子分區中資料點數目。

【0072】 再者，當過濾與計數模組 453 計算  $c1$  子分區中的資料點數目時，模擬切割 3 為第 3 維度 ( $SC\_dim="3"$ )。首先，令正在處理的資料維度  $Data\_dim="1"$ ，且  $R=R1$ ， $L=L1$ ，而第一多工器  $mux1$  輸出第二旗標  $Flag\_b$  作為區間信號  $Pt$ 。因此，於 <步驟 C1> 中，過濾模組 453a 先判斷第一資料點的第 1 維度數值是否位於第 1 資料維度的過濾區間  $R1\sim(R1+2L1)$ 。假設第一資料點的第 1 維度數值位於過濾區間  $R1\sim(R1+2L1)$ ，則切割拖曳器 45 以第二旗標  $Flag\_b="1"$  作為區間信號  $Pt$ ，並儲存於暫存器  $Reg$ 。區間信號  $Pt$  相當於第一資料點的第 1 維度數值是否位於過濾區間  $R1\sim(R1+2L1)$  的判斷結果。

【0073】 接著，令正在處理的資料維度  $Data\_dim="2"$ ，且  $R=R2$ ， $L=L2$ ，而第一多工器  $mux1$  輸出第二旗標  $Flag\_b$  作為區間信號  $Pt$ 。因此，於 <步驟 C2>，過濾模組 453a 判斷第一資料點的第 2 維度數值是否位於第 2 資料維度的過濾區間  $R2\sim(R2+2L2)$ 。假設第一資料點的第 2 維度數值位於過濾區間  $R2\sim(R2+2L2)$ ，則切割拖曳器 45 以第二旗標  $Flag\_b="1"$  作為區間信號  $Pt$ 。因此，區間信號  $Pt$  與前一級區間信號  $Pt-1$  經過及閘後儲存於暫存器  $Reg$ 。

【0074】 其中，前一級區間信號  $Pt-1$  相當於第一資料點的第 1 維度數值是否位於過濾區間  $R1\sim(R1+2L1)$  的判斷結果；且區間信號  $Pt$  相當於第一資料點的第 2 維度數值是否位於過濾區間  $R2\sim(R2+2L2)$  的判斷結果。因此，區間信號  $Pt$  與前一級區間信號  $Pt-1$  經過及閘後的結果，就相當於第一資料點的第 1 維度數值與第 2 維度數值是否皆位於過濾區間  $R1\sim(R1+2L1)$ 、 $R2\sim(R2+2L2)$  的判斷結果。

【0075】 接著，令正在處理的資料維度  $Data\_dim="3"$ ，且  $R=R3$ ， $L=L3$ ，而第一多工器  $mux1$  輸出第一旗標  $Flag\_a$  作為區間信號  $Pt$ 。因此，於 <步驟 C3>，過濾模組 453a 判斷第一資料點

的第 3 維度數值是否位於第 3 資料維度的過濾區間  $R3 \sim (R3+L3)$ 。假設第一資料點的第 3 維度數值位於過濾區間  $R3 \sim (R3+L3)$ ，則切割拖曳器 45 以第一旗標  $Flag\_a=“1”$  作為區間信號  $Pt$ 。因此，區間信號  $Pt$  與前一級區間信號  $Pt-1$  經過及閘後儲存於暫存器  $Reg$ 。

**【0076】** 其中，前一級區間信號  $Pt-1$  相當於第一資料點的第 1 維度數值與第 2 維度數值是否接位於過濾區間的判斷結果；且區間信號  $Pt$  相當於第一資料點的第 3 維度數值是否位於過濾區間  $R3 \sim (R3+L3)$  的判斷結果。因此，區間信號  $Pt$  與前一級區間信號  $Pt-1$  經過及閘後的結果，就相當於第一資料點的第 1 維度數值、第 2 維度數值與第 3 維度數值是否皆位於過濾區間  $R1 \sim (R1+2L1)$ 、 $R2 \sim (R2+2L2)$ 、 $R3 \sim (R3+L3)$  的判斷結果。

**【0077】** 很明顯地，上述 <步驟 C1>~<步驟 C3> 中，係根據第 3 維度的模擬切割 3 ( $SC\_dim=“3”$ ) 所建立的過濾條件。而根據正在處理的資料維度  $Data\_dim$  的變化，依序產生三個過濾區間。再者，利用三個過濾區間進行判斷所獲得的三個區間信號  $Pt$  必需都要為“1”時，才會使最後儲存在暫存器  $Reg$  的區間信號  $Pt$  為“1”。此時可確定第一資料點位在  $c1$  子分區中，並使得結果信號  $Check=“1”$ ，並累計至累加器 46 中，使得累加器 46 中的累計值遞增 1。反之，假設最後儲存在暫存器  $Reg$  的區間信號  $Pt$  為“0”時，即確定第一資料點沒有位在  $c1$  子分區中，使得結果信號  $Check=“0”$ ，而無法改變累加器 46 的數值。

**【0078】** 假設資料空間中，資料點數目共有  $M\_data$  筆，則上述 <步驟 C1> 至 <步驟 C3> 必須進行  $M\_data$  次之後，累加器 46 中的數值即為  $c1$  子分區中資料點數目。

**【0079】** 由以上的說明可知，為了要獲得  $a1$  子分區中的資料點數目，<步驟 A1> 至 <步驟 A3> 需要執行  $M\_data$  次。同理，為了要獲得  $b1$  子分區中的資料點數目，<步驟 B1> 至 <步驟 B3> 需要執行  $M\_data$  次；為了要獲得  $c1$  子分區中的資料點數目，<步驟 C1> 至 <步驟 C3> 需要執行  $M\_data$  次。



【0080】 由於上述<步驟 A1>~<步驟 A3>、<步驟 B1>~<步驟 B3>、<步驟 C1>~<步驟 C3>步驟非常類似，因此本發明提出第 7A 圖之過濾與計數模組的第二實施例，可將上述步驟有效地合併，用以降低過濾與計數模組計數的時間。

【0081】 如第 7A 圖所示，過濾與計數模組 553 包含過濾模組 553a 與計數模組 553b。過濾模組 553a 至少包括：一比較器電路 51、一旗標產生電路 52。計數模組 553b 至少包括：一切割拖曳器陣列(cutting trailer array)55、與一累加器 56。其中，SC\_dim 為模擬切割的資料維度；Data\_dim 為正在處理的資料維度，R 與 L 為一個分區中，單一維度的起始點與半長度，因此 R、(R+L)、(R+2L)即為單一維度的邊界資訊。

【0082】 相較於第一實施例之計數模組 453b，其差異在於切割拖曳器陣列 55。而第 7B 圖所示之比較器電路 51 與旗標產生電路 52 之運作原理與第 6B 圖相同，動作原理不再贅述。

【0083】 再者，如第 7C 圖所示，其為切割拖曳器陣列示意圖。切割拖曳器陣列 55 可建立各維度的過濾區間，並以旗標信號作為判斷的結果。切割拖曳器陣列 55 接收旗標產生電路 52 發出的旗標信號 Flag\_a、Flag\_b 後，根據 SC\_dim 與 Data\_dim 選擇一旗標信號。切割拖曳器陣列 55 中包括三個切割拖曳器 551、552、553 與一並列轉串列電路(parallel to serial circuit, P/S)57。其中，三個切割拖曳器 551、552、553 的架構皆相同於第 6C 圖之切割拖曳器 45。再者，切割拖曳器 551 係處理第 1 維度的模擬切割 1，因此，將模擬切割的資料維度 SC\_dim="1"輸入至切割拖曳器 551。切割拖曳器 552 係處理第 2 維度的模擬切割 2，因此，將模擬切割的資料維度 SC\_dim="2"輸入至切割拖曳器 552。切割拖曳器 553 係處理第 3 維度的模擬切割 3，因此，將模擬切割的資料維度 SC\_dim="3"輸入至切割拖曳器 553。

【0084】 因此，當第一資料點輸入至過濾模組 553a，且針對第一資料點在三個維度的數值與三個維度的過濾條件皆進行判

斷之後，根據計數模組 553b 輸出的結果信號 Chk1、Chk2、Chk3 即可分別得知該第一資料點是否位於子分區 a1、b1、c1。而並列轉串列電路 57 將結果信號 Chk1、Chk2、Chk3 轉換成為串列結果信號 Check，並傳遞至累加器 56，而累加器 56 即可分別針對子分區 a1、b1、c1 的資料點數目進行累計。

【0085】 因此，假設資料空間中，資料點數目共有  $M\_data$  筆，則上述步驟進行  $M\_data$  次之後，累加器 56 中即可分別累計 a1、b1、c1 子分區中資料點數目。

【0086】 相較於第一實施例的過濾與計數模組 453，第二實施例的過濾與計數模組 553 確實可以縮短計數的時間，並獲得一個分區中，所有模擬切割所生成的子分區中的資料點數目。

【0087】 請再參考第 5A 圖，於原始分區組合中，前一次的選定切割可決定第 B 分區與第 C 分區。再者，第 B 分區與第 C 分區僅在於第 1 維度的過濾區間不同，而其他維度的過濾區間完全相同。因此，而第 B 分區與第 C 分區可視為相互對稱。舉例來說，第 B 分區可稱為輸入分區(input region)，第 C 分區可稱為對稱分區(symmetrical region)。

【0088】 因此，根據以上描述之特性可知，一個原始分區組合中至少會有二個分區互為對稱，且該對稱的二個分區之位置係由原始分區組合的前一次選定切割決定。在輸入分區與對稱分區各個資料維度中，除了在前一次的選定切割所對應之資料維度的邊界不同而具有對稱關係外，輸入分區與對稱分區在其他資料維度的邊界均相同。再者，貝氏序列切割控制器 31 儲存於比較條件記憶體 33 中的分區資訊所對應的分區即為輸入分區，而邊界產生模組 351 可同時產生輸入分區與對稱分區的邊界資訊。

【0089】 根據於上述之特性，本發明提出第 8A 圖之過濾與計數模組的第三實施例，可更進一步地以提升過濾與計數模組執行的速度。

【0090】 第三實施例之過濾與計數模組 653 包含過濾模組

653a 與計數模組 653b。過濾模組 653a 至少包括：一比較器電路 61 與一旗標產生電路 62。計數模組 653b 至少包括：一切割拖曳器陣列 65 與一累加器 66。其中，SC\_dim 為模擬切割的資料維度；Data\_dim 為正在處理的資料維度；PC\_dim 為前一選定切割的資料維度；Sym\_part 為對稱部指示信號；R 與 L 為一個分區中，單一維度的起始點與長度，因此， $R$ 、 $(R+L)$ 、 $(R+2L)$ 、 $(R+3L)$ 、 $(R+4L)$  即為輸入分區與對稱分區在單一維度的邊界資訊。

