



(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 I474686 B

(45) 公告日：中華民國 104 (2015) 年 02 月 21 日

(21) 申請案號：101146838

(22) 申請日：中華民國 101 (2012) 年 12 月 12 日

(51) Int. Cl. : *H04L12/931 (2013.01)**H04L12/741 (2013.01)**H04L29/06 (2006.01)**G11C15/00 (2006.01)*

(30) 優先權：2012/06/28 美國

US 13/535,984

(71) 申請人：美國博通公司 (美國) BROADCOM CORPORATION (US)

美國

(72) 發明人：孫岩 SUN, YAN (CN)

(74) 代理人：莊志強

(56) 參考文獻：

US 2007/0130439A1

US 2011/0007565A1

審查人員：陳哲賢

申請專利範圍項數：10 項 圖式數：7 共 33 頁

(54) 名稱

用於在記憶體中儲存整數值的範圍的方法及其系統

SYSTEM AND METHOD FOR STORING INTEGER RANGES IN A MEMORY

(57) 摘要

公開了涉及在記憶體中儲存整數範圍的系統和方法。在記憶體中儲存整數值範圍包括：確定該範圍中的子範圍，使得在來自分別是子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，第一多個位元子序列中除了至多一個位元子序列外的所有位元子序列的值都等於第二多個位元子序列中的相應位元子序列或者值為 0，並且第二多個位元子序列的相應位元子序列具有最大值。在記憶體中儲存整數值的範圍進一步包括基於第一和第二多個位元子序列的值形成第一位元串，並且在記憶體中儲存第一位元串。

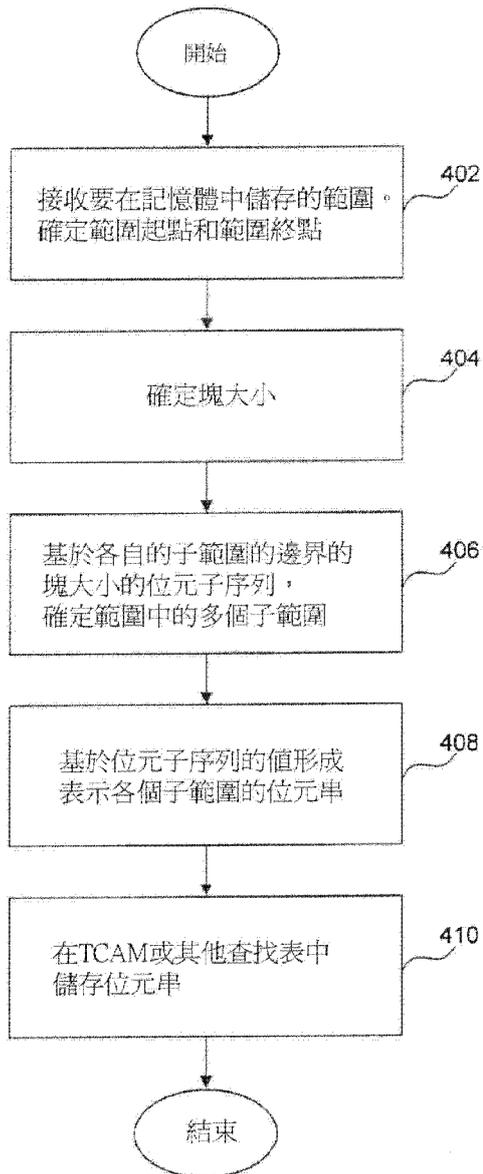


圖 4



發明摘要

※ 申請案號：101146838

※ 申請日：101.12.12

※ IPC 分類：H04L 12/931

12/741

29106

G11C 15/00

(2013.01)

(2013.01)

(2006.01)

(2006.01)

【發明名稱】(中文/英文)

用於在記憶體中儲存整數值的範圍的方法及其系統

SYSTEM AND METHOD FOR STORING INTEGER RANGES

IN A MEMORY

【中文】

公開了涉及在記憶體中儲存整數範圍的系統和方法。在記憶體中儲存整數值範圍包括：確定該範圍中的子範圍，使得在來自分別是子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，第一多個位元子序列中除了至多一個位元子序列外的所有位元子序列的值都等於第二多個位元子序列中的相應位元子序列或者值為 0，並且第二多個位元子序列的相應位元子序列具有最大值。在記憶體中儲存整數值的範圍進一步包括基於第一和第二多個位元子序列的值形成第一位元串，並且在記憶體中儲存第一位元串。

【英文】

【代表圖】

【本案指定代表圖】：圖 4。

【本代表圖之符號簡單說明】：

方法 400

步驟 402～410

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

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)

用於在記憶體中儲存整數值的範圍的方法及其系統

SYSTEM AND METHOD FOR STORING INTEGER RANGES
IN A MEMORY

【技術領域】

本發明的實施方式涉及在記憶體中儲存值。

【先前技術】

諸如路由器、交換機、橋接器(bridge)等的封包(packet)處理裝置需要適應逐漸增大的頻寬和處理需求。例如，頻寬的增大要求以接近網速在封包處理裝置中處理每個封包，同時，進入封包處理裝置的封包數量變大。此外，複雜的封包處理可能要求封包處理裝置將封包標頭中的較大欄位組與預配置的值相比較。

在許多封包處理裝置中，將封包標題欄位與三態內容存取記憶體(TCAM)中的條目(entry)相比較。TCAM是其中可使用二進位(例如1和0)位元和通用憑證位元來儲存值的條目表。“通用憑證(wildcard)”表示對應的位元可以是1或0。在現有技術中已知一些儲存1、0、和通用憑證的具體技術，並且可能依賴於實施方案。TCAM中的每個條目都可以對應於一個或多個欄位。可以將入站封包並行地與TCAM中的所有條目相比較。在基於入站封包欄位執行查找時，也可以使用其他記憶體資料結構。

檢查的封包標題欄位可以包括網際網路協定(IP)位址、埠值、協定值、虛擬區域網路識別字(VLANID)和其他標頭欄位。將部分標頭欄位與具體值相比較。可以將其他標頭欄位與值的範圍相比較。常常使用共同字首(common prefix)來表示IP位址的範圍。“字首”是在值的二進位表示的開頭開始的一個或多個位元的

序列。如這裡所使用的，如果兩個或更多值中的每個的二進位表示具有從彼此共同的從第一位元開始的一個或多個位元的序列，則該兩個或多個值被認為是具有共同字首。然而，使用字首可能不能有效地表示整數值的範圍（例如用於在封包標頭中表示埠值、協定識別字等的那些）。例如，對值的範圍進行儲存可能需要相對大量的字首，並且每個字首可能需要 TCAM 中的多個條目。例如，範圍的每個字首可能需要 TCAM 中的多個條目，以在其他欄位的組合中進行比較。

因此，在 TCAM 和記憶體中的其他查找結構中所需的儲存值的範圍的條目數可能過大。由於其速度和靈活性，TCAM 經常用於查找操作。然而，由於 TCAM 的相對高功耗和成本，需要減小 TCAM 的大小。其他記憶體也可以用於查找，並且可以因為在查找表中具有較少的條目而帶來好處。

【發明內容】

本發明實施方式涉及在 TCAM 或其他記憶體中有效地儲存值範圍的方法、系統和電腦可讀儲存介質。在記憶體中儲存整數值的範圍包括確定該範圍中的子範圍，使得在來自分別是子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列（bit subsequence）中，第一多個位元子序列中的除了至多一個位元子序列以外的所有位元子序列的值要麼等於第二多個中的對應位元子序列要麼值為 0 且第二多個位元子序列中的對應位元子序列具有最大值。在記憶體中儲存整數值的範圍進一步包括基於第一和第二多個位元子序列的值形成第一位元串（bit string），並且在記憶體中儲存第一位元串。

本發明的一個方面涉及一種用於在記憶體中儲存整數值的範圍的方法，包括：確定所述範圍中的子範圍，使得在來自分別是所述子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，在第一多個位元子序列中除了至多一個位元子序列

外的所有位元子序列的值等於第二多個位元子序列中的相應位元子序列的值或者值為 0 且第二多個位元子序列的相應位元子序列具有最大值；基於所述第一和第二多個位元子序列的值形成第一位元串；以及在所述記憶體中儲存所述第一位元串。

