

# 公告本

申請日期： <u>81. 4. 12</u>	案號： <u>1107kkz</u>
類別： <u>H04N 3/4</u>	

(以上各欄由本局填註)

## 發明專利說明書

526657

一、發明名稱	中文	用於移動估計的全域消除演算法及其硬體架構
	英文	
二、發明人	姓名 (中文)	1. 陳良基 2. 黃毓文 3. 簡韶逸
	姓名 (英文)	1. 2. 3.
	國籍	1. 中華民國 2. 中華民國 3. 中華民國
	住、居所	1. 台北縣新店市安忠路130號 2. 台北市農安街202之1號5樓 3. 台北市民生東路三段113巷6弄17號7樓
三、申請人	姓名 (名稱) (中文)	1. 國立臺灣大學
	姓名 (名稱) (英文)	1.
	國籍	1. 中華民國
	住、居所 (事務所)	1. 台北市大安區羅斯福路四段1號
	代表人姓名 (中文)	1. 陳維昭
	代表人姓名 (英文)	1.



本案已向

國(地區)申請專利

申請日期

案號

主張優先權

無

有關微生物已寄存於

寄存日期

寄存號碼

無

## 五、發明說明 (1)

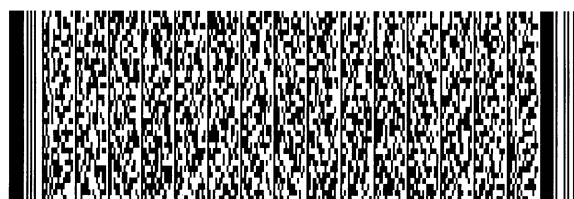
## 發明領域：

本發明係有關一種區塊比對移動估計 (Block Matching Motion Estimation) 方法，特別是關於一種高效率之全域消除演算法 (Global Elimination Algorithm) 及其硬體架構，以去除視訊序列上的多餘性，達到視訊壓縮之目的者。

## 發明背景：

按，隨著科技產業的快速發展，在視訊序列 (video sequence) 傳輸中視訊資料的流量及品質益顯重要，而所謂視訊序列就是由一連續時間上的一連串影像所組成。對視訊序列而言，由於視訊序列需要的儲存空間非常龐大，所以對於空間有限的儲存設備或傳輸頻寬而言，都希望能夠減少視訊序列所需要佔用的儲存空間，故須將視訊序列加以壓縮，因此需要視訊壓縮的技術。視訊壓縮通常係利用去除視訊資料中的多餘性 (redundancy) 來達到壓縮的目的；而移動估計 (motion estimation) 則是為了去除視訊序列中的時間多餘性 (temporal redundancy) 之一個壓縮技巧。

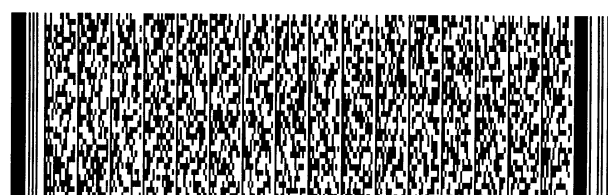
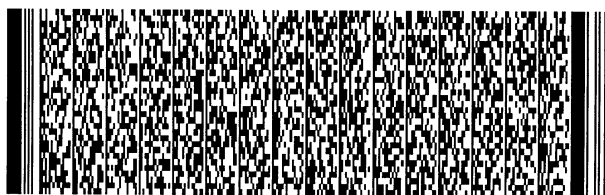
所謂的移動估計即在描述如何在視訊序列中，在兩個相鄰時間軸的畫面上，找到與現在處理區塊最為近似的區塊。在許多移動估計演算法中，最常採用的是全搜尋區塊比對演算法 (Full-Search Block Matching Algorithm)，但其運算量十分龐大，對即時性應用而言，遠超過現今一般微處理器的能力。由於全搜尋區塊比對演算法的資料



## 五、發明說明 (2)

流十分規律，許多平行化、多管線 (Pipelined) 硬體架構已被提出，但在這些架構中，一維陣列的運算速度太慢，對於大畫面與大搜尋範圍的應用，其操作頻率必須大幅增加；二維陣列的運算速度較快，但其邏輯閘的數量過於龐大，成本過高；數狀架構雖然速度與面積的表現十分良好，但其所需要的記憶體位元寬度 (Bit-Width) 太大，使其可行性大幅減低。

而為了可以減少全搜尋區塊比對演算法的高運算量，遂發展出一種連續消除演算法 (Successive Elimination Algorithm)，其係可得到和全搜尋區塊比對演算法相同的結果，使它比其他許多必須犧牲峰值信號雜訊比 (PSNR) 的快速搜尋演算法，例如三步搜尋 (Three-Step Search)、鑽石搜尋 (Diamond Search) 或二維對數搜尋 (2D Log Search) 等演算法，運算效果更佳。該連續消除演算法之演算流程如第一圖所示，首先，如步驟 S10 所示在計算出每個搜尋位置的連續消除演算值  $sea(m, n)$  之後，如步驟 S12 比較該連續消除演算值  $sea(m, n)$  是否大於最小的絕對差值總和  $SAD_{min}$  (Sum of Absolute Differences, SAD)，若  $sea(m, n) > SAD_{min}$ ，如步驟 S14 省略搜尋位置  $(m, n)$ ，直接進行步驟 S22；若  $sea(m, n) < SAD_{min}$ ，則如步驟 S16 繼續計算每個搜尋位置的絕對差值總和  $SAD(m, n)$ 。在取得  $SAD(m, n)$  之後，再如步驟 S18 比較  $SAD(m, n)$  是否大於  $SAD_{min}$ ，若  $SAD(m, n) > SAD_{min}$ ，直接進行步驟 S22；相反地，若  $SAD(m, n) < SAD_{min}$ ，則進行步驟 S20，更新最小的絕對差值總