【0091】 如第 8B 圖所示，其為比較器電路與旗標產生電路之運作原理。比較器電路 61 中至少包括六個比較器。其中，第一比較器判斷資料點是否小於 $(R+4L)$ ，當資料點小於 $(R+4L)$ 時，第一比較結果 Cmp1="1"，當資料點大於或等於 $(R+4L)$ 時，第一比較結果 Cmp1="0"。第二比較器判斷資料點是否小於 $(R+3L)$ ，當資料點小於 $(R+3L)$ 時，第二比較結果 Cmp2="1"，當資料點大於或等於 $(R+3L)$ 時，第二比較結果 Cmp2="0"。第三比較器判斷資料點是否大於等於 $(R+2L)$ ，當資料點大於或等於 $(R+2L)$ 時，第三比較結果 Cmp3="1"，當資料點小於 $(R+2L)$ 時，第三比較結果 Cmp3="0"。第四比較器判斷資料點是否小於 $(R+2L)$ ，當資料點小於 $(R+2L)$ 時，第四比較結果 Cmp4="1"，當資料點大於或等於 $(R+2L)$ 時，第四比較結果 Cmp4="0"。第五比較器判斷資料點是否小於 $(R+L)$ ，當資料點小於 $(R+L)$ 時，第五比較結果 Cmp5="1"，當資料點大於或等於 $(R+L)$ 時，第五比較結果 Cmp5="0"。第六比較器判斷資料點是否大於或等於 R，當資料點大於或等於 R 時，第六比較結果 Cmp6="1"，當資料點小於 R 時，第六比較結果 Cmp6="0"。

【0092】 再者，旗標產生電路 62 根據比較器電路 61 產生的六個比較結果來產生四個旗標。當資料點位於  $R \sim (R+L)$  的區間時，Cmp6 與 Cmp5 皆為"1"，此時設定第四旗標 Flag4="1"。當資料點位於  $R \sim (R+2L)$  的區間時，Cmp6 與 Cmp4 皆為"1"，此時設定第三旗標 Flag3="1"。當資料點位於  $(R+2L) \sim (R+3L)$  的區間時，Cmp3

與 Cmp2 皆為“1”，此時設定第二旗標 Flag2=“1”。當資料點位於 (R+2L)~(R+4L)的區間時，Cmp3 與 Cmp1 皆為“1”，此時設定第一旗標 Flag1=“1”。

**【0093】** 如第 8C 圖所示，其為切割拖曳器陣列示意圖。切割拖曳陣列 65 可建立輸入分區與對稱分區中各維度的過濾區間，並以旗標信號作為判斷的結果。切割拖曳器陣列 65 進一步包含輸入分區之切割拖曳器陣列 651，與對稱分區之切割拖曳器陣列 652。切割拖曳器陣列 65 接收模擬切割的資料維度 SC\_dim、正在處理的資料維度 Data\_dim、前一選定切割的資料維度 PC\_dim、對稱部指示信號 Sym\_part、以及四個旗標信號 Flag1~Flag4。

**【0094】** 如第 8D 圖所示，其為切割拖曳器陣列的詳細電路圖。以 3 維的資料空間為例，切割拖曳器陣列 55 包括六個切割拖曳器 651a、651b、651c 與 652a、652b、652c。這六個切割拖曳器 651a、651b、651c、652a、652b、652c 的架構類似，但輸入的信號不完全相同。

**【0095】** 首先，當對稱部指示信號 Sym\_part=“0”時，切割拖曳器使用於輸入分區；反之，當對稱部指示信號 Sym\_part=“1”時，切割拖曳器使用於對稱分區。第 8D 圖中上方的三個切割拖曳器 652a、652b、652c 形成對稱分區之切割拖曳器陣列 652，並使用於判斷資料點是否位於對稱分區之子分區；下方的三個切割拖曳器 651a、651b、651c 形成輸入分區之切割拖曳器陣列 651，並用於判斷資料點是否位於輸入分區之子分區。再者，由於每個切割拖曳器的架構完全相同，以下僅以切割拖曳器 651a 為例來作說明。

**【0096】** 當對稱部指示信號 Sym\_part=“0”時，無論正在處理的資料維度 Data\_dim 與前一選定切割的資料維度 PC\_dim 是否相同，第一多工器 mux1 維持將第三旗標 Flag3 作為第一信號 S1 且第二多工器 mux2 維持將第四旗標 Flag4 作為第二信號 S2。

**【0097】** 進一步的，當模擬切割的資料維度 SC\_dim 與正在

處理的資料維度  $Data\_dim$  相同時，第三多工器  $mux3$  即以第二信號  $S2$ (相當於第四旗標  $Flag4$ )作為區間信號  $Pt$ ；當模擬切割的資料維度  $SC\_dim$  與正在處理的資料維度  $Data\_dim$  不相同時，第三多工器  $mux3$  區即以第一信號  $S1$ (相當於第三旗標  $Flag3$ )作為區間信號  $Pt$ 。

【0098】 當對稱部指示信號  $Sym\_part="1"$ 時，第一多工器  $mux1$  與第二多工器  $mux2$  的輸出結果，則需根據正在處理的資料維度  $Data\_dim$  與前一選定切割的資料維度  $PC\_dim$  是否相同而決定。

【0099】 若正在處理的資料維度  $Data\_dim$  與前一選定切割的資料維度  $PC\_dim$  相同時，第一多工器  $mux1$  將第一旗標  $Flag1$  作為第一信號  $S1$  且第二多工器  $mux2$  將第二旗標  $Flag2$  作為第二信號  $S2$ 。反之，若正在處理的資料維度  $Data\_dim$  與前一選定切割的資料維度  $PC\_dim$  不相同時，第一多工器  $mux1$  將第三旗標  $Flag3$  作為第一信號  $S1$ ，且第二多工器  $mux2$  將第四旗標  $Flag4$  作為第二信號  $S2$ 。

【0100】 再者，當模擬切割的資料維度  $SC\_dim$  與正在處理的資料維度  $Data\_dim$  相同時，第三多工器  $mux3$  即以第二信號  $S2$ (第二旗標  $Flag2$  或第四旗標  $Flag4$ )作為區間信號  $Pt$ 。當模擬切割的資料維度  $SC\_dim$  與正在處理的資料維度  $Data\_dim$  不相同時，第三多工器  $mux3$  區即以第一信號  $S1$ (第一旗標  $Flag1$  或第三旗標  $Flag3$ )作為區間信號  $Pt$ 。

【0101】 再者，暫存器  $Reg$  儲存的資料經由第四多工器  $mux4$  輸出為前一級區間信號  $Pt-1$ ，而間信號  $Pt$  與前一級區間信號  $Pt-1$  輸入及閘( $and\ gate$ )而產生的結果再次儲存於暫存器  $Reg$ 。再者，由第 8D 圖可知，當正在處理的資料維度  $Data\_dim$  為第 1 維度時，第四多工器  $mux2$  會將“1”作為前一級區間信號  $Pt-1$ 。

【0102】 在第三實施例的過濾與計數模組 653 中，切割拖曳器 651 係根據模擬切割的資料維度  $SC\_dim$  與正在處理的資料維

度 Data\_dim，產生區間信號 Pt，並將區間信號 Pt 儲存於暫存器 Reg 中。而所有資料維度皆判斷完成後，暫存器 Reg 即產生一檢查結果信號 Chki1。

【0103】 假設切割拖曳器 651 係處理輸入分區時，於結果信號 Chki1="1"時，代表資料點位在輸入分區模擬切割所生成的一個子區間內；反之，結果信號 Chki1="0"時，代表資料點沒有位在輸入分區模擬切割所生成的一個子區間。

【0104】 由以上之說明可知，切割拖曳器 651a 係處理輸入分區中第 1 維度的模擬切割 1，因此 SC\_dim="1"；切割拖曳器 651b 係處理輸入分區中第 2 維度的模擬切割 2，因此 SC\_dim="2"；切割拖曳器 651c 係處理輸入分區中第 3 維度的模擬切割 3，因此 SC\_dim="3"。再者，切割拖曳器 652a 係處理對稱分區中第 1 維度的模擬切割 1，因此 SC\_dim="1"；切割拖曳器 652b 係處理對稱分區中第 2 維度的模擬切割 2，因此 SC\_dim="2"；切割拖曳器 652c 係處理對稱分區中第 3 維度的模擬切割 3，因此 SC\_dim="3"。

【0105】 因此，當第一資料輸入至比較器電路 61 且三個維度的過濾條件皆進行判斷之後，根據切割拖曳器 651a、651b、651c 輸出的結果信號 Chki1、Chki2、Chki3 即可得知該第一資料點是否位於輸入分區所生成的三個子分區，並且，根據切割拖曳器 652a、652b、652c 輸出的結果信號 Chks1、Chks2、Chks3，即可得知該第一資料點是否位於對稱分區所生成的三個子分區。

【0106】 再者，而並列轉串列電路 671 將結果信號 Chki1、Chki2、Chki3 轉換成為串列結果信號 Checki，並傳遞至累加器 66，並列轉串列電路 672 將結果信號 Chks1、Chks2、Chks3 轉換成為串列結果信號 Checks，並傳遞至累加器 66。

【0107】 因此，而累加器 66 即可分別進行輸入分區以及對稱分區中個別子分區的資料點數目之累計。其中，各個子分區的資料點數目之累計結果可分別儲存在計數結果記憶體內。當與一子

分區對應的結果信號為 1 時，先由計數結果記憶體讀出與該子分區對應的累計數值。之後，再利用累加器 66 對與該子分區對應的累計數值進行累加。並且，於累加後，再將累加後的數值存回計數結果記憶體。

【0108】 因此，假設資料空間中，資料點數目共有  $M_{data}$  筆，則上述步驟進行  $M_{data}$  次之後，累加器 66 中即可分別累計輸入分區中三個子分區的資料點數目，以及累計對稱分區中三個子分區的資料點數目。

【0109】 相較於第二實施例的過濾與計數模組 553，第三實施例的過濾與計數模組 653 確實可以降低計數的時間，並獲得一個分區中，所有模擬切割所生成的子分區中的資料點數目。

#### 具平行化處理的貝氏循序切割系統

【0110】 多維資料空間的貝氏循序切割系統可進一步以平行化架構來實現，用以同步判斷兩個資料點是否位於資料分區。採用此種作法的過濾與計數模組，可更進一步地以提升過濾與計數模組執行的速度。

【0111】 請參見第 9 圖，其係本發明多維資料空間的貝氏循序切割系統的另一實施例的示意圖。第 9 圖雖未繪示貝氏序列切割控制器，但其系統架構與第 4 圖大致類似。