在上述方法中，優選所述儲存包括在三態內容定址記憶體（TCAM）中儲存所述第一位元串。

在上述方法中，優選所述形成包括：產生與來自所述第一多個位元子序列的位元子序列與來自所述第二多個位元子序列的相應位元子序列的各個配對相應的各自的位元子串；以及連接（concatenate）所產生的位元子串以產生所述位元串。

在上述方法中，優選各個所述各自的位元子串的長度至少等於來自所述第一多個位元子序列的位元子序列的最大值。

在上述方法中，優選產生各自的位元子串包括：確定與所述各個配對相應的有效值的範圍；以及基於所確定的有效值的範圍來設置所述位元子串中的位元值，其中，所述位元子串中的各自的位元位置與所確定的有效值的範圍中的值相關聯。

在上述方法中，優選在所述位元子串中設置位元值包括：將所述位元子串的前 m 個位元設置為 1，其中， m 是來自所述第一多個位元子序列的位元子序列的有效值；以及將所述前 m 個位元後的 n 個位元設置為通用憑證，其中， n 是來自所述第一多個位元子序列的位元子序列和來自所述第二多個位元子序列的位元子序列之間的差。

在上述方法中，優選還包括：從資料封包的欄位值的二進位表示確定第三多個位元子序列；基於所述第三多個位元子序列的值形成第二位元串；將所述第二位元串與所儲存的第一位元串相比較；以及基於所述比較來處理所述資料封包。

在上述方法中，優選根據預定的位元子序列長度來確定所述第一、第二和第三多個位元子序列。

在上述方法中，優選所述第一、第二和第三多個位元子序列中的位元子序列數量（a number of）相等。

在上述方法中，優選所述形成第二位元串包括：產生與來自所述第三多個位元子序列的各個位元子序列相對應的各自的第二位元子串；以及連接所述第二位元子串以產生所述第二位元串。

在上述方法中，優選產生第二位元子串包括：確定來自所述第三多個位元子序列的相應位元子序列的值；以及基於所確定的值來設置所述第二位元子串中的位元值，其中，所述第二位元子串中的各個位元位置與各自的確定值相關聯。

在上述方法中，優選設置所述第二位元子串中的位元值包括：將所述第二位元子串的前 m 個位元設置為 1，其中， m 是來自所述第三多個位元子序列的位元子序列的值。

在上述方法中，優選還包括：基於記憶體特性，配置所述第一和第二多個位元子序列的位元子序列的長度或者應用於所述第一位元子串的位元子串的長度中的一個或多個。

在上述方法中，優選還包括：基於位元串的預定長度，配置所述第一和第二多個位元子序列的位元子序列的長度，其中，所述預定長度被應用於所述第一位元串。

在上述方法中，優選還包括：基於所述範圍的大小，配置所述第一和第二多個位元子序列的位元子序列的長度。

本發明的又一個方面涉及一種系統，包括：處理器；耦接至所述處理器的至少一個記憶體；記憶體儲存模組，被配置為：確定範圍中的子範圍，使得在來自分別是所述子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，在第一多個位元子序列中除了至多一個位元子序列外的所有位元子序列的值等於第二多個位元子序列中的相應位元子序列的值或者值為 0 且第二多個位元子序列的相應位元子序列具有最大值；基於第一和第二位元子序列的值形成第一位元串；以及在所述至少一個記憶

體中儲存所述第一位元串。

在上述系統中，優選所述至少一個記憶體包括三態內容定址記憶體（TCAM）。

還包括：封包欄位比較模組，被配置為：從資料封包的欄位值的二進位表示確定第三多個位元子序列；基於所述第三多個位元子序列的值形成第二位元串；將所述第二位元串與儲存的第一位元串相比較。

還包括：封包處理模組，被配置為基於所述比較來處理所述資料封包。

本發明的又一方面涉及一種記錄了指令的電腦可讀儲存介質，當處理器執行指令時，使得處理器執行用於在記憶體中儲存比較值的範圍的方法，包括：確定所述範圍中的子範圍，使得在來自分別是所述子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，在第一多個位元子序列中除了至多一個位元子序列外的所有位元子序列的值等於第二多個位元子序列中的相應位元子序列的值或者值為 0 且第二多個位元子序列的相應位元子序列具有最大值；基於所述第一和第二多個位元子序列的值形成第一位元串；以及在所述記憶體中儲存所述第一位元串。

以下結合附圖詳細描述了本公開的其他特徵和優點以及本公開的各實施方式的結構和操作。值得注意的是，本公開不限於這裡描述的具體實施方式。這裡給出了這種實施方式僅用於說明目的。基於本文包含的教導，對於該發明所屬領域中具有通常知識者來說，其他實施方式將是顯而易見的。

【圖式簡單說明】

圖 1A 顯示了在記憶體中儲存整數值範圍的範例性傳統方法。

圖 1B 顯示了範例性封包格式。

圖 2A 和圖 2B 顯示了根據本公開實施方式的表示範圍的記憶體中的範例性範圍和條目。

圖 3 顯示了根據本公開實施方式的用於在記憶體中儲存整數值範圍的範例性系統。

圖 4 顯示了根據本公開實施方式的用於在記憶體中儲存整數值範圍的方法的流程圖。

圖 5 顯示了根據本公開實施方式的用於將輸入欄位和整數值的範圍相比較的方法的流程圖。

圖 6 顯示了根據實施方式的用於確定範圍中的子範圍的方法的流程圖。

圖 7 顯示了根據實施方式的用於對子範圍編碼的方法的流程圖。

【實施方式】

雖然這裡參照用於特定應用的說明性實施方式描述了本公開，但是應當理解，本公開不限於此。該發明所屬領域中具有通常知識者根據本文的教導將認識到在其範圍內的其他修改、應用和實施方式以及本公開將具有顯著實用性的其他領域。

本說明書中公開的實施方式提供了在 TCAM 和其他記憶體結構中對整數範圍的更有效的儲存。例如，實施方式使得能夠使用比傳統技術所需的顯著減少的條目在 TCAM 中儲存整數範圍。使用較少的條目來儲存範圍可以使得用於條目表的 TCAM 或其他記憶體的成成本較低、查找效率提高以及能夠比較更多欄位。

圖 1A 顯示了在諸如 TCAM 或其他記憶體的記憶體中儲存整數值範圍的範例性傳統方法。如圖 1A 所示，當使用儲存基於字首的範圍的傳統方法時，必須使用至少六個字首來覆蓋整數 1 至 24 的範圍。如 101 項所示，範圍 1 至 24 中的各個值以二進位顯示。基於數位的二進位表示中的共同字首，可以確定六個子範圍。這六個子範圍是 1、2 至 3、4 至 7、8 至 15、16 至 23 和 24。105 項顯示了可以用於表示 TCAM 條目中的範圍的基於字首的欄位值的最小組。如圖所示，六個單獨字首需要表示範圍 1 至 24，即，

“00001”、“0001”、“001”、“01”、“10”和“11000”。

圖 1B 顯示了可以在封包處理裝置中處理的資料封包 110 的範例性格式。資料封包 110 可以包括標頭欄位 (header field) 部分 112 和資料部分 114。標頭欄位 112 包括諸如源位址 (source address)、目的地址、源埠、目的埠、協定類型 116、以及虛擬區域網路識別字 (VLAN, 未顯示) 的一個或多個子段。資料部分 114 可以包括實際資料有效負載, 或者實際資料有效負載封裝在一個或多個其他協定封包格式中。可以根據一個或多個通訊協定來將封包格式化, 通訊協定例如但是不限於網際網路協定 (IP)、傳輸控制協定 (TCP)、用戶資料報協定 (UDP)、乙太網路等。在顯示的例子中, 源位址和目的地址可以是 IP 位址。至少部分地由於 IP 位址的層次結構, IP 位址的範圍可以有效地以字首的形式來表示。整數值範圍, 諸如源埠、目的埠、和協定類型 116 欄位可以與其比較的那些整數值範圍, 也可以表示為如圖 1A 所示的基於字首的欄位。然而, 如結合圖 1A 所描述的, 整數值的基於字首的表示可能需要非常多的 TCAM 條目, 浪費記憶體空間和功率。這尤其如此是因為每個字首都可以與其他待比較的欄位組合來組合。在本說明書中公開的實施方式使得能夠儲存整數值, 諸如但是不限於用於比較資料封包 110 中的埠和協定類型 116 欄位的值的範圍。