## 五、發明說明 (3)

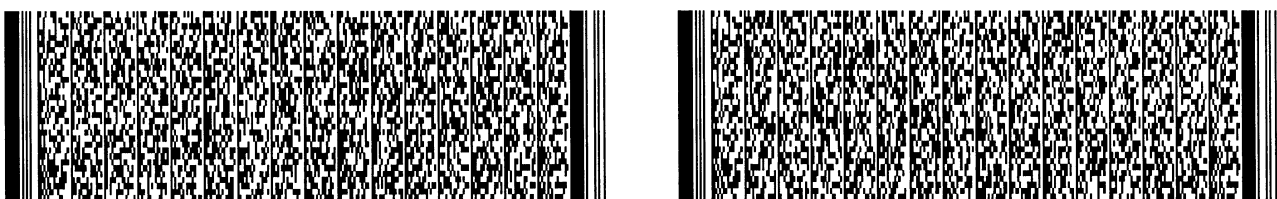
和  $SAD_{min}$ ，然後才進行步驟 S22。步驟 S22 係判斷是否為最後一個搜尋位置，若是最後一個  $(m, n)$ ，表示找到具有最小 SAD 值的搜尋位置，如步驟 S26 所示，此即為估算出來的移動向量 MV，結束整個過程；若步驟 S22 之  $(m, n)$  尚有搜尋位置未掃描到，則如步驟 S24 更新下一次的搜尋位置  $(m, n)$ ，並繼續回到步驟 S10，重複上述步驟。

前述在計算每個搜尋位置的 sea 值後，程序出現了分支，使資料流十分不規則且無法提前被預測，故無法採用心脈式跳動 (Systolic) 陣列架構來設計硬體架構。而後來才發展出來的多階層連續消除演算法 (Multi-Level Successive Elimination Algorithm) 亦仍然存有相同之問題。

再者，連續消除演算法必須對移動向量 MV (Motion Vector) 有一個很好的初始猜測，才能有效地降低運算量，這對影像中不規則移動的區域而言是十分困難的；此外，若真正的移動向量超出搜尋範圍，連續消除演算法對搜尋位置的省略比率甚至有可能會低到使該移動向量之運算時間比全搜尋區塊比對演算法還久；再則，為了增加省略計算 SAD 的次數，連續消除演算法往往以螺旋狀掃描

(Spiral Scan) 來決定搜尋位置之先後順序，這對硬體所要付出的代價高於傳統光柵掃描 (Raster Scan)。

因此，本發明即在針對上述之困擾，提出一種全域消除演算法及其對應的硬體架構，以改善習知連續消除演算法的缺點。



## 五、發明說明 (4)

## 發明目的與概述：

本發明之主要目的係在提出一種用於移動估計的全域消除演算法及其硬體架構，其係將連續消除演算法資料流的分支做適當的移除，使資料流更加規律、順暢，且更適合於硬體架構實現，以大幅改善連續消除演算法之缺失者。

本發明之另一目的係在提出一種用於移動估計的全域消除演算法，其搜尋結果與全搜尋區塊比對演算法的搜尋結果有很高的相似度，有時亦具有更佳的峰值信號雜訊比 (PSNR)，使得本發明之全域消除演算法具有很高的可靠性。

本發明之再一目的係在提出一種用於移動估計的全域消除演算法之硬體架構，與其它許多全搜尋區塊比對演算法架構設計比較，其所使用的每個邏輯閘之運算能力係為最高，且在相同的移動向量生產率下邏輯閘所消耗的功率則為最低。

本發明之又一目的係在提出一種用於移動估計的全域消除演算法及其硬體架構，其係具有易於支援先進預測模式之優點者。

為達到上述之目的，本發明全域消除演算法之步驟包括：在視訊序列中將參考區塊與每個搜尋位置的候選區塊以較少點數的粗特徵表示，並比對兩者之粗特徵；接著再搜尋粗特徵與該參考區塊最相似的M個候選區塊，再比對該參考區塊與該M個候選區塊的細特徵；以及在該M個候選



## 五、發明說明 (5)

區塊中選出最小細特徵差異的候選區塊，即可結束整個演算過程。

本發明用於移動估計的硬體架構則包括：一用以平行計算複數區塊中的每個粗特徵的心脈式跳動模組；一樹狀加法器，其係比對參考區塊及候選區塊的每個粗特徵，且可重複使用來比對該參考區塊與候選區塊的細特徵；並利用至少一樹狀比較器來平行搜尋該粗特徵與該參考區塊最相似的M個候選區塊；且該心脈式跳動模組、該樹狀加法器及該樹狀比較器之作動係由一控制裝置所控制；以及至少一組用以儲存該參考區塊與候選區塊所有資料的記憶體。

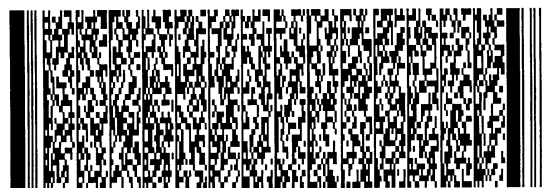
底下藉由具體實施例配合所附的圖式詳加說明，當更容易瞭解本發明之目的、技術內容、特點及其所達成之功效。

## 圖號說明：

10	心脈式跳動模組	12	平行樹狀加法器
14	平行樹狀比較器	16、16'	記憶體
18	控制單元	20	多工器
22	第一多工器網路	24	第二多工器網路
26	暫存器		