【0112】 比較條件記憶體 73 包含：起點記憶體 73a、半長度記憶體 73b。其中，起點記憶體 73a 用於儲存貝氏循序切割控制器輸出的 R；半長度記憶體 73b 用於儲存貝氏循序切割控制器輸出的 L。計數結果記憶體 77 包含計數結果暫存區 77a、77b。其中，計數結果暫存區 77a 用於儲存與輸入分區對應之計數結果；計數結果暫存區 77b 用於儲存與對稱分區對應之計數結果。

【0113】 邊界產生模組 751 電連接至比較條件記憶體 73、過濾與計數模組 753，且過濾與計數模組 753 電連接至計數結果

記憶體 77。邊界產生模組 751 可自比較條件記憶體 73 取得分區資訊  $R$ 、 $L$ ，並據此而產生每一維度的邊界資訊  $R$ 、 $(R+L)$ 、 $(R+2L)$ 、 $(R+3L)$ 、 $(R+4L)$  至過濾與計數模組 753。

【0114】 過濾與計數模組 753 包含兩個過濾模組 753a、753b 與兩個計數模組 753c、753d。其中，過濾模組 753a、753b 的運作方式與第 8A 圖的過濾模組 653a 類似；計數模組 753c、753d 的運作方式與第 8A 圖的計數模組 653b 類似。

【0115】 在此實施例中，過濾與計數模組 753 同時對第一資料點  $data1$  與第二資料點  $data2$  進行過濾與計數。過濾模組 753a 與計數模組 753c 用於判斷第一資料點  $data1$  的串列結果信號  $Checki$ 、 $Checks$ ；過濾模組 753b 與計數模組 753d 用於判斷第二資料點  $data2$  的串列結果信號  $Checki$ 、 $Checks$ 。

【0116】 其中，判斷第一資料點  $data1$  是否位於輸入分區而產生的串列結果信號  $Checki$ ，以及判斷第二資料點  $data2$  是否位於輸入分區而產生的串列結果信號  $Checki$  經過累加器 781 累計後，存放至計數結果暫存區 77a。同樣的，判斷第一資料點  $data1$  是否位於對稱分區而產生的串列結果信號  $Checks$ ，以及判斷第二資料點  $data2$  是否位於對稱分區而產生的串列結果信號  $Checks$  經過累加器 782 累計後，存放至計數結果暫存區 77b。

【0117】 實際應用時，同步輸入的資料點個數與搭配使用之計數引擎的個數並不以此為限。進一步的，本發明可將多個計數引擎結合於一可設定的計數引擎 (configurable counting engine，簡稱為 CCE) 內。藉由多個過濾與計數模組的結合，可設定的計數引擎能快速的具有更高資料維度的資料點進行分析。或者，加速對資料點的分析速度。

【0118】 請參見第 10 圖，其係本發明多維資料空間的貝氏循序切割系統的再一實施例之示意圖。此實施例假設計數引擎 85 包含八個過濾與計數模組 853a、853b、...853h。針對這些過濾與計數模組 853a、853b、...853h，比較條件記憶體 83 分別



提供相對應的比較條件暫存區 83a、83b...83h；且計數結果記憶體 87 亦分別提供相對應的計數結果暫存區 87a、87b、...87h。

**【0119】** 在此實施例中，計數引擎 85 能夠被彈性設定，即，根據資料點的資料維度多寡而對過濾與計數模組 853a、853b、...853h 進行設定，進而使過濾與計數模組 853a、853b、...853h 分為一個或多個群組。在同一個群組內的過濾與計數模組以串接方式彼此相連，透過串接的方式，讓計數引擎 85 能過濾的資料維度，能夠高於單一個過濾與計數模組能判斷的資料維度。

**【0120】** 請參見第 11A、11B、11C、11D 圖，其係本發明實施例對計數引擎進行設定的示意圖。當計數引擎 80 將過濾與計數模組被設定為第一分組模式(mode1)時，過濾與計數模組 DFC1、DFC2、DFC3、DFC4、DFC5、DFC6、DFC7、DFC8 被區分為一個群組(partition group)。假設每一過濾與計數模組可處理 128 個資料維度。當計數引擎 80 採用第 11A 圖之設定時，可對具有  $1024(128*8=1024)$ 個資料維度的過濾條件進行資料點的計數。

**【0121】** 當計數引擎 80 被設定為第二分組模式(mode2)時，過濾與計數模組 DFC1、DFC2、DFC3、DFC4、DFC5、DFC6、DFC7、DFC8 被區分為兩個群組。第一個群組 G1 包含過濾與計數模組 DFC1、DFC2、DFC3、DFC4；第二個群組 G2 包含過濾與計數模組 DFC5、DFC6、DFC7、DFC8。當計數引擎 80 採用第 11B 圖之設定時，可同時對兩組具有  $512(128*4=512)$ 個資料維度的過濾條件進行資料點的計數。

**【0122】** 當計數引擎 80 被設定為第三分組模式(mode3)時，計數引擎 80 將過濾與計數模組 DFC1、DFC2、DFC3、DFC4、DFC5、DFC6、DFC7、DFC8 區分為四個群組。第一個群組 G1 包含過濾與計數模組 DFC1、DFC2；第二個群組 G2 包含過濾與計數模組 DFC3、DFC4；第三個群組 G3 包含過濾與計數模組

DFC5、DFC6；第四個群組 G4 包含過濾與計數模組 DFC7、DFC8。當計數引擎 80 採用第 11C 圖之設定時，可同時對四組具有 256( $128*2=256$ )個資料維度的過濾條件進行資料點的計數。

**【0123】** 當計數引擎 80 將過濾與計數模組被設定為第四分組模式(mode4)時，每一個過濾與計數模組 DFC1、DFC2、DFC3、DFC4、DFC5、DFC6、DFC7、DFC8 均對應於一個群組 G1、G2、G3、G4、G5、G6、G7、G8。當計數引擎 80 採用第 11D 圖之設定時，可同時對八組具有 128 個資料維度的過濾條件進行資料點的計數。

**【0124】** 請參見第 12 圖，其係本發明實施例之計數晶片提供多個計數引擎(CCE)之示意圖。本案實施例的計數晶片 81 包含八個計數引擎 81a、81b、81c、81d、81e、81f、81g、81h，每一個計數引擎進一步包含 8 個過濾與計數模組 DFC1、DFC2、DFC3、DFC4、DFC5、DFC6、DFC7、DFC8。實際應用時，計數晶片包含的計數引擎個數，以及每個計數引擎包含之過濾與計數模組的個數並不以此為限。

**【0125】** 請參見第 13 圖，其係本發明實施例以序列方式連接多個計數晶片的示意圖。將計數晶片以串接方式連接時，能對具有更高資料維度的分區進行資料點的計數。貝氏序列切割控制器 910 將外部儲存裝置 919 內的資料點依序傳送至第一計數晶片 911、第二計數晶片 913、第三計數晶片 915、第四計數晶片 917。第一計數晶片 911 用於判斷資料點在資料維度 1-128 的數值是否符合過濾條件。第二計數晶片 913 用於判斷資料點在資料維度 129-256 的數值是否符合過濾條件。第三計數晶片 915 用於判斷資料點在資料維度 257-384 的數值是否符合過濾條件內。第四計數晶片 917 用於判斷資料點在資料維度 385-512 的數值是否符合過濾條件。

**【0126】** 請參見第 14 圖，其係本發明實施例以並聯方式連接多個計數晶片的示意圖。將計數晶片以並聯方式連接時，

能提升對資料點進行計數速度。例如：將  $M$  個資料點區分為四份各  $(1/4)*M$  個資料點的資料集合 928a、928b、928c、928d 後，輸入相同的邊界參數至四個計數晶片。其中，第一計數晶片 921、第二計數晶片 923、第三計數晶片 925、第四計數晶片 927 均只要計算  $(1/4)*M$  個資料點。之後，將四個計數晶片的計數結果進行加總後，貝氏序列切割控制器 920 便可據此進行後續判斷。將計數晶片並聯的作法，使單一計數晶片需要計算的資料量減少，能進一步減少整體對資料點計數所需的時間。

**【0127】** 進一步的，本發明亦可結合第 13、14 圖的概念，以管線(pipeline)方式提升整體的計數速度。例如：在第一時段內，第一計數晶片對第一組資料點的在資料維度為 1-128 的數值進行判斷。在第二時段內，第一計數晶片對第二組資料點在資料維度為 1-128 的數值進行判斷，且第二計數晶片對第一組資料點在資料維度為 129-256 的數值進行判斷等。在第三時段內，第一計數晶片對第三組資料點在資料維度為 1-128 的數值進行判斷；第二計數晶片對第二組資料點在資料維度為 129-256 的數值進行判斷；以及，第三計數晶片對第一組資料點在資料維度為 257-384 的數值進行判斷。其餘類推。

**【0128】** 根據前述說明可以得知，本發明可透過多個面向加速資料點個數的計數。包含：以扣除方式減少需計算之子分區數量；同時計算輸入分區與對稱分區；同步輸入兩個資料點等。此外，本發明進一步以精簡而彈性的電路實現計數引擎，並能搭配多種平行化處理的作法提升整體的計數速度。