圖 2A 和圖 2B 顯示了根據本公開實施方式的表示範圍的 TCAM 或其他記憶體中的範例性範圍和條目。

圖 2A 顯示了根據實施方式在 TCAM 中儲存整數 11 至 54 的範圍。使用 8 位元寬度二進位位元序列 202 來指定範圍 (例如 11 至 54) 的邊界 (例如開始和結束) 點的二進位表示。每個二進位位元序列 202 都被分成二進位位元子序列 204 的 3 位元寬的塊 (chunk)。3 位元的塊大小可以是可配置的或者可以動態地確定。這樣, 範圍的邊界點表示為元素的序列 205, 稱為值序列, 包括由

3 位元塊表示的對應確定值。序列中的每個元素都對應於塊。為了解釋和說明的目的，值序列作為十進位值序列顯示，例如，(0 1 3) 和 (0 6 6)。所顯示的塊大小不是限制性的。

如覆蓋範圍 11 至 54 的三個子範圍 210 所示，該範圍然後被分成若干子範圍。產生子範圍（例如 [0 1 3] 至 [0 1 7]、[0 2 0] 至 [0 5 7]、[0 6 0] 至 [0 6 6]），使得當子範圍表示為對應於該子範圍的邊界點的一對值序列時，任何子範圍的邊界都符合一組預定的性質。該組預定性質包括：除了至多一對，在兩個十進位序列之間的所有對對應元素中具有匹配值。一對對應元素包括來自子範圍的各個邊界點的元素。例如，在圖 2A 中，子範圍 (0 1 3) 至 (0 1 7) 在子範圍的起點和終點之間只有第三個元素不同。當對應塊的值的整個允許範圍對特殊的元素對有效時，來自值序列的特殊元素對免除具有匹配值的性質。例如，如圖 2A 所示，在子範圍 (0 2 0) 至 (0 5 7) 中，第二和第三個元素都不同，但是第三個元素容納 (accommodate) 3 位元塊的值的整個範圍。具體地，在起點的第三個元素（例如 0）和終點的第三個元素（例如 7）之間容納可以使用 3 位元（例如整數值 0 至 7）表示的整個範圍。第三個子範圍 (0 6 0) 至 (0 6 6) 在兩個邊界點之間只有第三個元素不同。

然後，將指定為值序列的每個子範圍表示為在實施方式中儲存為 TCAM 條目 212 或其他記憶體中的條目的位元串。如圖 2A 中所示，使用來自子範圍 210 的子範圍的值序列來從條目 212 中的一個確定對應於該子範圍的條目。每個位元串 206 都包括多個位元子串 208。位元串 206 中的位元子串 208 的數量等於對應的一個十進位序列中的元素的數量。每個位元子串 208 表示位於值序列中的對應元素對的價值之間的整數值。在一個實施方式中，位元串“000 0000001 xxxx111”儲存在記憶體或 TCAM 中用於子範圍 (0 1 3) 至 (0 1 7) 的條目。位元子串中的每個位元表示兩個邊界點中的對應塊的價值之間的不同整數。例如，從左邊開始，第一位元表示

整數“1”，第二位元表示整數“2”等。在上述範例位元串中，第一位元子串是“000”，表示在兩個十進位序列的各自的第一元素之間，除了 0 之外，沒有整數。第二位元子串“0000001”表示“1”是兩個十進位序列的各自的第二元素之間的唯一整數。第三位元子串“xxxx111”表示在第三十進位序列的各自的第一元素之間的出現的整數 3 至 7。“x”是特別值（例如，通用憑證或“不關心的位元”），表示 x 出現的任何位置在位元串中可以是 0 或 1。注意，為了表示位元子串中的整數 3，將表示 3 的位元以及所有不太重要的位元設置為 1。

換種描述，每個位元子串表示對應塊的有效值的範圍。如下面的描述，入站封包的標頭欄位值用於導出塊的序列，並且每個塊與一個或多個位元串的對應位元子串相比較。例如，上述的（位元串“000 0000001 xxxx111”的）第一、第二、和第三位元子串將分別匹配具有例如 0、1、和 3 至 7（含 3 和 7）之間的任何值的塊序列中的第一、第二、和第三塊的任何封包標頭值。可以認為位元子串具有各自的塊（或位元子序列）的位元編碼的（還稱為“位元編碼的”或“逐位元編碼的”）值。

圖 2B 顯示了整數範圍 677 至 947 的儲存。待表示的欄位值是 12 位元整數，因此能夠分成四個 3 位元塊。因此，範圍（例如 001 010 100 101 至 001 110 001 011）的成塊的二進位表示產生了十進位值序列（顯示為 260 項）中的邊界點(1 2 4 5)至(1 6 1 3)。然後，基於範圍的值序列 260 導出子序列（262 項）。如上文所述，在特定子範圍的起點和終點的值序列之間，除非元素對容納對應塊的值的整個範圍（例如，3 位元的塊的值 0 至 7），否則除了來自起點值序列的至多一個元素，各個元素將與來自終點值序列的對應元素相同。例如，第一子範圍(1 2 4 5)至(1 2 4 7)僅在第四個元素不同。第二子範圍(1 2 5 0)至(1 2 7 7)在第三和第四個元素不同，但是第四個元素容納對應的 3 位元塊的值 0 至 7 的全部範圍。

第三子範圍(1 3 0 0)至(1 5 7 7)在第二、第三、和第四元素不同，但是第三和第四元素每個都容納各自的塊的全部範圍。264 項顯示了對應於子範圍 262 的 TCAM 中儲存的位元串。另一顯示的子範圍(1 6 0 0)至(1 6 0 7)僅在第四個元素不同，第一個元素也容納塊的有效值的全部範圍。

圖 3 顯示了根據本公開實施方式的用於在 TCAM 中儲存整數值範圍的範例性系統 300。系統 300 可以包括封包處理裝置、資料庫系統、或需要快速查找操作的其他裝置或系統。系統 300 包括處理器 302、查找表(lookup table)304、記憶體 306、記憶體儲存模組 310、欄位比較模組 312、處理模組 314、以及通訊基礎設施（也稱為“互接匯流排(interconnection bus)”）316。

處理器 302 可以包括任何中央處理單元（CPU）或專用處理器，例如但不限於專用積體電路（ASIC）或現場可程式閘陣列（FPGA）。處理器 302 可以執行邏輯指令，以控制系統 300 的一個或多個元件的操作。處理器 302 可以控制系統 300 中的處理，並且例如，可以執行或控制一個或多個封包處理管線（packet processing pipeline）的執行。封包處理管線每個都包括封包從進入系統 300 到退出系統 300 的處理路徑的邏輯視圖。

查找表 304 包括資料結構 320，其能夠用於將對應封包的一個輸入欄位值或一組輸入欄位值與一個或多個儲存值相比較。資料結構 320 包括多個條目 322。每個條目 322 可以包括將要與單個欄位比較的值，或者將要與多個選擇的欄位比較的值組合。根據一個實施方式，資料結構 320 是 TCAM。將輸入位元串 326 與條目 322 比較，以確定匹配 328，其被輸出到應用或其他接收實體或進程（未顯示）。根據一個實施方式，例如，在 TCAM 中，輸入位元串 326 可以同時與資料結構 320 中儲存的一個、全部、或多個條目相比較，並且可以輸出選擇的一個或多個匹配 328。根據另一實施方式，資料結構 320 被實施為諸如記憶體 306 的記憶體中的

鏈表、哈希表、或其他表格資料結構。

記憶體 306 可以包括諸如靜態隨機存取記憶體 (SRAM)、動態隨機存取記憶體 (DRAM)、FLASH 記憶體等中的一個或多個。記憶體 306 可以包括在系統 (例如封包處理裝置) 中執行的指令以及在裝置中處理封包期間的任何封包和相關資訊。在各實施方式中，記憶體 306 也可以包括諸如磁片、光碟、快閃記憶體等的永久性資料儲存介質。這種電腦可讀儲存介質可以用於儲存實現系統 300 的一個或多個部件的功能的軟體程式和/或邏輯指令。此外，記憶體 306 可以包括輸入封包 324，根據一個實施方式，欄位比較模組 312 將輸入封包 324 與查找表 304 相比較。例如，可以經由將系統 300 連接至網路 (未顯示) 的輸入介面 (未顯示) 來接收輸入封包 324。輸入封包 324 可以包括諸如圖 1B 中顯示的封包格式。

記憶體儲存模組 310 運行以在查找表 304 (諸如，例如在 TCAM 或其他記憶體中) 中儲存範圍。根據一個實施方式，記憶體儲存模組 310 可以包括以下結合圖 4、6 和 7 描述的處理邏輯。