## 詳細說明：

移動估計是視訊壓縮中最關鍵的技術，可應用於數位攝影機等視訊產品上，本發明即針對移動估計提出一種全域消除演算法及其硬體架構，其係將運算資料流的分支做



## 五、發明說明 (6)

適當的移除，使得資料流更規律、更適合硬體實現，並具有可靠、快速及高效率之特性，以大幅改善習知（多階層）連續消除演算法的缺點。

第二圖為本發明之全域消除演算法的流程示意圖，如圖所示，全域消除演算法係包括下列步驟：首先，在視訊序列中，計算出每個搜尋位置（候選區塊）之多階層連續消除演算值  $msea(m, n)$ ，如步驟 S30 所示；然後如步驟 S32 所示判斷搜尋位置  $(m, n)$  是否為最後一個，若搜尋位置  $(m, n)$  不是最後一個，則如步驟 S34，更新下一個搜尋位置  $(m, n)$ ，然後再回到步驟 S30 繼續重複上述步驟；在步驟 S34 中，搜尋位置更新的順序可為任意，且不會影響最後結果，因此採用傳統之光柵掃描即可；若搜尋位置  $(m, n)$  為最後一個，則直接進行步驟 S36。以  $-p \sim p-1$  代表搜尋範圍，步驟 S36 為找出 M 個搜尋位置，其  $msea(m, n)$  值為在全部的  $(2p)^2$  個搜尋位置中最小的 M 個，然後忽略其它  $(2p)^2 - M$  個搜尋位置。在完成步驟 S36 之後，如步驟 S38 計算該 M 個搜尋位置各自之絕對差值總和  $SAD(m, n)$  值；最後如步驟 S40 所示，在步驟 S38 計算出的 M 個搜尋位置的  $SAD(m, n)$  值中，選擇一個最小的 SAD 值，該最小 SAD 值的搜尋位置為全域消除演算法所估計出來的移動向量 MV。

該步驟 S32 即為將演算法命名為全域消除的原因，它並不像（多階層）連續消除演算法是把搜尋位置逐一檢查能否省略，而是在計算完所有搜尋位置的  $msea$  值（多階層連續消除演算值）後，才決定何者將被省略。由於在計算





## 五、發明說明 (7)

每個搜尋位置的mse值之過程中，流程只會走右邊的分支，資料流是連續有規律的，因此，這一部份的硬體設計就可以採用心脈式跳動陣列架構。

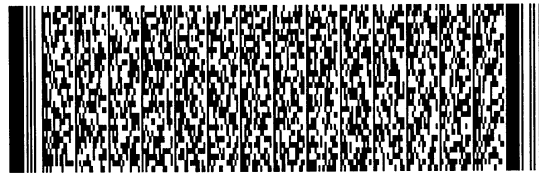
其中，M值的選擇為速度和編碼效率的取捨，M值之較佳者係介於多階層連續消除演算值，通常是介於1~63之間。一般而言，M越大速度越慢，但編碼效率較高；M越小可以省較多的運算，但編碼效率較低。無論採用的M值為何，現在每個移動向量所需要的處理時間均為固定且可預測的，這對以硬體實現之編碼系統的工作排程將更有幫助。

雖然全域消除演算法不能像(多階層)連續消除演算法一樣可保證搜尋結果百分之百與全搜尋區塊比對演算法相同，但全域消除演算法仍是非常可靠的。本發明對常見的兩種情況做了許多測試，第一種情況(a)為QCIF(176×144)畫面、16×16區塊、-16~+15搜尋範圍、第三階層連續消除之mse值及M=7，省略SAD計算之搜尋位置比率為99.31%；第二情況(b)為CIF(352×288)畫面、16×16區塊、-32~+31搜尋範圍、第三階層連續消除之mse值及M=7，省略SAD計算之搜尋位置比率為99.83%。測試結果如表一所示，此驗證過程實驗了許多標準測試視訊序列，全域消除演算法所補償回來的畫面其平均峰值信號雜訊比與全搜尋區塊比對演算法的結果非常相近，最大的差別為Hall Monitor CIF，只比全搜尋區塊比對演算法低了0.08dB；此外，有時候全域消除演算法所補償回來的畫面，其峰值



## 五、發明說明 (8)

信號雜訊比會比全搜尋區塊比對演算法所得到的結果還要高，如Foreman QCIF、Silent QCIF與Table Tennis QCIF則可看出此現象。一般認為，全搜尋區塊比對演算法的峰值信號雜訊比會最高是不正確的，因為最小的SAD值（Sum of Absolute Difference，絕對差值總和）並不能保證最小的平均平方差（Mean Square Error），例如： $1 + 9 < 5 + 6$ ，但 $1^2 + 9^2 > 5^2 + 6^2$ 。在大部份時候，全域消除演算法的結果和全搜尋區塊比對演算法所非常接近，第三圖及第四圖即可說明此事，第三圖所示是Mobile Calendar CIF視訊中，全域消除演算法的移動向量MV和全搜尋區塊比對演算法相同的百分比，在三百張畫面中，平均有高達98.1%的移動向量是相同的；第四圖所示則是在Mobile Calendar CIF視訊中，全域消除演算法的峰值信號雜訊比曲線以及全搜尋區塊比對演算法的峰值信號雜訊比曲線，由於兩條曲線實在非常接近，因此要分辨它們並不太容易。因此，藉由以上數據可以顯示，本發明所提出的全域消除演算法具有很高的可靠性。



## 五、發明說明 (9)

表一

單位為 dB

標準視訊序列	(a)		(b)	
	全搜尋區塊 比對演算法	全域消除演 算法	全搜尋區塊 比對演算法	全域消除演 算法
Coastguard	32.93	32.93	31.59	31.55
Container	43.11	43.11	38.53	38.53
Foreman	32.21	32.22	32.85	32.82
Hall Monitor	32.98	32.97	34.90	34.82
Mobile Calendar	26.15	26.15	25.20	25.16
Silent	35.14	35.16	36.12	36.11
Stefan	24.71	24.67	25.73	25.71
Table Tennis	32.10	32.11	33.03	32.96
Weather	38.42	38.42	37.45	37.45

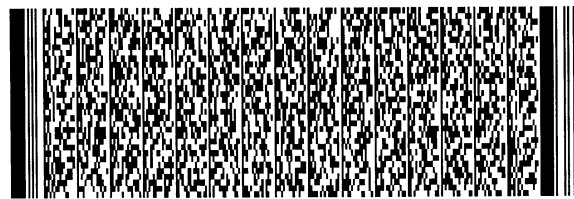
在說明本發明之全域消除演算法之後，接著將詳細說明其相對應之硬體架構，底下將以 $16 \times 16$ 的區塊大小，第三階層連續消除之mse值以及 $M=7$ 為例子，並以第五圖為主配合各部份架構來解釋本發明，以使熟習此項技術者將可參酌此實施例之描述而獲得足夠的知識而據以實施。如第五圖所示，用於移動估計的該硬體架構係包括：心脈式跳動模組 (systolic module) 10、平行樹狀加法器 (parallel adder tree) 12、平行樹狀比較器 (parallel comparator tree) 14、用以控制各元件作動