**【0129】** 綜上所述，雖然本發明已以較佳實施例揭露如上，然其並非用以限定本發明。本發明所屬技術領域中具有通常知識者，在不脫離本發明之精神和範圍內，當可作各種之更動與潤飾。因此，本發明之保護範圍當視後附之申請專利範圍所界定者為準。

#### **【符號說明】**

## 【0130】

S11、S13、15、17、S18：步驟

200：資料空間

3：密度分析系統

31、910、920：貝氏序列切割控制器

33、73、83：比較條件記憶體

35、85、80、81a、81b、81c、81d、81e、81f、81g、81h：

## 計數引擎

37、77、87：計數結果記憶體

351、751：邊界產生模組

353a、53a、553a、653a、753a、753b：過濾模組

353b、453b、553b、653b、753c、753d：計數模組

353、453、553、653、753、853a、853b、853h、DFC1、DFC2、

DFC3、DFC4、DFC5、DFC6、DFC7、DFC8：過濾與計數模組

30：資料點儲存單元

a1、a2、b1、b2、c1、c2：子分區

3：密度分析系統

42、52、62：旗標產生電路

41、51、61：比較器電路

45、551、552、553、651a、651b、651c、652a、652b、652c：

## 切割拖曳器

46、56、66：累加器

55、65：切割拖曳器陣列

57、671：並列轉串列電路

651：輸入分區之切割拖曳器陣列

652：對稱分區之切割拖曳器陣列

73a：起點記憶體

73b：半長度記憶體

77a、77b、87a、87b、87h：計數結果暫存區

G1：第一個群組

G2：第二個群組

G3：第三個群組

G4：第四個群組

G5：第五個群組

G6：第六個群組

G7：第七個群組

G8：第八個群組

81：計數晶片

911、921：第一計數晶片

913、923：第二計數晶片

915、925：第三計數晶片

917、927：第四計數晶片

919：外部儲存裝置

928a、928b、928c、928d：資料集合

## 申請專利範圍

1. 一計數引擎，應用於 D 維度資料空間的一貝氏循序切割系統中，包括：
  - 一過濾模組，比較至少一受測資料點及與一子分區所對應的 D 個邊界資訊產生 D 個旗標組；以及
  - 一計數模組，連接至該過濾模組，根據該 D 個旗標組判斷該至少一受測資料點是否位於該子分區而產生一結果信號，並根據該結果信號而選擇性遞增與該子分區對應之一累計值。
2. 如申請專利範圍第 1 項所述之計數引擎，其中該計數引擎係電連接於一資料點儲存單元，且該過濾模組係自該資料點儲存單元接收複數個資料點，並自該等資料點中選擇該至少一受測資料點。
3. 如申請專利範圍第 2 項所述之計數引擎，其中當該等資料點均被輪續選擇作為該至少一受測資料點後，該累計值代表該等資料點中，位於該子分區的數目。
4. 如申請專利範圍第 1 項所述之計數引擎，其中該計數引擎更包括：
  - 一邊界產生模組，電連接於該過濾模組與一比較條件記憶體，其係自該比較條件記憶體接收與該子分區對應之一分區資訊，並產生該 D 個邊界資訊。
5. 如申請專利範圍第 4 項所述之計數引擎，其中該過濾模組包括：
  - 一比較器電路，電連接於該邊界產生模組，接收該 D 個邊界資訊，於該 D 個維度分別比較該 D 個邊界資訊與該受測資料點的數值，進而產生 D 個比較結果組；以及
  - 一旗標產生電路，電連接於該比較器電路與該計數模組，根據該 D 個比較結果組而產生 D 個旗標組。
6. 如申請專利範圍第 5 項所述之計數引擎，其中該計數模組包

括：

至少一切割拖曳器，根據該 D 個旗標組決定 D 個過濾區間以及對應的 D 個區間信號，並根據該 D 個區間信號產生該結果信號；以及

一累加器，於該結果信號動作時，遞增該累計值。

7. 一種多維資料空間的貝氏循序切割系統，連接至一資料點儲存單元，該資料點儲存單元儲存複數個資料點的各維度數值，該貝氏循序切割系統包括：

一控制器，產生一分區所對應的一分區資訊；

一比較條件記憶體，連接至該控制器，並暫存該分區資訊；

一計數引擎，連接於該比較條件記憶體與該資料點儲存單元，該計數引擎根據一第一模擬分割將該分區切割為一第一子分區與一第二子分區，以及該計數引擎根據該分區資訊產生一過濾條件運用於過濾該些資料點，並計數該些資料點位於該第一子分區中的一第一數目；以及

一計數結果記憶體，連接於該計數引擎與該控制器，用以暫存該第一數目，並傳遞至該控制器；

其中，該控制器中記錄該些資料點中有一第二數目位於該分區中，並將該第二數目減去該第一數目而獲得一第三數目，使得該控制器決定該些資料點中有該第三數目位於該第二子分區中；以及該控制器根據該第一數目與該第三數目而獲得該第一模擬切割個的一第一切割權重。

8. 如申請專利範圍第 7 項所述之貝氏循序切割系統，其中該計數引擎包括：

一邊界產生模組，連接至該比較條件記憶體，用以將該分區資訊轉換為多個邊界資訊；

一過濾模組，連接至該邊界產生模組，並根據該些邊界資訊建立該過濾條件用以判斷該些資料點，並決定該些資料點是否為於該第一子分區中；以及

計數模組，連接至該過濾模組，當該過濾模組決定該些資料點中的一個資料點位於該第一子分區時，遞增一累計值。

9. 如申請專利範圍第 8 項所述之貝氏循序切割系統，其中該過濾模組包括：

一比較器電路，根據該些邊界資訊決定該分區的一第一過濾區間，並接收該些資料點中一第一資料點，其中該比較電路比較該第一過濾區間與該第一資料點，產生複數個比較信號；以及

一旗標產生電路，接收該些比較信號並產生複數個旗標信號。

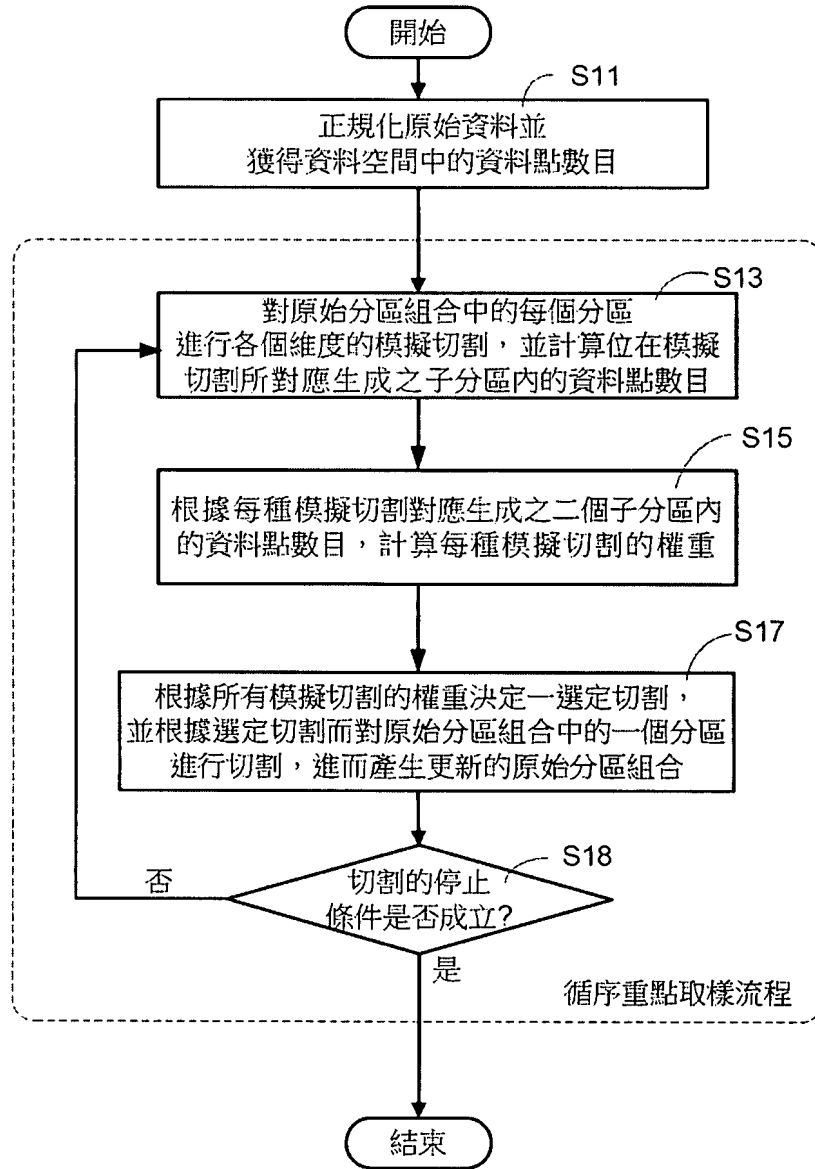
10. 如申請專利範圍第 9 項所述之貝氏循序切割系統，其中該記數模組包括：

一切割拖曳器，根據該些旗標信號，決定該第一資料點是否位於該第一過濾區間，其中當該第一資料點的各維度數值符合該些過濾區間時，該第一資料點位於該第一子分區，使得該切割拖曳器動作一結果信號；以及