欄位比較模組 312 運行以將一個或多個欄位與查找表中的條目相比較。根據一個實施方式，將來自入站資料封包的一個或多個選擇的欄位與 TCAM 的條目相比較。根據另一實施方式，將所選擇的欄位與記憶體 (諸如，RAM) 中的查找表的條目相比較。在另外一些實施方式中，可以將欄位與 TCAM 和記憶體結構兩者相比較。執行欄位與 TCAM 或其他查找表的比較，以確定一個或多個匹配條目。匹配可以是完全匹配或部分匹配。如本公開中其他地方的描述，條目中通用憑證的使用使得能夠確定部分匹配。根據一個實施方式，欄位比較模組可以包括以下結合圖 5 描述的處理邏輯。

處理模組 314 運行，以根據欄位比較模組 312 和/或查找表 304 的匹配輸出來處理封包。處理可以包括路由和/或轉發、頭處理、

封包分類、入侵防禦、資料庫查找、或在確定封包或其他消息所屬的類型時對資料封包或其他消息執行的任何類型的處理。

互接匯流排 316 可以包括通訊耦接封包處理裝置 300 的各模組的一個或多個互接匯流排結構。例如，互接匯流排 316 可以包括諸如先進高性能匯流排（AHB）的匯流排或其他內部部件互接機制，AHB 使用 ARM Ltd 發佈的 AMBA 規範版本 2 中定義的匯流排協定。

圖 4 顯示了根據本公開實施方式的用於在記憶體中儲存整數值的範圍的方法 400（步驟 402 至 410）的流程圖。並不是所有步驟 402 至 410 都是必需的，並且可以以不同於圖 4 中顯示的順序來進行步驟 402 至 410。在一個實施方式中，可以使用方法 400 來在將用於封包欄位查找操作的 TCAM 中儲存整數的範圍。例如，可以透過結合圖 3 描述的查找表 304 和記憶體儲存模組 310 來實施方法 400。

在步驟 402，接收將要儲存在查找表中的整數範圍。該範圍可以作為用戶手動輸入或機器產生輸入被接收。例如，該範圍可以是在系統初始化期間配置的用戶提供參數。在另一實施方式中，該範圍可以由系統基於一個或多個其他參數自動確定。例如，系統可以基於將要在入站封包中識別的協定的用戶配置選擇來自動確定整數範圍。確定範圍的邊界點，即，起點和終點。

在步驟 404，確定塊大小。術語“塊大小”是指來自範圍或子範圍的邊界點的二進位表示的連續位元組的位元大小。邊界點的二進位表示的連續位元組可以稱為位元的子序列（也稱為“位元子序列”）。圖 2A 中的 204 項是位元子序列，其還可以稱為塊。如圖 2A 所示，子序列（除了各個位元序列中具有最高有效位（most significant bit）的子序列）的長度為 3 位元。

在實施方式中，塊大小可能影響查找的各方面。例如，塊大小影響 TCAM 條目的長度。根據一個實施方式，隨著塊大小增大，

TCAM 條目的長度也增大。此外，當塊大小減小時，TCAM 條目的數量減小。實際上，塊大小決定 TCAM 條目中諸如位元子串 208 的位元子串中的位元數量。因此，可以基於考慮每個 TCAM 條目的所需大小和/或 TCAM 條目的所需數量來決定塊大小。

在步驟 406，在接收到的輸入範圍中確定多個子範圍。所確定的子範圍基於來自各自的子序列的邊界的二進位表示的塊大小的位元子序列。可以根據以下描述的方法 600 來確定子範圍。

在步驟 408，對於步驟 406 中確定的每個子範圍，產生位元串。位元串是子範圍的位元表示。更具體地，位元串包括多個位元子串，並且每個位元子串都是可以在對應塊的子範圍的邊界點之間有效存在的整數的位元表示。

在步驟 410，在 TCAM 中儲存位元串形式的所確定的子範圍。根據另一實施方式，在另一記憶體中以例如但不限於鏈表或哈希表的資料結構儲存子範圍。然後可以使用所儲存的子範圍來確定輸入封包是否與一個或多個儲存的條目相匹配。可以根據以下描述的方法 700 來執行位元串的編碼和/或到 TCAM 或其他記憶體的儲存。

雖然顯示了儲存單個欄位作為條目，但是應當注意，根據其他實施方式，TCAM 或查找表中的條目還可以包括兩個或更多欄位的組合。對於每個欄位，可以如結合圖 4 所描述的那樣來產生所儲存的位元串。

圖 5 顯示了根據本公開實施方式的用於將入站封包與整數值的範圍相比較的方法 500（步驟 502 至 510）的流程圖。並不是所有的步驟 502 至 510 都是必需的，並且步驟 502 至 510 可以以不同於圖 5 顯示的順序來進行。在一個實施方式中，方法 500 可以用於將輸入封包的一個或多個欄位與查找表相比較。例如，可以透過結合圖 3 描述的查找表 304 和欄位比較模組 312 來實施方法 500。

在步驟 502，確定來自輸入封包的所選擇的欄位的值。根據一個實施方式，輸入封包可以是透過輸入介面進入例如系統 300 的封包處理裝置的封包。輸入封包可以具有頭部分和資料部分，並且頭部分可以包括一個或多個欄位。根據一個實施方式，輸入封包可以具有與圖 1B 類似的格式。可以使用選擇的欄位將封包與 TCAM 或其他查找表相比較。確定值包括識別對應於特定所選欄位的位元。

在步驟 504，用二進位表示欄位值並且確定塊。根據一個實施方式，將來自輸入封包的欄位值的二進位串表示的長度配置為等於查找表中配置的範圍和子範圍的二進位串表示的長度。塊大小可以是配置參數。如在範圍或子範圍的邊界點的情況下所進行的，將輸入封包的所選欄位的二進位表示分組成塊大小的子序列。

在步驟 506，形成包括多個位元子串的位元串，以表示待查找的欄位值。每個位元子串(bit substring)逐位 (bit-wise) 表示來自所選欄位的二進位表示的各自的塊的值。與為 TCAM 條目配置的位元子串 (諸如圖 2A 中所示的 208 項) 類似，對於可以在塊中表示的每個不同有效整數，在表示入站封包的欄位值時使用的位元子串具有保留位元。

在步驟 508，將表示輸入封包的欄位的位元串 (“輸入欄位位元串”) 與查找表相比較。根據一個實施方式，將輸入欄位位元串與 TCAM 比較，以確定匹配。如上文所述，可以將輸入欄位位元串同時與全部或多個 TCAM 條目相比較。根據另一實施方式，可以執行與另一記憶體資料結構的比較。該比較操作產生所確定的一個或多個匹配。一個或多個匹配指示輸入欄位值是否在範圍以及匹配的子範圍內。

在步驟 510，基於在步驟 508 確定的一個或多個匹配來處理輸入封包。根據一個實施方式，在步驟 508 執行的查找確定用於封包轉發的下一跳，並且根據一個或多個匹配的 TCAM 條目來轉發

封包。其他範例性實施方式包括但不限於，選擇用於被本地應用處理的封包，執行諸如分片（fragmentation）和重組或欄位值的更改的封包處理功能、選擇待丟棄或優先化的封包等。

以上結合使用 TCAM 或其他查找表來比較入站封包的單個欄位描述了方法 500。然而，執行類型處理，以基於欄位值的任意組合來執行查找。在為每個入站封包匹配多個欄位的情況下，TCAM 或查找表中的每個條目將包括對應於若干欄位的值的連接，並且例如，對應於每個欄位的值將如圖 4 所述的那樣被產生。類似地，與查找表比較的輸入值將包括輸入封包的若干欄位值，並且每個欄位值可以以與結合圖 5 描述的類似方式來制定。

圖 6 顯示了根據一個實施方式的用於確定範圍中的子範圍的方法 600（步驟 602 至 612）的流程圖。並不是所有的步驟 602 至 612 都是必需的，並且步驟 602 至 612 可以以不同於圖 6 顯示的順序來進行。在一個實施方式中，可以使用方法 600 來確定特定範圍內的多個子範圍。例如，可以透過結合圖 3 描述的記憶體儲存模組 310 來實施方法 600。

在步驟 602，初始化子範圍識別字（“子範圍 id”）。子範圍 id 是表示特定範圍中的子範圍的順序編號的計數器。在所描述的實施方式中，子範圍 id 初始化為 0，並且從範圍的低端到高端來確定子範圍。