## 五、發明說明 (10)

之控制裝置，以及儲存參考區塊資料的記憶體16與搜尋區域資料的記憶體16'；其中，該控制裝置包含控制單元 (CONTROL UNIT) 18以及由多工器 (MUX) 20與多工器網路 (MUX NETWORK 1, 2) 22、24組成的控制電路。

如第五圖所示，該心脈式跳動模組10負責在同一個時脈循環 (Clock Cycle) 中計算十六個 $4 \times 4$ 大小的子區塊內之像素強度和，即粗特徵，並平行輸出十六個子區塊的結果。請同時參考第六圖所示，其係顯示心脈式跳動模組的資料流，其中 $c_{l,k}$ 與 $s_{l,k}$ 分別代表參考區塊資料 $c(k, l)$ 與搜尋區域資料 $s(k, l)$ ，長方形則代表轉移暫存器 (Shift Register) 26，搜尋範圍是以 $-16 \sim +15$ 為例子。區塊資料係以一欄一欄的方式平行載入心脈式跳動模組10，當 $t=0 \sim 15$ 時，參考區塊資料被載入，在 $16 \times 16$ 大小的參考區塊中之十六個 $4 \times 4$ 子區塊，其各自的像素強度和 (即圖六中之 $sum_{00} \sim sum_{33}$ ，表示為 $csum_{00} \sim csum_{33}$ ) 在 $t=15$ 時被計算出來，並在 $t=16$ 時脈的正緣被寫入至十六個十二位元暫存器中；接著，換搜尋區域資料以一欄一欄的方式平行載入心脈式跳動模組10，當 $t=16 \sim 62$ 時，在搜尋位置 $(-16, -16) \sim (+15, -16)$ 的候選區塊資料被載入，搜尋位置 $(-16, -16) \sim (+15, -16)$ 所對應的候選區塊中之十六個子區塊像素強度和 (即圖六中之 $sum_{00} \sim sum_{33}$ ，表示為 $rsum_{00} \sim rsum_{33}$ ) 分別在 $t=31 \sim 62$ 時被計算出來；同理，下一列的搜尋區域資料也是以相同的方式運作，搜尋位置 $(-16, -15) \sim (+15, -15)$ 的候選區塊資料在 $t=63 \sim 109$ 時被載入，搜尋



## 五、發明說明 (11)

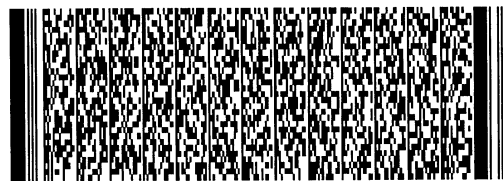
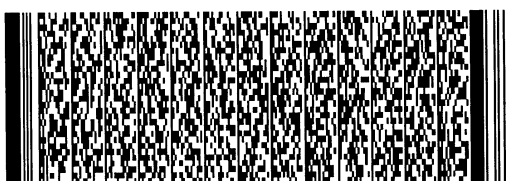
位置  $(-16, -15) \sim (+15, -15)$  所對應的候選區塊中之十六個子區塊像素強度和分別在  $t=78 \sim 109$  時被計算出來。由上述內容可知，每一列搜尋位置需要  $(2p+N-1)$  個時脈， $2p$  列搜尋位置則需要  $2p(2p+N-1)$  個時脈，再加上一開始仍需要  $N$  個時脈來載入參考區塊資料，故此模組 10 總共需要  $N+2p(2p+N-1)$  個時脈來計算所有區塊中的子區塊像素強度和（粗特徵）。

由心脈式跳動模組 10 計算出來的子區塊像素強度和等資料將被傳送至該平行樹狀加法器 12，請同時參閱第六圖及第七圖所示，平行樹狀加法器 12 之目的係在利用下列公式計算出  $msea$  值，

$$\begin{aligned}
 SAD(m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |c(i,j) - s(i+m,j+n)| \\
 &\geq \sum_{q=0}^{L-1} |K_q - SB_q(m,n)| \equiv msea(m,n) \\
 &\geq \left| \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c(i,j) - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s(i+m,j+n) \right| \equiv |K - SB(m,n)| \equiv sea(m,n)
 \end{aligned}$$

在上式中， $K$  代表參考區塊之像素和， $SB(m,n)$  代表在搜尋位置  $(m,n)$  的候選區塊之像素和， $K$  與  $SB$  之絕對差值

（Absolute Difference）即為  $sea$  值，亦可稱為第一階層連續消除之  $msea$  值；若將一個區塊切分成  $L$  個子區塊， $K_q$  代表參考區塊之第  $q$  個子區塊的像素和， $SB_q(m,n)$  代表在搜尋位置  $(m,n)$  的候選區塊之第  $q$  個子區塊的像素和，再將