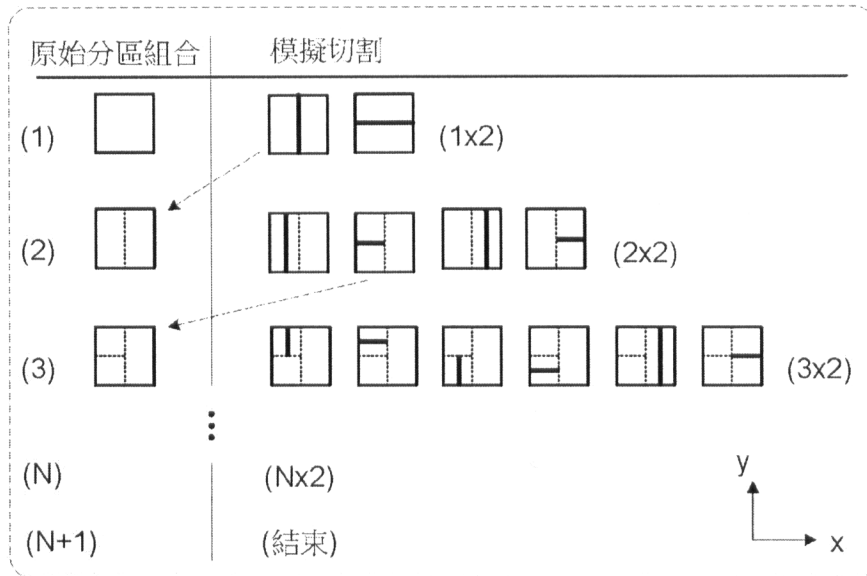
一累加器，於該結果信號動作時，遞增該累計值。



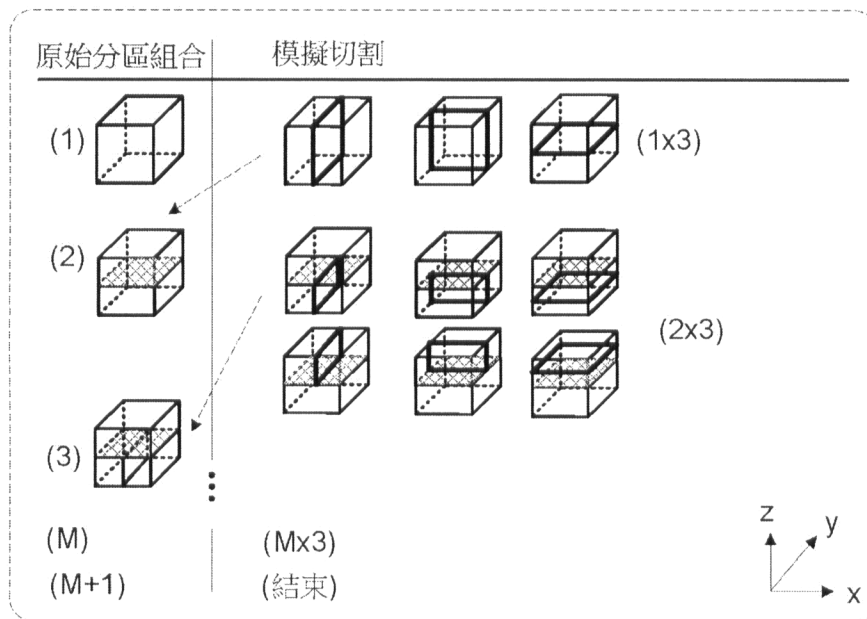
# 圖式



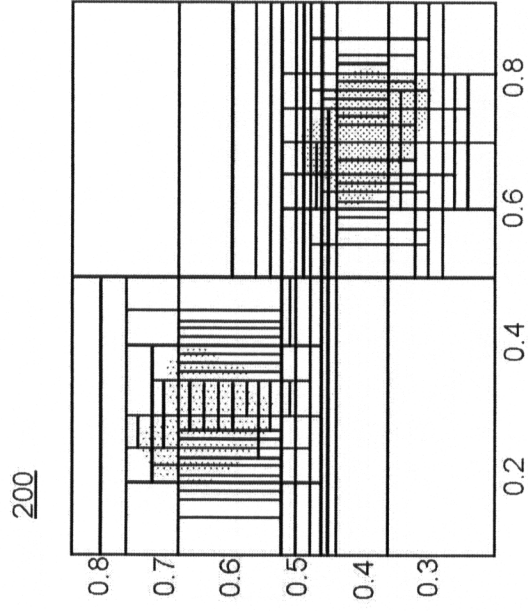
第1圖



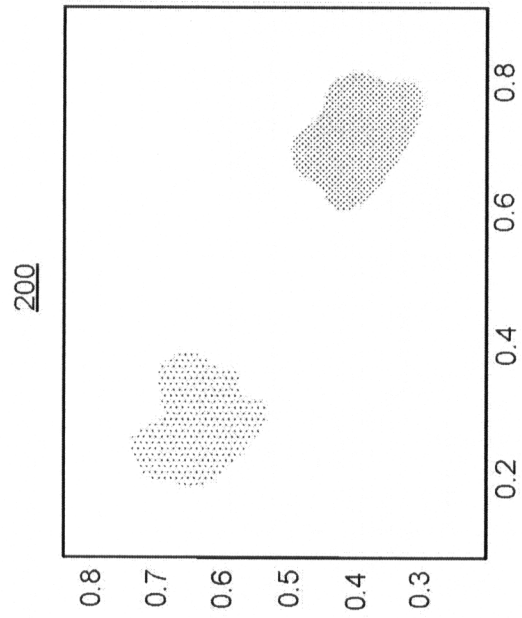
第2A圖



第2B圖



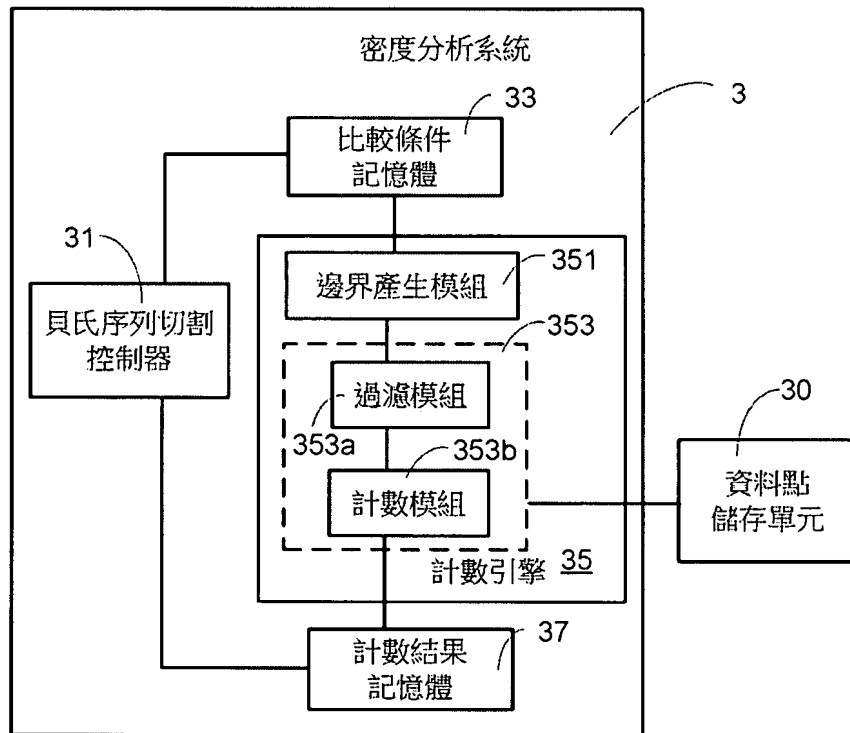
第2D圖



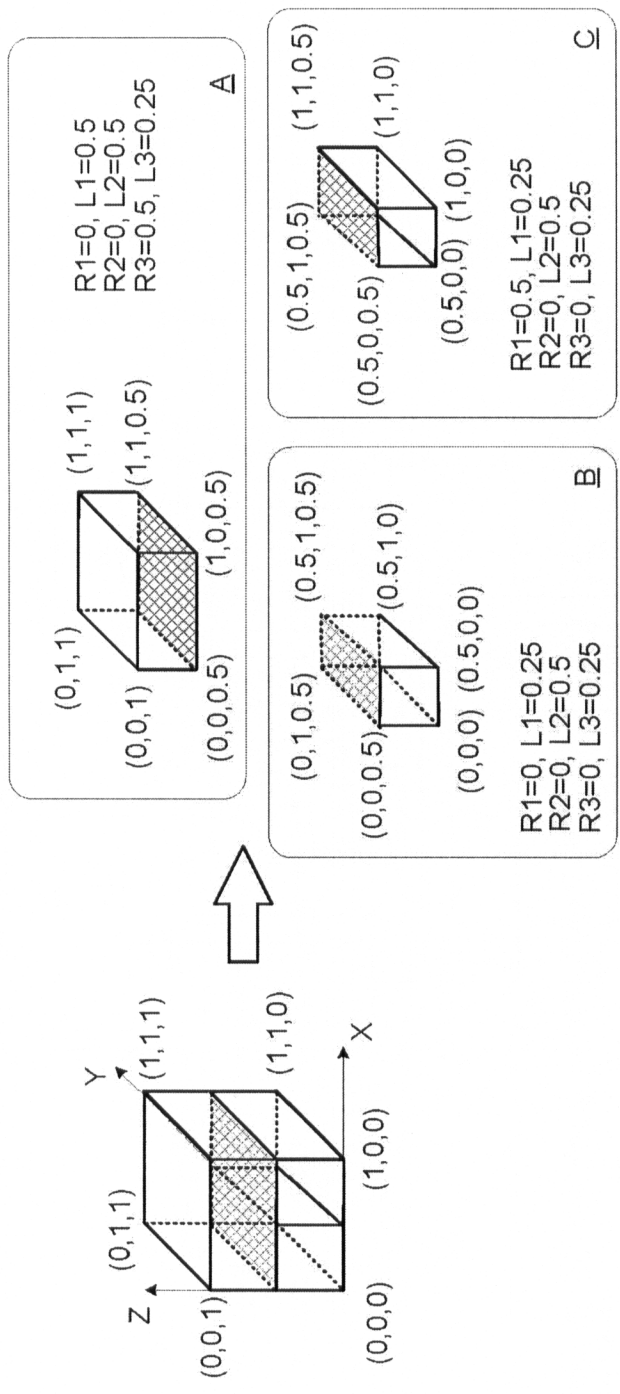
第2C圖

第P次切割流程	模擬切割的資料維度			
	原始分區	第1資料維度	第2資料維度	第d資料維度
1	WP_11	WP12	WP1d	WP1D
2	WP_21	WP22	WP2d	WP2D
3	WP_31	WP32	WP3d	WP3D
⋮	⋮	⋮	⋮	⋮
p	WP_p1	WP_p2	WP_pd	WP_pD
⋮	⋮	⋮	⋮	⋮
P	WP_P1	WP_P2	WP_Pd	WP_PD