在步驟 604，將第一子範圍初始化到範圍的低端。如圖所示，可以透過將第一子範圍的開始邊界設置為範圍的開始邊界點（例如，範圍_起點）來執行初始化。例如，如圖 2A 所示，對於範圍 11 至 54，第一子範圍（子範圍 id 0）的開始邊界設置為範圍的開始邊界點 11。

可以反覆地執行步驟 602 至 612，直到整個範圍被子範圍覆蓋。在步驟 606，對於當前子範圍，確定結束邊界點，使得除了子範圍的開始邊界點的位元序列的至多一個元素，其餘元素都等於

結束邊界點中的對應元素。這裡提到的元素可以是邊界點的二進位表示中的位元子序列或者來自對應於邊界點的十進位序列的十進位數字。當來自兩個邊界點的對應元素對覆蓋對應塊的有效值的全部範圍時，提供了除了子範圍的開始邊界點的位元序列的至多一個元素、其餘元素都等於結束邊界點中的對應元素這一規則的例外。關於確定子範圍的進一步資訊，讀者可以參照結合圖 2A 和圖 2B 的描述。

根據一個實施方式，可以透過接收用戶輸入來執行確定子範圍結束邊界點。根據另一實施方式，可以如下所述自動執行子範圍結束邊界點的確定。

將開始邊界點的二進位表示分組成塊，並且確定每個塊的值。為了說明和討論的目的，假設將各自的塊的值確定為十進位整數。然而，在實施方式中，可以以各種其他形式來確定塊的值，包括透過保持塊的二進位形式。當以十進位來確定時，開始邊界點可以表示為十進位序列。以上描述了“十進位序列”，並且實例在圖 2A 的 205 項中顯示。在圖 2A 中，第一子範圍的開始邊界點表示為 (0 1 3)。

可以使用與開始邊界點相同數量的塊來初始化子範圍結束邊界點。

以從最小值塊（例如，最右）到最高值塊（例如，最左）的順序來考慮子範圍開始邊界點的每個塊。例如，對於圖 2A 中顯示的第一子範圍的開始邊界點 (0 1 3)，按順序考慮塊值 3、1 和 0。

如果當前開始邊界塊為 0 且在當前結束邊界塊的右邊的塊（如果有的話）為最大塊值，則將當前結束邊界塊設置為最大有效值。最大有效值是最大塊值或可分配給塊而不會超越範圍邊界的最大值。

如果當前開始邊界塊為 0 並且當前結束邊界塊是最大塊值，或者如果當前開始和結束邊界塊有相同值（例如，圖 2A 中的子範

圍(0 1 3)至(0 1 7)的第一塊)，則繼續考慮當前塊的左邊的塊並且對下一塊重複前一步驟（例如，將當前塊指標移動到其當前位置的左側）。如果當前的左邊沒有其他塊，則已經確定了子範圍。

否則，如果當前開始邊界塊和當前結束邊界塊具有不同值，則將結束邊界中當前的左邊的塊設置成與開始邊界中的對應元素的相同值。已經確定了子範圍。

在步驟 608，確定當前子範圍結束點是否是範圍的結束點。如果是，則已經確定了所有子範圍，並且已經完成了方法 600 的處理。

如果不是，則在步驟 610，將子範圍識別字遞增，以開始確定下一子範圍。實際上，這意味著對下一步驟中提及當前子範圍是指接下來要確定的子範圍，即，對應於新的子範圍識別字的子範圍。

然後，在步驟 612，當前子範圍起點確定為前一子範圍結束點之後的下一整數。根據一個實施方式，透過向前一子範圍終點加 1 來確定當前子範圍起點。

在步驟 612 之後，方法 600 的處理前進到步驟 606，以確定當前子範圍終點。

圖 7 顯示了根據實施方式的用於對子範圍編碼的方法 700(步驟 702 至 714)的流程圖。並不是所有的步驟 702 至 714 都是必需的，並且步驟 702 至 714 可以以不同於圖 7 顯示的順序來進行。在一個實施方式中，可以使用方法 700 對 TCAM 條目編碼。例如，可以透過結合圖 3 描述的查找表 304 和記憶體儲存模組 310 來實施方法 700。

在步驟 702，選擇待編碼的子範圍。例如，可能已經根據圖 6 確定了子範圍。

在步驟 704，選擇子範圍的塊。根據一個實施方式，與子範圍的邊界點對應的十進位序列的各自的元素可以作為塊被存取。根

據另一實施方式，與子範圍的邊界點對應的位元序列的各自的位元子序列可以作為塊被存取。重複步驟 704 至 714，直到所選擇的子範圍的所有塊都儲存在 TCAM 或其他查找表資料結構中。根據一個實施方式，從最右側的塊（例如，具有最低有效位（least significant bit）的塊）開始，順序選擇來自位元串的塊用於使用步驟 704 至 714 的處理。

在步驟 706，確定當前塊在子範圍開始序列和子範圍結束序列兩者中是否具有相同值。如果是（即，兩個序列中具有相同值），則在步驟 708，在位元串中將塊值編碼。逐位元執行編碼，例如，對應於當前塊的（位元串的）位元子串具有用於塊的每個有效值的保留位元。根據一個實施方式，透過對應於待編碼的值的保留位元和保留位元右邊的所有其他位（例如，具有更低值的位元）設置為“1”，來執行位元子串的編碼。例如，如圖 2A 所示，將子範圍(0 6 0)至(0 6 6)中的第二元素（塊）編碼產生位元子串“0111111”，將六個最高有效位設置為“1”並且其餘位元設置為“0”。

如果在步驟 706 確定開始和結束邊界點中的當前塊值不同，則方法 700 的處理進行到步驟 710。在步驟 710，確定當前塊的有效值是否包括該塊的全部範圍。根據一個實施方式，確定開始邊界點的 0 當前塊值和結束邊界點的最大塊值（例如，塊中的所有位元為 1），以指示當前塊的有效值包括該塊的全部範圍。

如果是（例如，當前塊的有效值包括塊的全部範圍），則在步驟 714，將塊編碼為通用憑證。例如，在圖 2A 中，將子範圍(0 2 0)至(0 5 7)的第三元素編碼為“xxxxxxx”，表示 7 個位元的任一個可以是“1”或“0”。

如果不是（例如，當前塊的有效值不包括塊的全部範圍），則當前塊的有效值包括值的範圍。因此，將對應的位元子串編碼，以容納開始子範圍和結束子範圍之間的當前塊的有效值的整個範圍。例如，如圖 2A 所示，將子範圍(0 1 3)至(0 1 7)中的第三元素

(塊) 編碼產生了位元子串“xxxx111”：透過將三個最低有效位設置為“1”，記錄了塊的最低有效值 3，並且將位元 4 至 7 設置為通用憑證，以指示這些位元中的每個可以是“1”或“0”。

如上所述，重複步驟 704 至 714，以對每個塊編碼。各自的塊的編碼導致 TCAM 或記憶體中的其他資料中儲存的條目。

本文描述的通訊裝置的代表性功能可以在硬體、軟體、或他們的一些組合中來實現。例如，基於本文中給出的討論，該發明所屬領域中具有通常知識者將理解，可以使用電腦處理器、電腦邏輯、ASIC、FPGA、DSP 等來實施處理 400、500、600 和 700。因此，執行本文描述的處理功能和/或演算法的任何處理器都在本公開的範圍和精神內。

應當理解，將使用詳細說明書部分而非摘要部分來解釋申請專利範圍。摘要部分可能闡述了發明人所設想的本公開的一個或多個而非全部的範例性實施方式，因此，不以任何方式限制本公開和所附的申請專利範圍。

以上已經借助於顯示實現具體功能及其關係的功能構造塊來描述了本公開。為了描述的方便，本文任意地限定了這些功能構造塊的邊界。只要能適當地執行具體功能及其關係，也可以限定其他邊界。

具體實施方式的上述描述完全揭示了本公開的總體性質，並且透過應用本領域的技術知識，在不背離本公開的一般概念的情況下，無需過分實驗，其他人就能夠容易地對具體實施方式進行修改和/或調整用於各種應用。因此，基於本文給出的教導和引導，這種適應性修改和調整也在所公開的實施方式的等同方案的含義和範圍內。應當理解，本文的措辭和術語用於描述性而非限制性的，本說明書的措辭和術語將由該發明所屬領域中具有通常知識者根據本文的教導和指引來解釋。