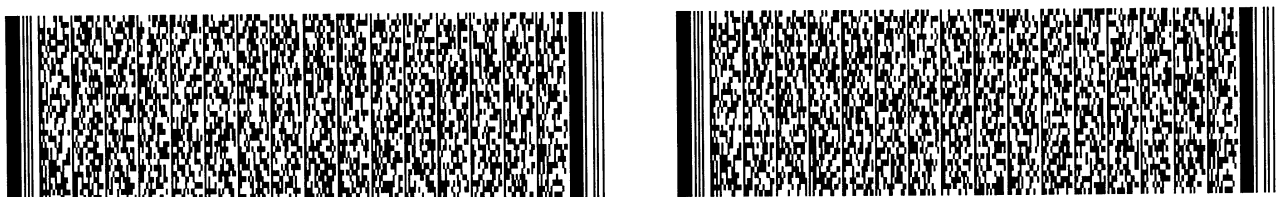


## 五、發明說明 (12)

L個 $K_q$ 和 $SB_q$ 之絕對差值加起來即可得到 $msea$ 值；若一個區塊被分成 $4^{Level-1}$ 個大小相同的子區塊，則將其稱為第Level階層連續消除；在所舉的例子中，第三階層連續消除則是把一個 $16 \times 16$ 區塊分成16個 $4 \times 4$ 子區塊。第七圖中所示之 $AD_{xx}$ 係負責計算參考區塊之子區塊像素強度和 $csum_{xx}$ 與搜尋區塊之子區塊像素強度和 $rsum_{xx}$ 之間的絕對差值，樹狀加法器12係將 $AD00 \sim AD33$ 之結果相加起來，以得到 $msea$ 值。

在依序取得各區塊之 $msea$ 值之後，旋即進入平行樹狀比較器14，其目的是要找出具有最小 $msea$ 值的M個搜尋位置，其做法是將目前最小的M個 $msea$ 值與其對應之移動向量存在暫存器中，若輸入的 $msea$ 值比這M個 $msea$ 值其中一個或更多個還小，就把這M個 $msea$ 值中最大的那一個換成現在輸入的 $msea$ 值；若這M個 $msea$ 值中有兩者以上同為最大值，只選擇其中一個換成現在輸入的 $msea$ 值。

第八圖為本發明之平行樹狀比較器的電路示意圖，圖中有「\_reg」的符號代表暫存器，MAX代表比較器。在(a)圖部份中的電路必須在第一個來自平行樹狀加法器12的有效 $msea$ 值來臨前，先適當地將暫存器 $msea1\_reg \sim msea7\_reg$ 初始設定為 $0xFFFF$  (65535)，此部份的電路會計算出 $msea\_in\_reg$ 與 $msea1\_reg \sim msea7\_reg$ 中的 $msea$ 最大值 $msea\_max$ ，且比較器MAX會將兩個輸入中較大者輸出。(b)圖部份的電路是判斷 $msea1\_reg \sim msea7\_reg$ 有沒有等於最大值 $msea\_max$ ；其中 $EQU_x$ 負責比較 $msea_x\_reg$ ， $x=1$

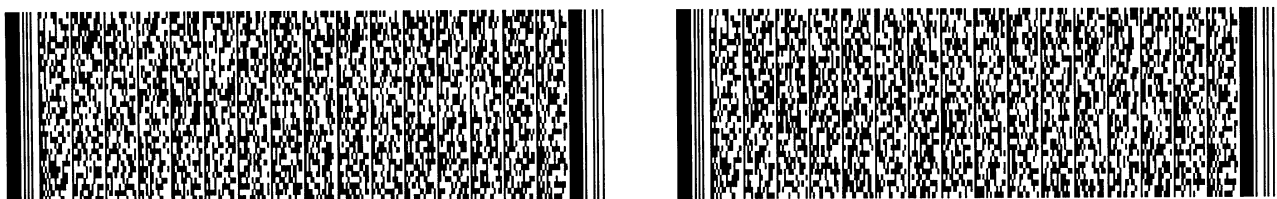


## 五、發明說明 (13)

~ 7；CHECK 電路則是當有兩個以上的  $msea\_reg$  為最大值  $msea\_max$  時，僅選擇其中一個，也就是說，取代信號  $replace_x$  為主動 (active)，代表  $msea\_reg$  與  $mvx\_reg$  應該分別被  $msea\_in\_reg$  與  $mv\_in\_reg$  所取代，且最多只會有一個  $replace_x$  信號為主動。(c) 圖部份的電路就是負責執行取代的動作，MUX 為受取代信號  $replace_x$  控制的多工器 (Multiplexer)。

如此一來，目前最小的  $M$  個  $msea$  值與其對應之移動向量就隨時儲存在暫存器中，直到所有的搜尋位置 (候選區塊) 的  $msea$  值皆輸入平行樹狀比較器 14 後，暫存器中就會有在  $(2p)^2$  搜尋位置為最小的  $M$  個  $msea$  值與其對應之移動向量，後續再去計算這  $M$  個搜尋位置的搜尋區域資料 SAD 值，找出最小者，輸出移動向量，全域消去演算法就完成了一個移動向量的估計。特別要注意的是，在每一列搜尋位置之欄位資料輸入心脈式跳動模組 10 時，其前  $N-1$  個時脈透過平行樹狀加法器 12 所產生出來的  $msea$  值是無效的，此時輸入平行樹狀比較器 14 的  $msea$  值必須被替換成  $0xFFFF$  (65535)，如此才能得到正確的結果。

其中，為了平行輸出區塊的欄位資料，作法如下所述：在搜尋範圍中的資料，總共有  $(2p+N-1)$  列，本發明將列資料 (Row Data) 編號為  $0 \sim (2p+N-2)$ ，編號除以  $N$  後餘數為 0 的列資料就放置在記憶體 16 之 RAM00，餘數為 1 的列資料就放置在 RAM01，如第五圖所示，依此類推；參考區塊資料則是放在另外一個寬度為 128 位元的記憶體 16' 中；



## 五、發明說明 (14)

使參考區塊和候選區塊的欄位資料皆可被平行輸出。區塊的欄位資料平行輸出後，在進入心脈式跳動模組10前，必須先經過第一多工器網路 (MUX NETWORK 1) 22，使其進入正確的子區塊；在 $N=16$ 與第三階層mseal下，其係為16個4對1之八位元多工器 (4-to-1 8-bit Multiplexer)，在不同列的搜尋位置上，控制第一多工器網路22的控制訊號必須做適當的改變。