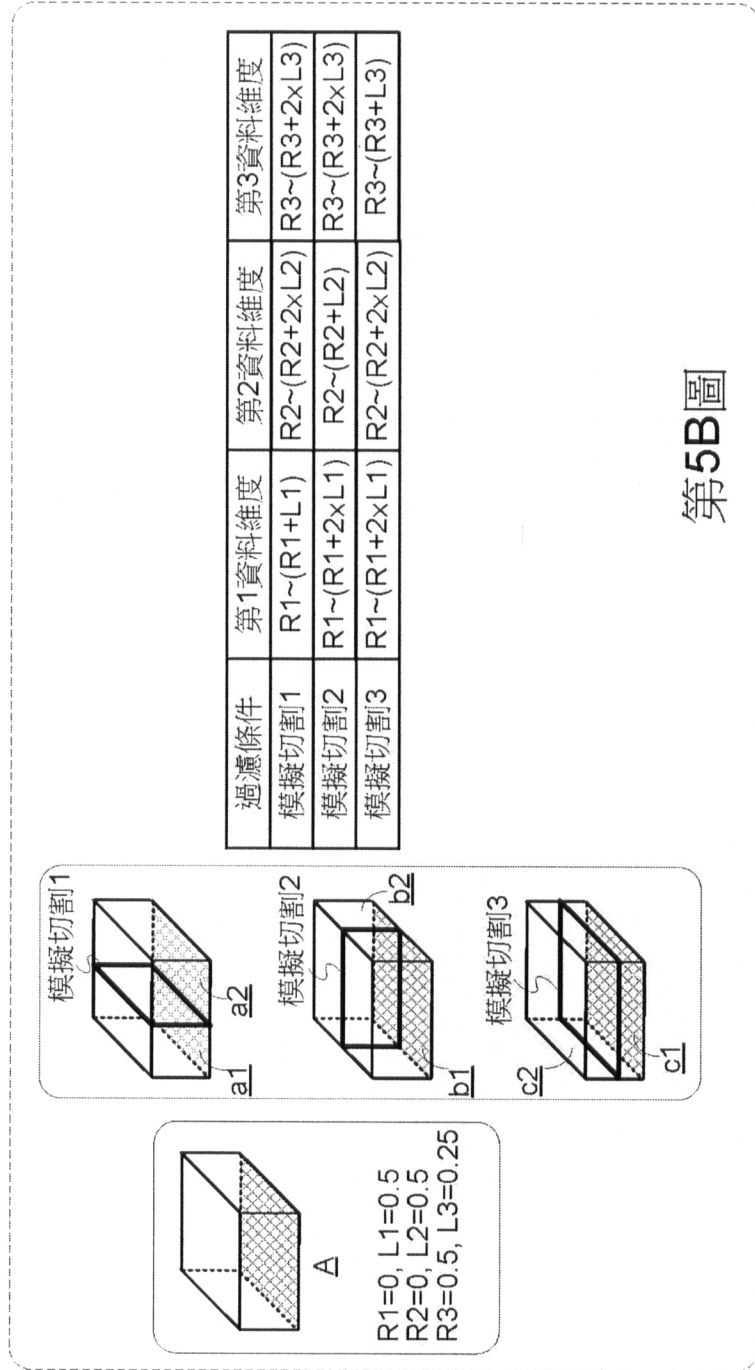
第3圖



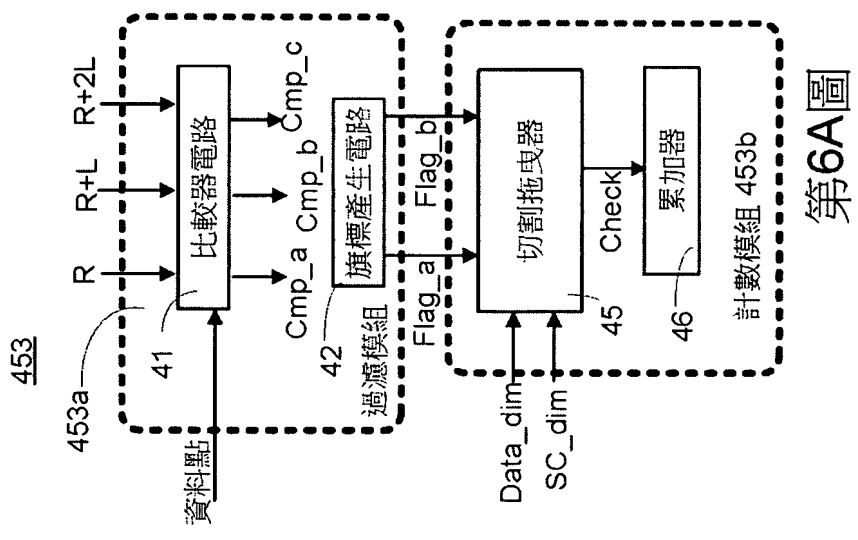
第4圖



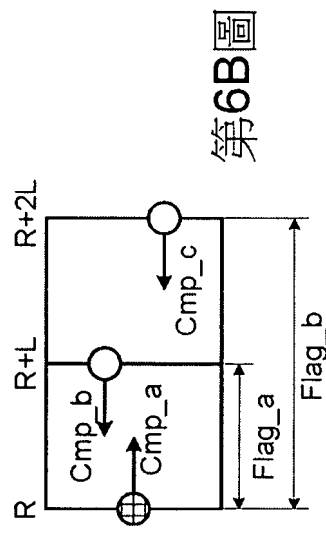
第5A圖



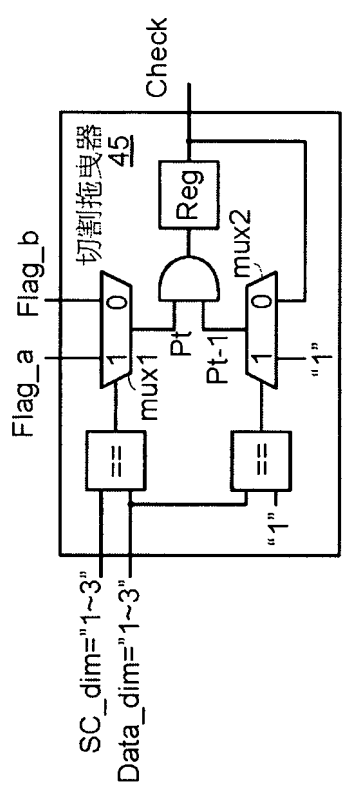
第5B圖



第6A圖

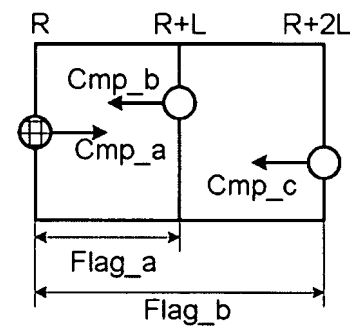
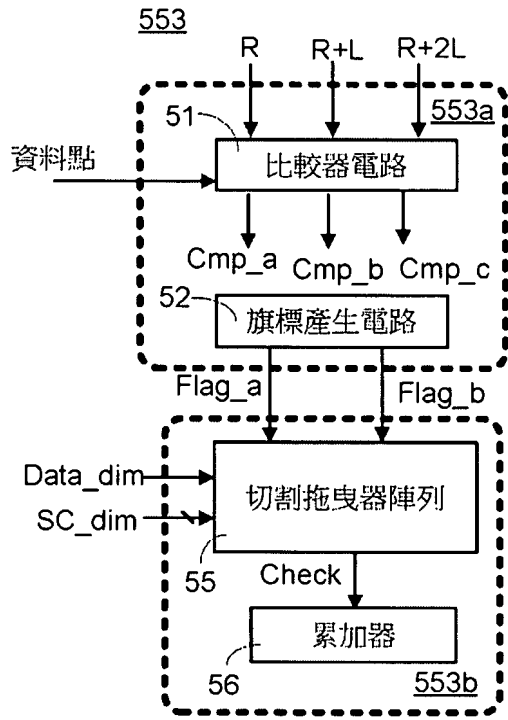