本發明的外延和範圍不應由任何上述範例性實施方式限制，

而是應當僅根據申請專利範圍及其等同方案來限定。

【符號說明】

項 101、105
資料封包 110
標頭欄位部分(標頭欄位)112
資料部分(資料)114
協定類型 116
位元序列 202
位元子序列 204
序列 205
位元串 206、264
位元子串 208
子範圍 210
TCAM 條目 212
十進位值序列 260
子序列 262
系統 300
處理器 302
查找表 304
記憶體 306
記憶體儲存模組 310
欄位比較模組 312
處理模組 314
通訊基礎設施 316
資料結構 320
條目 322
輸入封包 324
輸入位元串 326

匹配輸出 328

方法 400、500、600、700

步驟 402~410、502~510、602~612、702~714

申請專利範圍

1. 一種用於在記憶體中儲存整數值的範圍的方法，包括：

確定所述範圍中的子範圍，使得在來自分別是所述子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，在第一多個位元子序列中除了至多一個位元子序列外的所有位元子序列的值等於第二多個位元子序列中的相應位元子序列的值或者值為 0 且第二多個位元子序列的相應位元子序列具有最大值；

基於所述第一和第二多個位元子序列的值形成第一位元串；以及

在所述記憶體中儲存所述第一位元串。
2. 根據申請專利範圍第 1 項所述的方法，其中，所述儲存包括在三態內容定址記憶體中儲存所述第一位元串。
3. 根據申請專利範圍第 1 項所述的方法，其中，所述形成包括：

產生與來自所述第一多個位元子序列的位元子序列和來自所述第二多個位元子序列的相應位元子序列的各個配對相應的各自的位元子串；以及

連接所產生的位元子串以產生所述位元串。
4. 根據申請專利範圍第 3 項所述的方法，其中，各個所述各自的位元子串的長度至少等於來自所述第一多個位元子序列的位元子序列的最大值。
5. 根據申請專利範圍第 3 項所述的方法，其中，產生各自的位元子串包括：

確定與所述各個配對相應的有效值的範圍；以及

基於所確定的有效值的範圍來設置所述位元子串中的位元值，其中，所述位元子串中的各自的位元位置與所確定的有效值的範圍中的值相關聯。
6. 根據申請專利範圍第 5 項所述的方法，其中，在所述位元子串

中設置位元值包括：

將所述位元子串的前 m 個位元設置為 1，其中， m 是來自所述第一多個位元子序列的位元子序列的有效值；以及

將所述前 m 個位元後的 n 個位元設置為通用憑證，其中， n 是來自所述第一多個位元子序列的位元子序列和來自所述第二多個位元子序列的位元子序列之間的差。

7. 根據申請專利範圍第 1 項所述的方法，還包括：

從資料封包的欄位值的二進位表示確定第三多個位元子序列；

基於所述第三多個位元子序列的值形成第二位元串；

將所述第二位元串與所儲存的第一位元串相比較；以及

基於所述比較來處理所述資料封包。

8. 根據申請專利範圍第 7 項所述的方法，其中，根據預定的位元子序列長度來確定所述第一、第二和第三多個位元子序列。
9. 根據申請專利範圍第 7 項所述的方法，其中，所述第一、第二和第三多個位元子序列中的位元子序列數量相等。
10. 一種用於在記憶體中儲存整數值的範圍的系統，包括：

處理器；

耦接至所述處理器的至少一個記憶體；

記憶體儲存模組，被配置為：

確定範圍中的子範圍，使得在來自分別是所述子範圍的開始值和結束值的二進位表示的第一和第二多個位元子序列中，在第一多個位元子序列中除了至多一個位元子序列外的所有位元子序列的值等於第二多個位元子序列中的相應位元子序列的值或者值為 0 且第二多個位元子序列的相應位元子序列具有最大值；

基於第一和第二位元子序列的值形成第一位元串；以及在所述至少一個記憶體中儲存所述第一位元串。

圖式

	二進制	十進制
	00001	= 1
	00010	= 2
	00011	= 3
	00100	= 4
	00101	= 5
	00110	= 6
	00111	= 7
	01000	= 8
	01001	= 9
	01010	= 10
	01011	= 11
	01100	= 12
	01101	= 13
	01110	= 14
	01111	= 15
	10000	= 16
	10001	= 17
	10010	= 18
	10011	= 19
	10100	= 20
	10101	= 21
	10110	= 22
	10111	= 23
	11000	= 24

要儲存的範圍:	1-24	
基於字首的條目:		
00001:	1	}
0001x:	2-3	
001xx:	4-7	
01xxx:	8-15	
10xxx:	16-23	
11000:	24	
		105

圖 1A

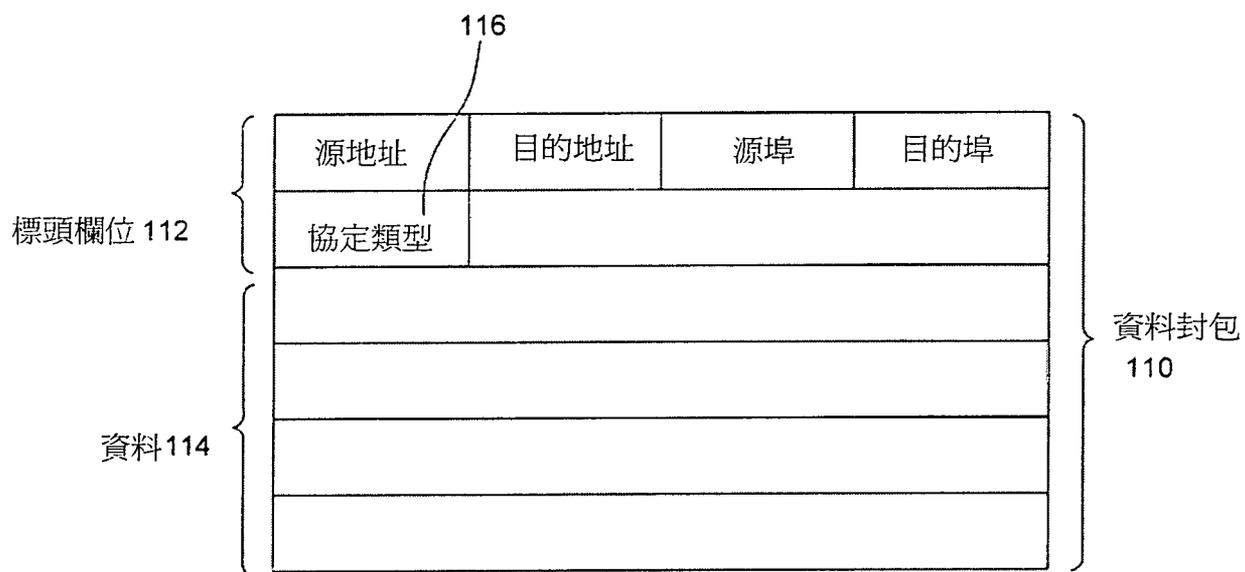


圖 1B

實例 1:
 設寬度 = 8, 範圍 = [11, 54], 塊大小 = 3
 a = 11: 00 001 011 (0 1 3) ~ 值 (十進制) 序列 205
 b = 54: 00 110 110 (0 6 6) ~ 值 (十進制) 序列 205

子範圍	TCAM 條目	
(0 1 3) — (0 1 7)	(000 0000001 xxxx111)	} 212
(0 2 0) — (0 5 7)	(000 00xxx11 xxxxxxx)	
(0 6 0) — (0 6 6)	(000 0111111 0xxxxxx)	

圖 2A

實例 2:
 設寬度 = 12, 塊大小 = 3
 a = (1 2 4 5) ~ 260
 b = (1 6 1 3) ~ 260

範圍和 TCAM 條目:

(1 2 4 5) — (1 2 4 7)	0000001 0000011 0001111 xx11111
(1 2 5 0) — (1 2 7 7)	0000001 0000011 0011111 xxxxxxx
(1 3 0 0) — (1 5 7 7)	0000001 00xx111 xxxxxxx xxxxxxx
(1 6 0 0) — (1 6 0 7)	0000001 0111111 0000000 xxxxxxx
(1 6 1 0) — (1 6 1 3)	0000001 0111111 0000001 1111xxx