同樣地，最後在計算 $M$ 個搜尋位置的SAD值時，候選區塊的資料必須先經過第二多工器網路 (MUX NETWORK 2) 24再進入平行樹狀加法器12，其為16個16對1之位元多工器 (16-to-1 8-bit Multiplexer)；在不同列的搜尋位置，控制第二多工器網路24的控制訊號亦得做適當的改變。因此，本發明係需要 $N+2p(2p+N-1)$ 時脈找出擁有最小mseal值的 $M$ 個搜尋位置，然後接著要計算這 $M$ 個搜尋位置之SAD值時，還可重複利用平行樹狀加法器12的資源；每個搜尋位置之SAD值需要 $N$ 個時脈來計算， $M$ 個搜尋位置之SAD值則需要 $MN$ 個時脈來計算。綜合以上所述，以 $N=16$ 與第三階層mseal為例，本發明所提出的硬體架構需要 $N+2p(2p+N-1)+MN$ 個時脈來計算一個移動向量。

至此，本發明之精神已說明完畢，以下特以一具體實驗範例來驗證說明上述之原理及功效。為了分析本發明所提出來之硬體架構的效能，本發明提出來的硬體架構與基於全搜尋區塊比對演算法之硬體架構做比較，比較對象是從參考文獻[1]~[7]中提出來的，結果顯示在表二與表三



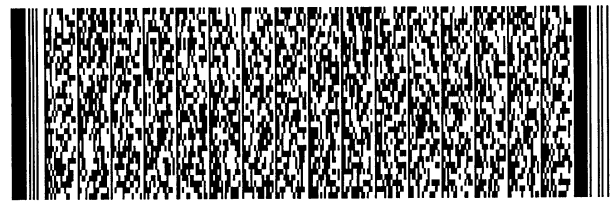
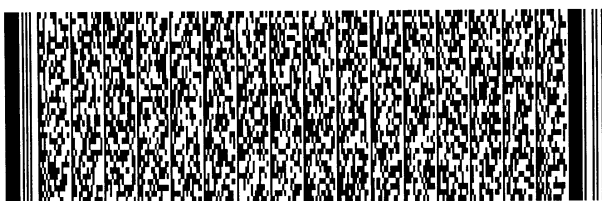


## 五、發明說明 (15)

中，其中，表二是在 $16 \times 16$ 區塊、 $-16 \sim +15$ 搜尋範圍、第三階層連續消除以及 $M=7$ 之條件下，不同架構之比較；表三則是在 $16 \times 16$ 區塊、 $-32 \sim +31$ 搜尋範圍、第三階層連續消除以及 $M=7$ 之條件下，不同架構之比較。

本發明比較的是各架構之處理元素 (Processing Element) 陣列的部份，控制電路只佔很小的部份，所以並沒有將它以硬體實現出來。每種架構之處理元素陣列是以 SYNOPSIS Design Analyzer 所合成，並採用 AVANT! 0.35  $\mu$ m Cell Library，所下的最短路徑限制 (Critical Path Constraint) 為 20 ns，也就是電路可達之工作頻率至少為 50 MHz。在表二和表三中有\*號標記的架構除了處理元素之外，尚需要許多額外的邏輯電路，大多為轉移暫存器 (Shift Register)，以增加資料的重複使用性，因此，這些架構實際上在邏輯電路方面的閘數 (Gate Count) 與消耗功率會比模擬的結果還要高不少。在表二和表三中，本發明除了記憶體、第二多工器網路與控制單元未實現，其它皆有在模擬中考慮到，此外，本發明在模擬時切了3級管線 (pipelines)。

為了要更公平地比較這些架構，必須在相同的位移向量生產率 (Throughput of Motion Vectors, NO. of Motion Vectors / Sec.) 之下才行。因此，我們定義了「正常化每邏輯閘之處理能力」 (Normalized Processing Capability per Gate, NPCPG) 與「正常化功率消耗」 (Normalized Power, NP)：



## 五、發明說明 (16)

$$NPCPG_{XXX} = \frac{[(\text{Re quired Freq. for CIF 30fps})^{-1} / (\text{Gate Count @ 50MHz})] \text{ for XXX}}{[(\text{Re quired Freq. for CIF 30fps})^{-1} / (\text{Gate Count @ 50MHz})] \text{ for GEA}}$$

$$NP_{XXX} = \frac{[(\text{Power @ 50MHz}) \times (\text{Re quired Freq. for CIF 30fps / 50MHz})] \text{ for XXX}}{[(\text{Power @ 50MHz}) \times (\text{Re quired Freq. for CIF 30fps / 50MHz})] \text{ for GEA}}$$

一般而言，一維陣列架構的處理速度不夠快，對於大畫面與大搜尋範圍的應用，其操作頻率必須大幅增加。二維陣列架構的運算速度較快，但其邏輯閘的數量過於龐大，成本過高；參考文獻[6]的架構雖然屬於一維陣列，但它是採用資料交錯（Data-Interlacing）與二維資料重複利用（2-D Data Reuse），所以它的問題與二維陣列架構相同，也就是邏輯閘太多。數狀架構雖然速度與面積的表現十分良好，但其所需要的記憶體位元寬度太大，使其可行性大幅減低。本發明所提出的硬體架構，其運算速度大致上只比二維陣列架構與樹狀架構慢一些（架構[3]比本發明所提出的架構還慢），但邏輯閘數目遠比它們小；一維陣列架構的運算速度遠低於本發明所提出的架構，架構[1]的邏輯閘數目甚至在較大搜尋範圍時比本發明所提出的架構還大。因此，本發明在「正常化每邏輯閘之處理能力」與「正常化功率消耗」之表現比其它架構好很多。