第6B圖



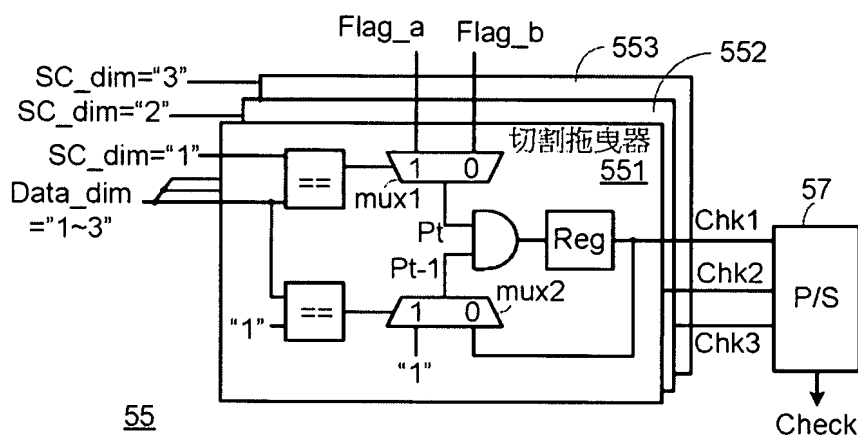
第6C圖



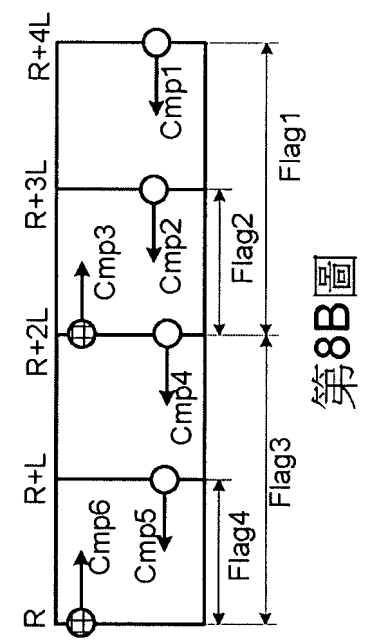


第7B圖

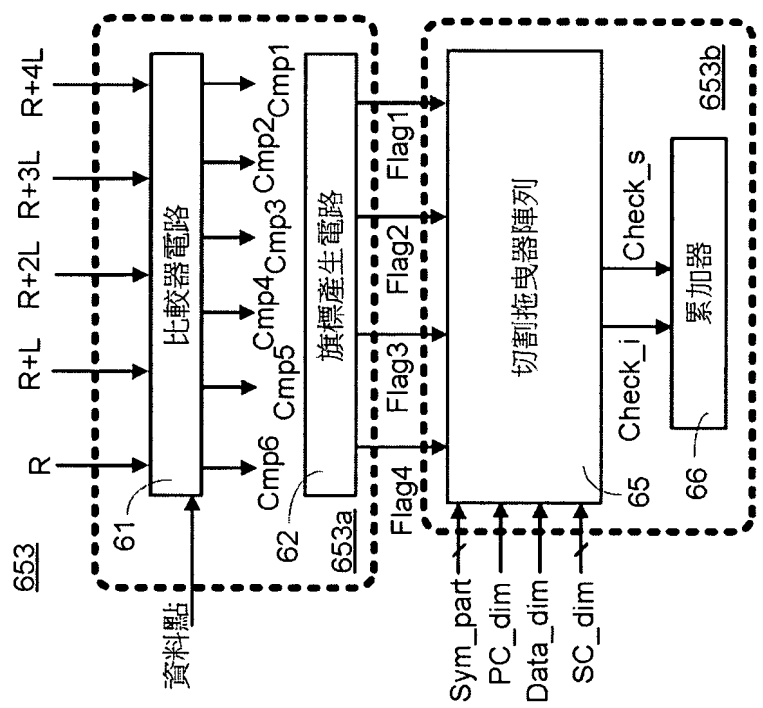
第7A圖



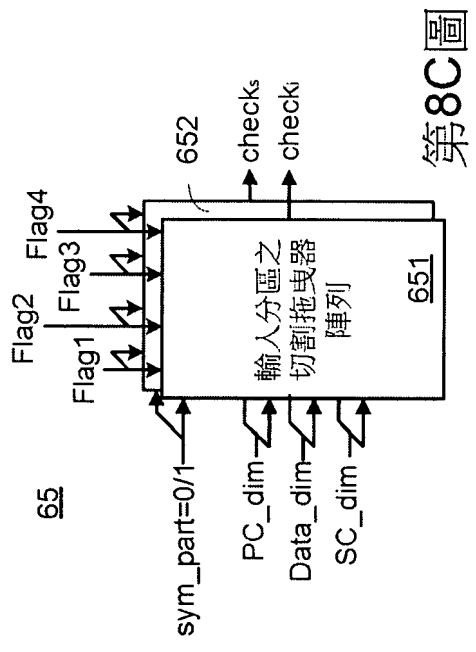
第7C圖



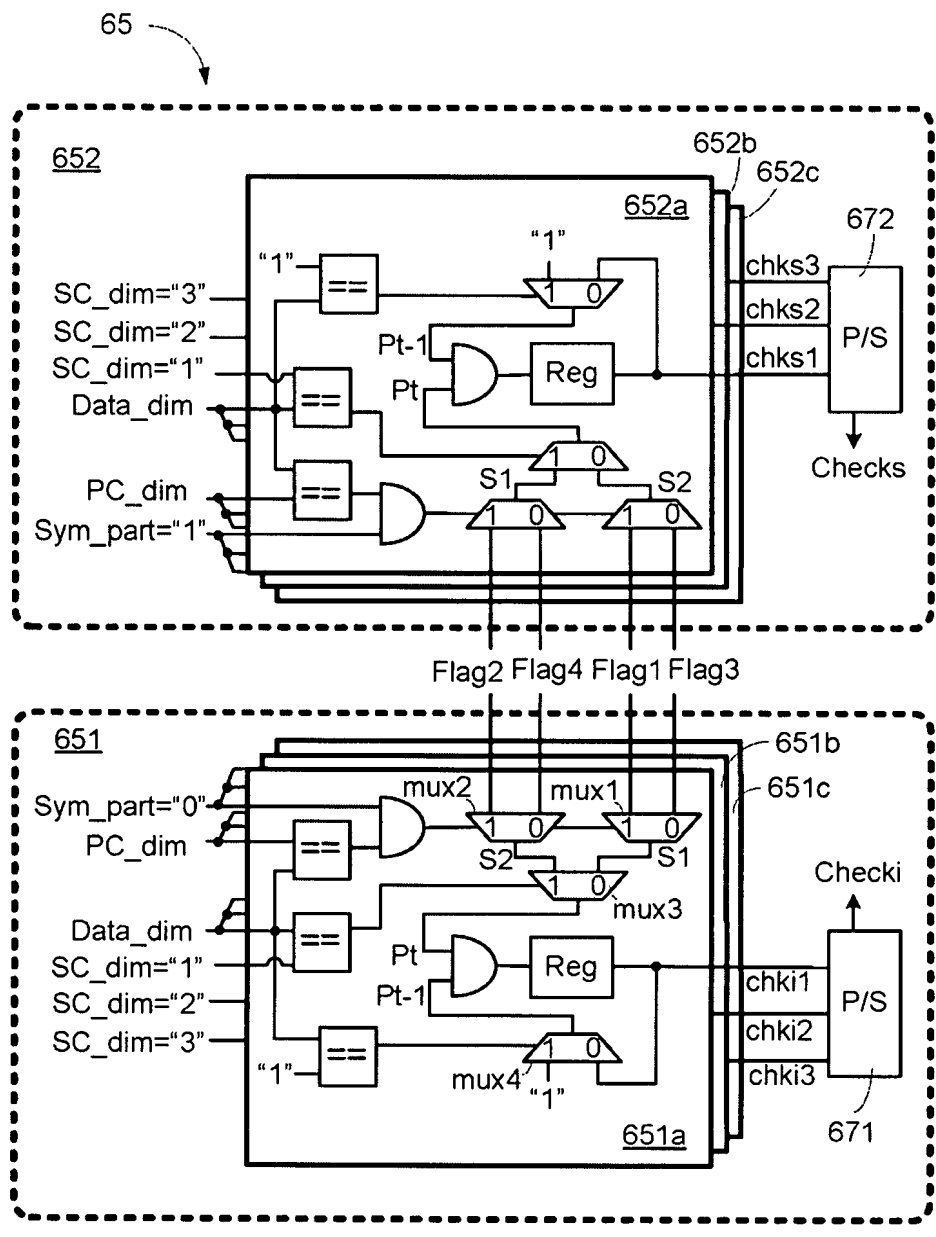
第8B圖



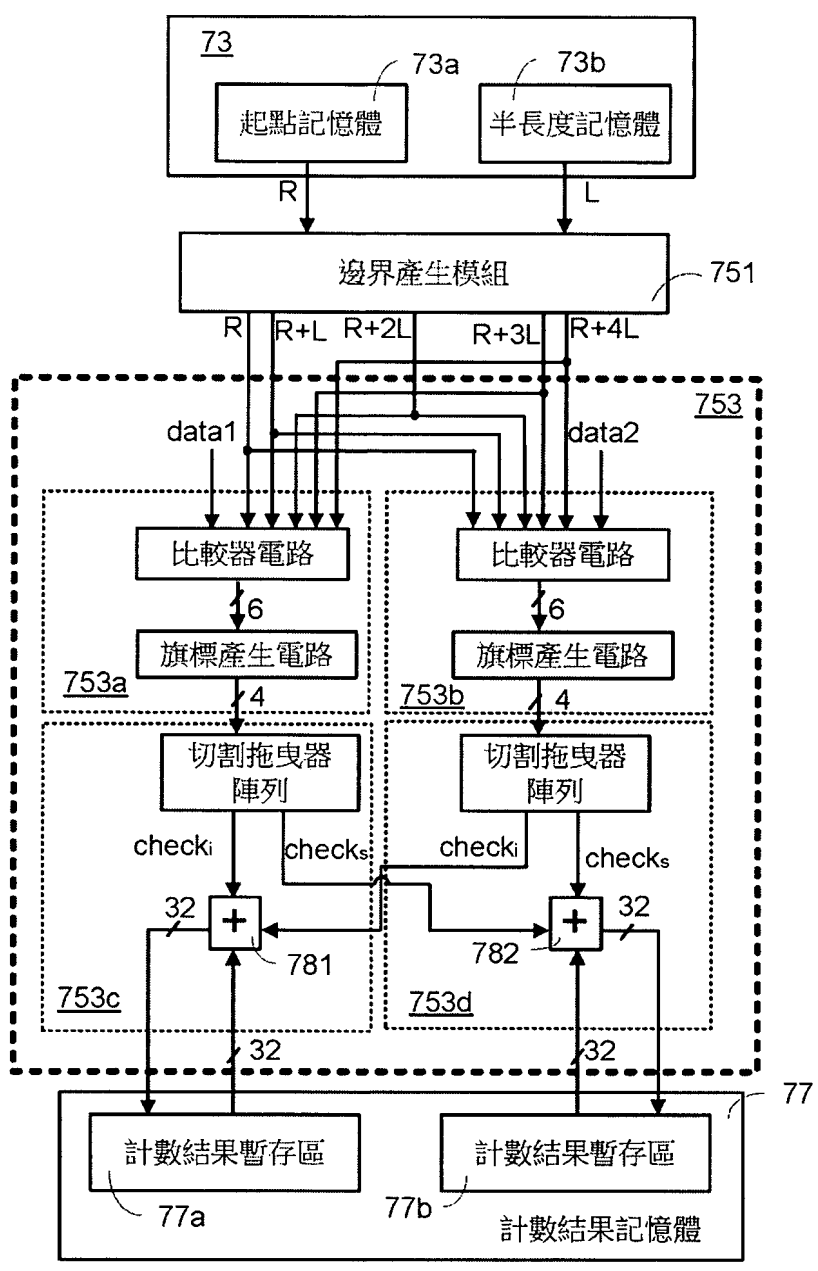
第8A圖



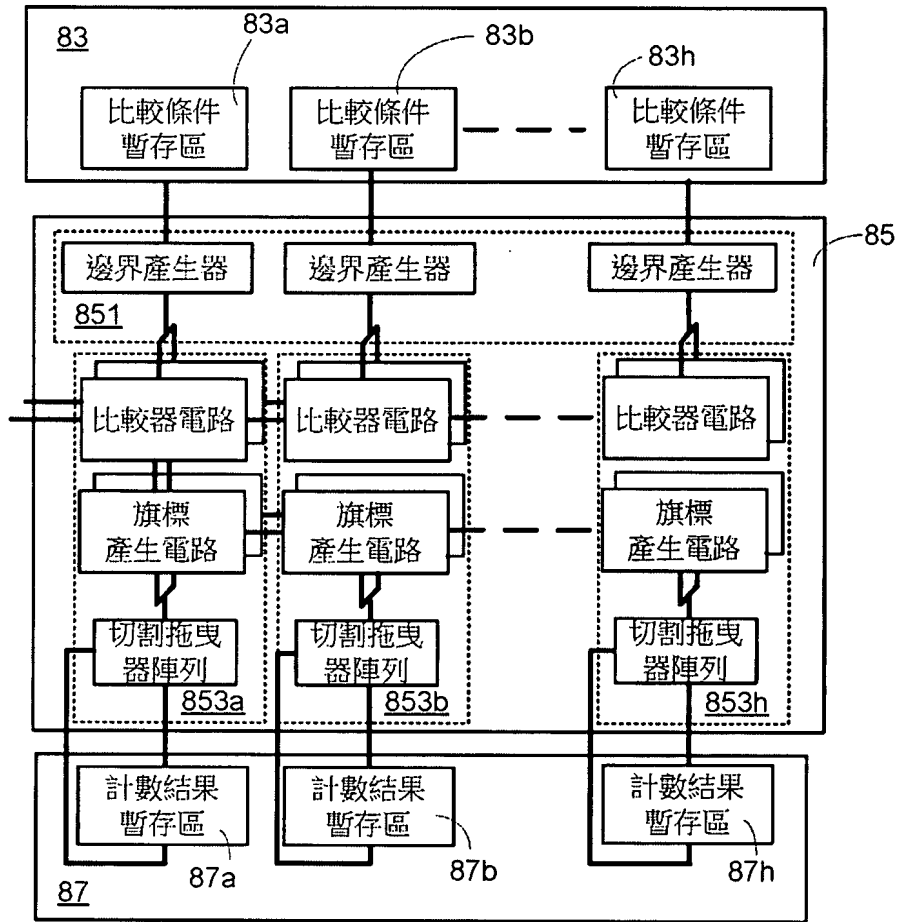
第8C圖