圖 2B

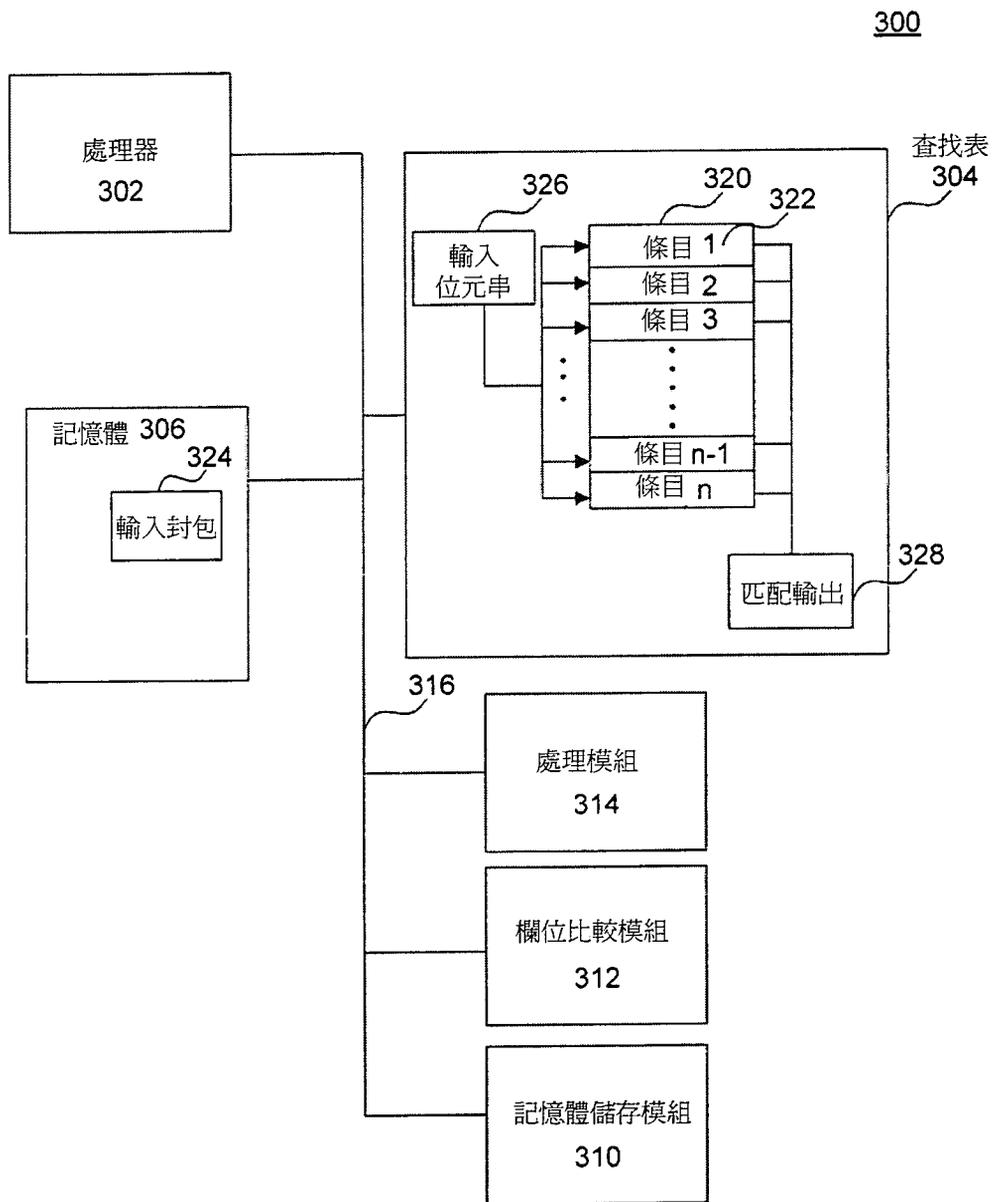


圖 3

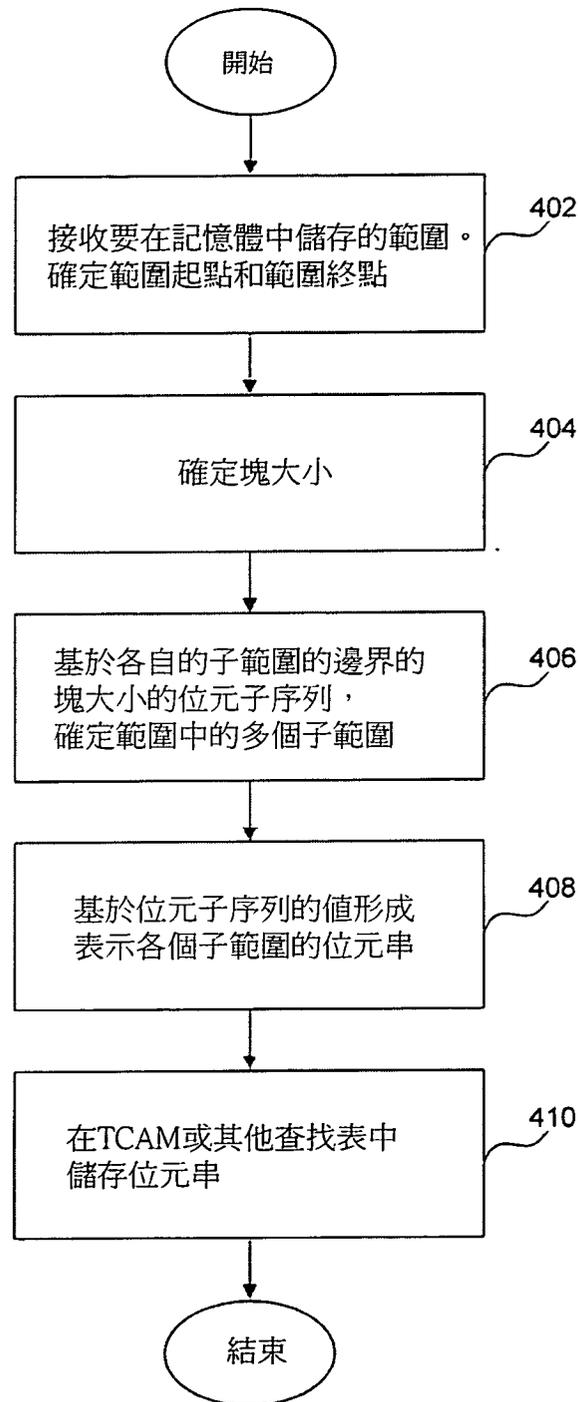


圖 4

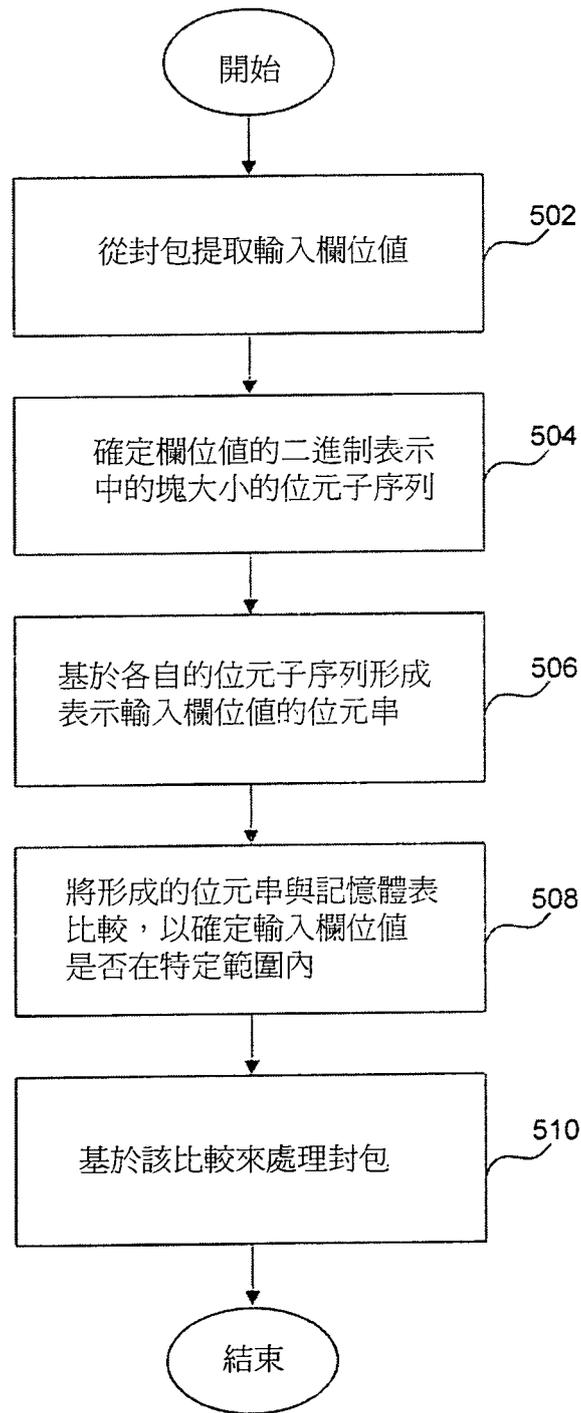