## 五、發明說明 (17)

表二

Architecture	Description	No. of PE	Cycles per MV	Required Memory I/O	Required Freq. for CIF 30 fps	Gate Count @50MHz	NPCPG	Gate-Level Power @50MHz	NP
[1] Yang	1-D semi-systolic	32	8192	24 bits	97.32 MHz	28.0K	0.13	26.0 mW	2.99
[2] AB1	1-D systolic	16	24064	256 bits	285.88 MHz	3.8K	0.32	11.7 mW	3.95
[2] AB2	2-D systolic	256	1504	128 bits	17.87 MHz	95.1K	0.20	227.8 mW	4.82
[3] Hsieh*	2-D systolic	256	2209	8 bits	26.24 MHz	100.6K	0.13	147.2 mW	4.57
[4] Tree	Tree structure	256	1024	2048 bits	12.17 MHz	56.1K	0.51	179.5 mW	2.59
[5] Yeo	2-D semi-systolic	1024	256	24 bits	3.04 MHz	447.4K	0.26	1052.6 mW	3.79
[6] Lai	1-D semi-systolic	1024	256	24 bits	3.04 MHz	387.6K	0.30	845.6 mW	3.04
[7] SA*	2-D systolic	256	1024	16 bits	12.17 MHz	126.5K	0.23	258.0 mW	3.72
[7] SSA*	2-D semi-systolic	256	1024	16 bits	12.17 MHz	106.0K	0.27	280.1 mW	4.04
Ours	Based on GEA	16	1635	256 bits	19.42 MHz	17.9K	1.00	43.4 mW	1.00

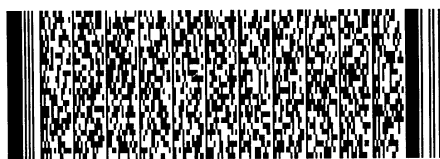


## 五、發明說明 (18)

表三

Architecture	Description	No. of PE	Cycles per MV	Required Memory I/O	Required Freq. for CIF 30 fps	Gate Count @50MHz	NPCPG	Gate-Level Power @50MHz	NP
[1] Yang	1-Dsemi-systolic	32	16384	24 bits	194.64 MHz	56.0K	0.10	52.0 mW	3.78
[2] AB1	1-D systolic	16	80896	256 bits	961.04 MHz	3.8K	0.30	11.7 mW	4.20
[2] AB2	2-D systolic	256	5056	128 bits	60.07 MHz	95.1K	0.19	227.8 mW	5.12
[3] Hsieh*	2-D systolic	256	6241	8 bits	74.14 MHz	100.6K	0.15	147.2 mW	4.08
[4] Tree	Tree structure	256	4096	2048 bits	48.66 MHz	56.1K	0.40	179.5 mW	3.27
[5] Yeo	2-Dsemi-systolic	1024	256	24 bits	3.04 MHz	1790.0K	0.20	4210.3 mW	4.79
[6] Lai	1-Dsemi-systolic	1024	256	24 bits	3.04 MHz	1550.4K	0.23	3382.4 mW	3.84
[7] SA*	2-D systolic	256	4096	16 bits	48.66 MHz	126.5K	0.18	258.0 mW	4.69
[7] SSA*	2-Dsemi-systolic	256	4096	16 bits	48.66 MHz	106.0K	0.21	280.1 mW	5.09
Ours	Based on GEA	16	5187	256 bits	61.62 MHz	17.9K	1.00	43.4 mW	1.00

在新一代的視訊壓縮標準中，如 H.263+、MPEG-4 等等，提供了其它類型之移動估計模式，所採用的區塊大小不再限定是傳統的  $16 \times 16$  大小，而是在原本  $16 \times 16$  區塊中以



## 五、發明說明 (19)

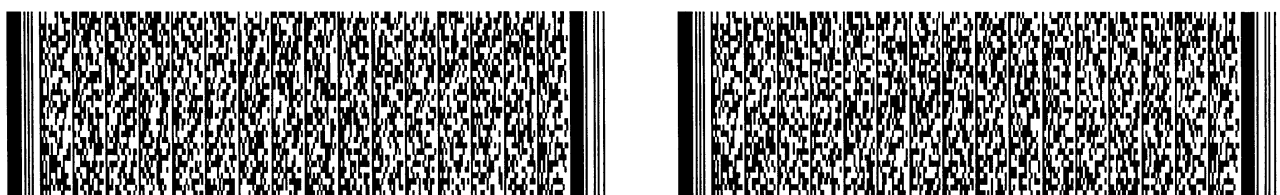
四個  $8 \times 8$  大小的區塊產生四組移動向量，若壓縮演算法能適當地判斷該採用何者，編碼效果可以有顯著提升，此種移動估計模式稱為「先進預測模式」(Advanced Prediction Mode)。本發明所提出的硬體架構只要增加四組平行樹狀比較器，就可以輕易地支援先進預測模式，如第九圖所示。若要以本架構支援先進預測模式，採用第四階層 msea 來設計電路，可以得到比較好的編碼效果。

因此，本發明使資料流更加規律、順暢，更適合於硬體架構實現，以大幅改善連續消除演算法之缺失者，並同時兼具有可靠性高，邏輯閘運算能力高，且在相同的移動向量生產率下邏輯閘消耗的功率為最低等之特性。

以上所述之實施例僅係為說明本發明之技術思想及特點，其目的在使熟習此項技藝之人士能夠瞭解本發明之內容並據以實施，當不能以之限定本發明之專利範圍，即大凡依本發明所揭示之精神所作之均等變化或修飾，仍應涵蓋在本發明之專利範圍內。

## 參考文獻：

[1] K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," IEEE Trans. on Circuits and Systems, vol. 36, no. 2, pp. 1317-1358, Oct. 1989.



## 五、發明說明 (20)

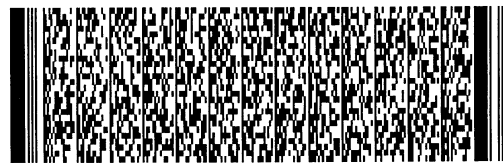
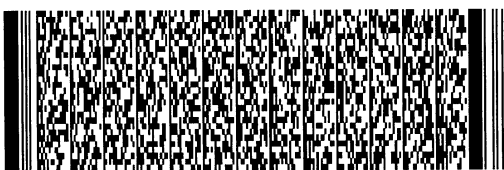
[2] T. Komarek and P. Pirsch, "Array architectures for block matching algorithms," IEEE Trans. on Circuits and Systems, vol. 36, no. 2, pp. 1301-1308, Oct. 1989.