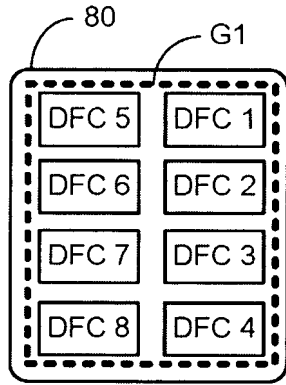
第8D圖



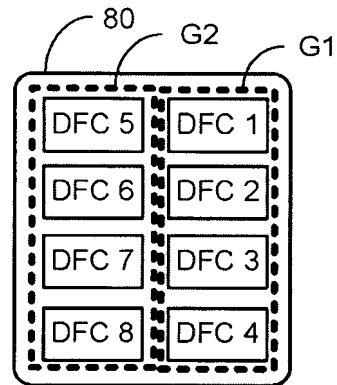
第9圖



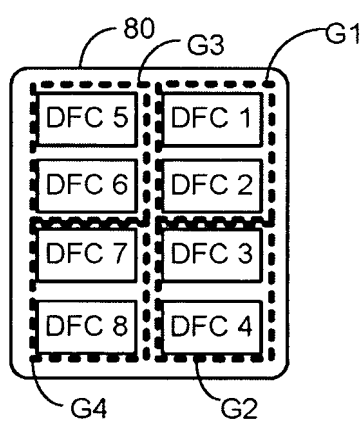
第10圖



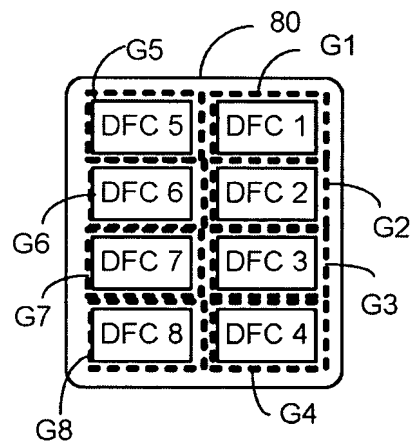
第11A圖



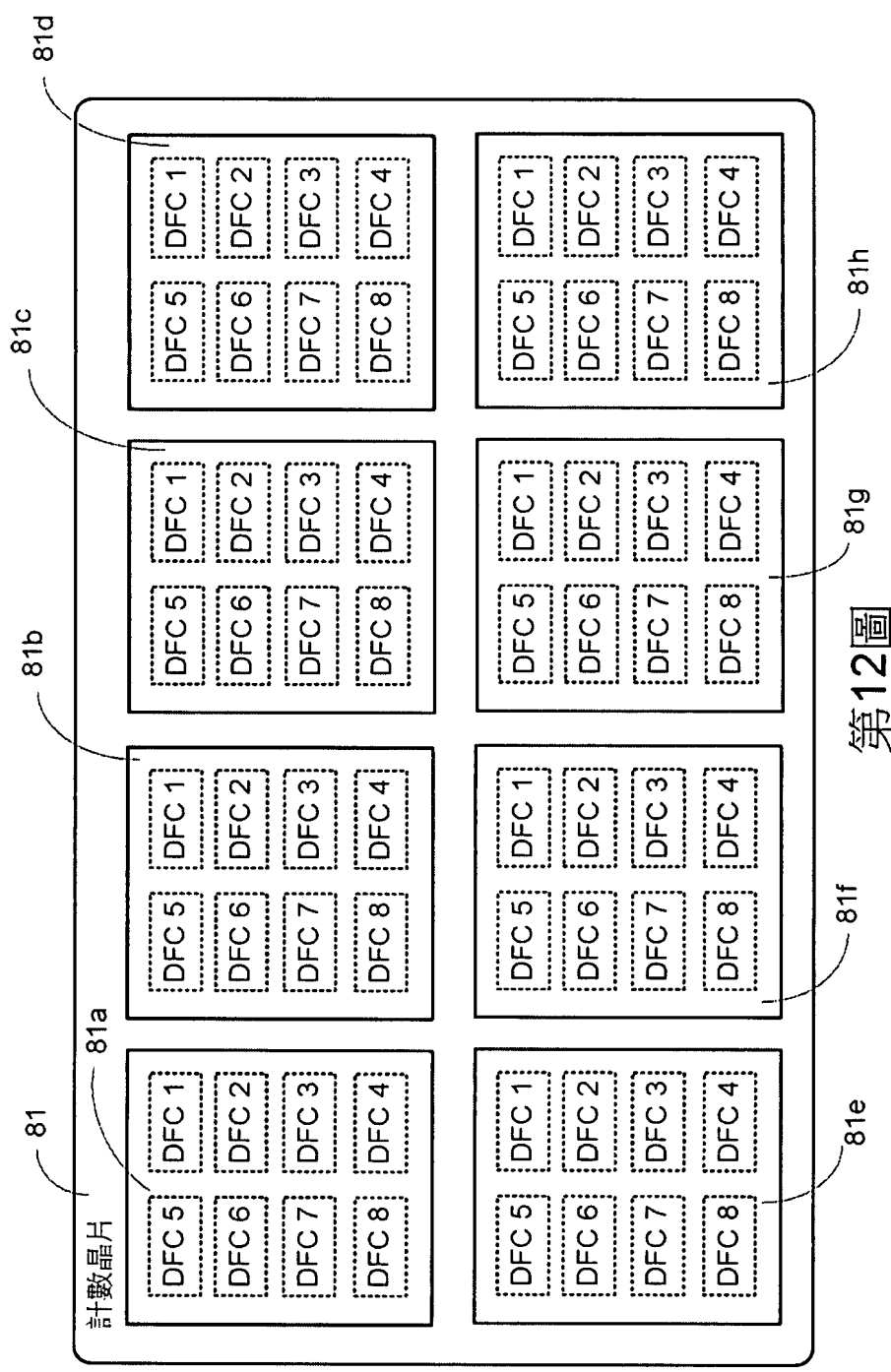
第11B圖



第11C圖

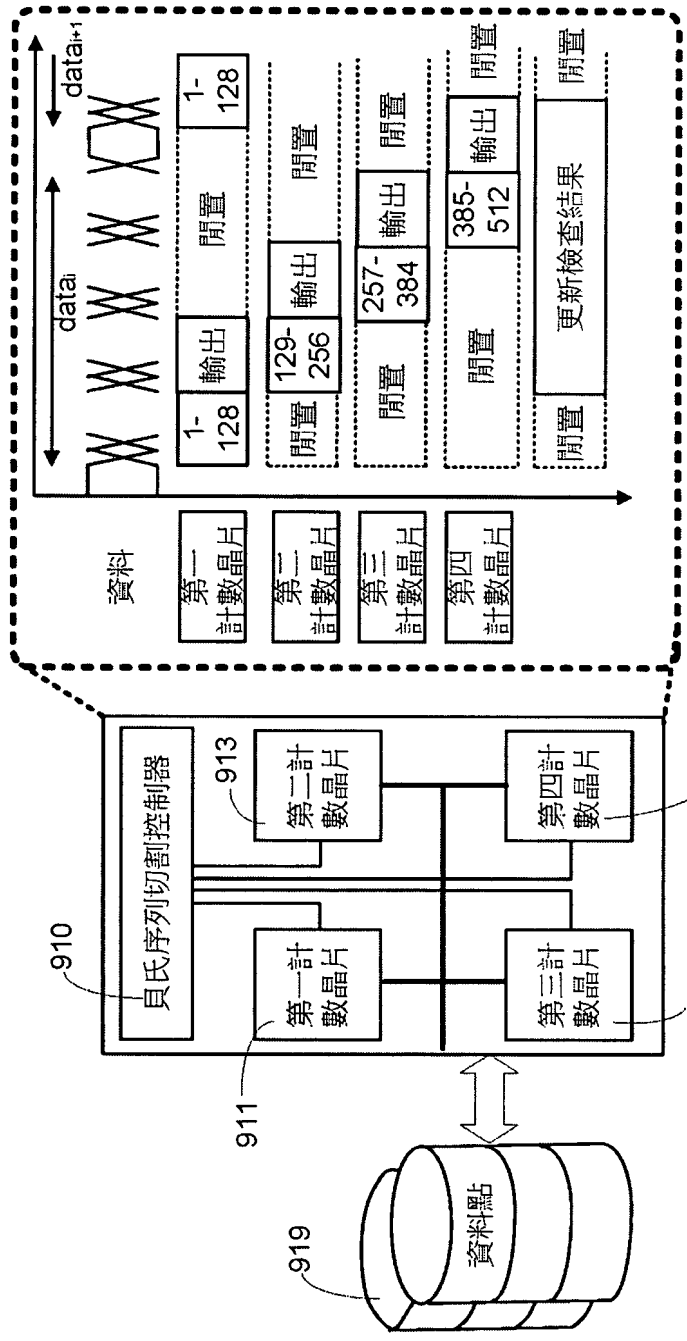


第11D圖



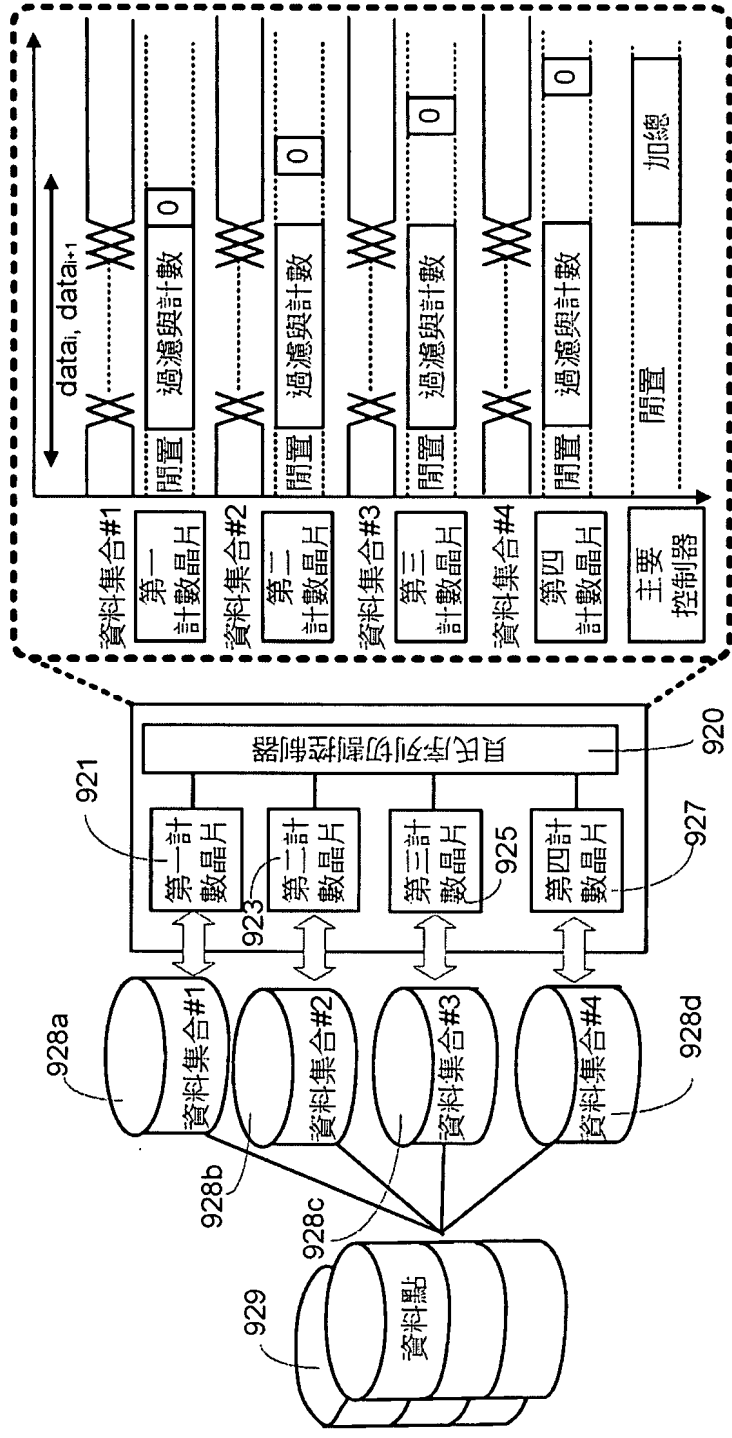
第12圖





第13圖





第14圖