圖 5

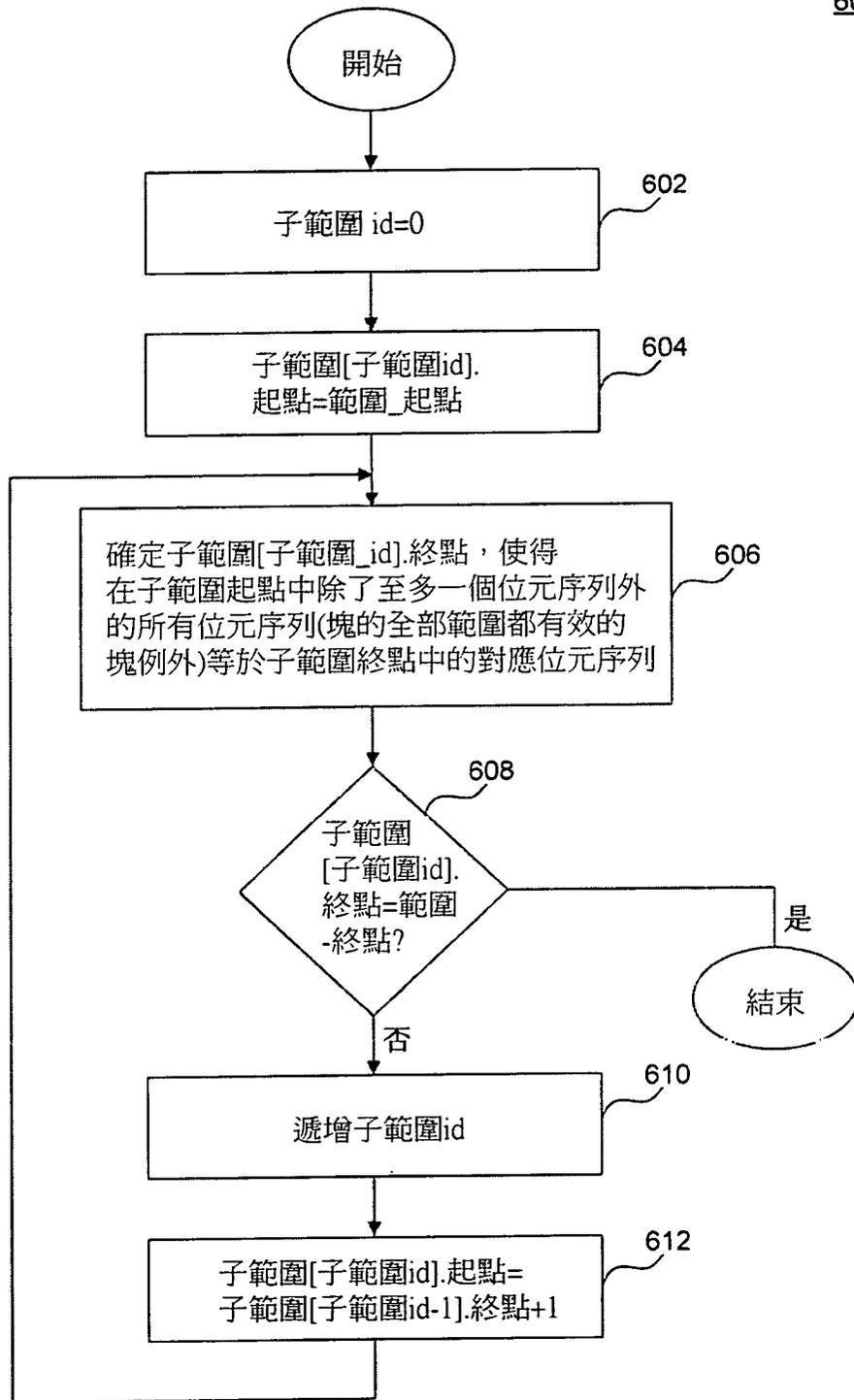


圖 6

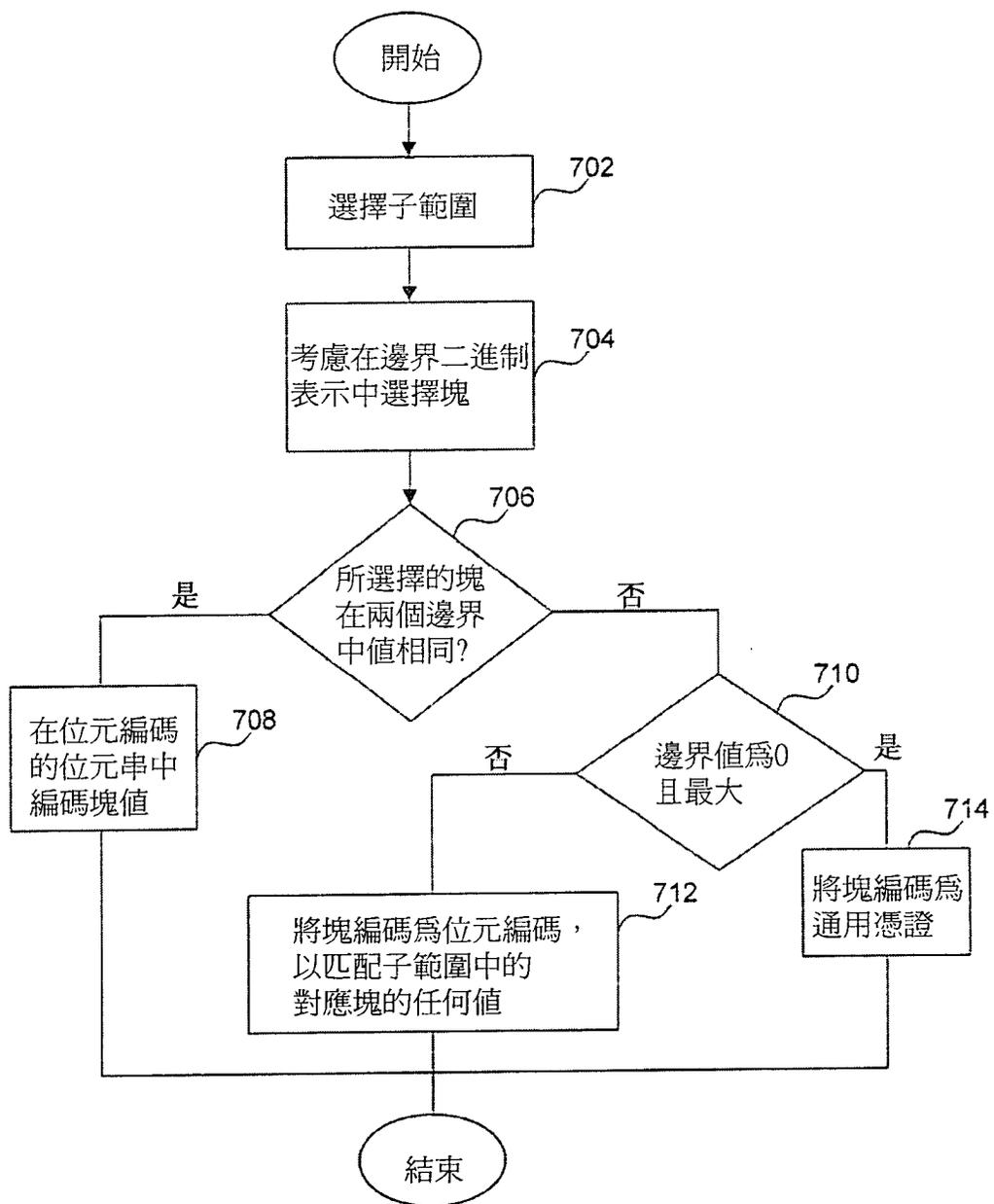


圖 7