[3] C.H. Hsieh and T.P. Lin, "VLSI architecture for block-matching motion estimation algorithm," IEEE Trans. on Circuits and Systems for Video Technology, vol. 2, no. 2, pp. 169-175, Jun. 1992.

[4] Y.S. Jehng, L.G. Chen and T.D. Chiueh, "An efficient and simple VLSI tree architecture for motion estimation algorithms," IEEE Trans. on Signal Processing, vol. 41, no. 2, pp. 889-900, Feb. 1993.

[5] H. Yeo and Y.H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 5, no. 5, pp. 407-416, Oct. 1995.

[6] Y.K. Lai and L.G. Chen, "A data-interlacing architecture with two-dimensional data-reuse for



## 五、發明說明 (21)

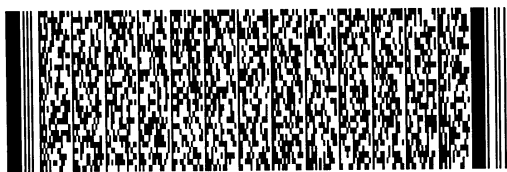
full-search block-matching algorithm," IEEE Trans. on Circuits and Systems for Video Technology, vol. 8, no. 2, pp. 124-127, Apr. 1998.

[7] Y.H. Yeh and C.Y. Lee, "Cost-effective VLSI architectures and buffer size optimization for full-search block matching algorithms," IEEE Trans. on VLSI Systems, vol. 7, no. 3, pp. 345-358, Sep. 1999.



## 圖式簡單說明

- 第一圖為習知連續消除演算法之流程示意圖。
- 第二圖為本發明之全域消除演算法的流程示意圖。
- 第三圖為本發明與全搜尋區塊比對演算法的移動向量相同之百分比曲線圖。
- 第四圖為本發明與全搜尋區塊比對演算法的峰值信號雜訊比之曲線圖。
- 第五圖為本發明之整體硬體架構示意圖。
- 第六圖為本發明之心脈式跳動模組的架構示意圖。
- 第七圖為本發明之樹狀加法器的架構示意圖。
- 第八圖為本發明之平行樹狀比較器的架構示意圖。
- 第九圖為本發明支援先進預測模式之架構示意圖。

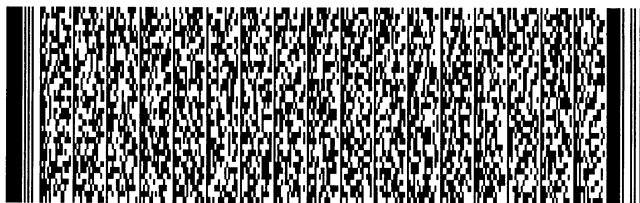




## 四、中文發明摘要 (發明之名稱：用於移動估計的全域消除演算法及其硬體架構)

本發明係提供一種用於移動估計的全域消除演算法及其硬體架構，其係將連續消除演算法中資料流的分支有效去除，使資料流更加順暢且適合硬體實現；且因每個移動向量之處理時間為固定者，初始猜測可以省略，搜尋位置的省略比率係不隨時間改變並可大幅提升。該全域消除演算法具有準確性高之搜尋結果，其結果大多與影像之平均品質為最佳的全搜尋區塊比對演算法相同，有時甚至具有更佳的峰值信號雜訊比，使得本發明具有很高的可靠度。全域消除演算法的硬體架構主要係由心脈式跳動模組、樹狀加法器及樹狀比較器所組成，與其它許多實現全搜尋區塊比對演算法的硬體架構比較，此硬體架構所使用每一邏輯閘的運算能力為最高，且在相同的移動向量生產率下邏

## 英文發明摘要 (發明之名稱：)



四、中文發明摘要 (發明之名稱：用於移動估計的全域消除演算法及其硬體架構)

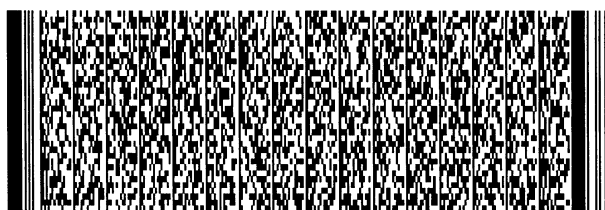
輯開所消耗的功率則最低。

英文發明摘要 (發明之名稱：)



## 六、申請專利範圍

1. 一種用於移動估計的全域消除演算法，包括下列步驟：  
在視訊序列中將參考區塊與每個搜尋位置的候選區塊以較少點數的粗特徵表示；  
比對該參考區塊與所有該候選區塊的粗特徵；  
搜尋粗特徵與該參考區塊最相近的M個候選區塊，再比對該參考區塊與該M個候選區塊的細特徵；以及在該M個候選區塊中選出最小細特徵差異的候選區塊，即可結束整個演算過程。
2. 如申請專利範圍第1項所述之用於移動估計的全域消除演算法，其中該M值係介於1~63之間。
3. 如申請專利範圍第1項所述之用於移動估計的全域消除演算法，其中該最小細特徵差異的候選區塊所對應之移動向量係為估計出來的移動向量。
4. 如申請專利範圍第1項所述之用於移動估計的全域消除演算法，其中該粗特徵係為連續消除演算值及多階層連續消除演算值其中之一者。
5. 如申請專利範圍第1項所述之用於移動估計的全域消除演算法，其中該細特徵差異係為絕對差值總和。
6. 如申請專利範圍第1項所述之用於移動估計的全域消除演算法，其中該M個候選區塊係為具有最小粗特徵的M個搜尋位置。
7. 一種用於移動估計的全域消除演算法之硬體架構，包括：



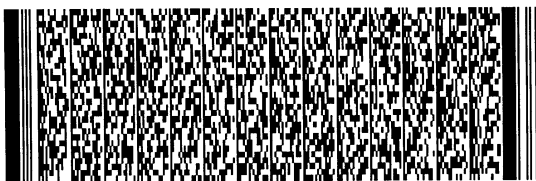
## 六、申請專利範圍

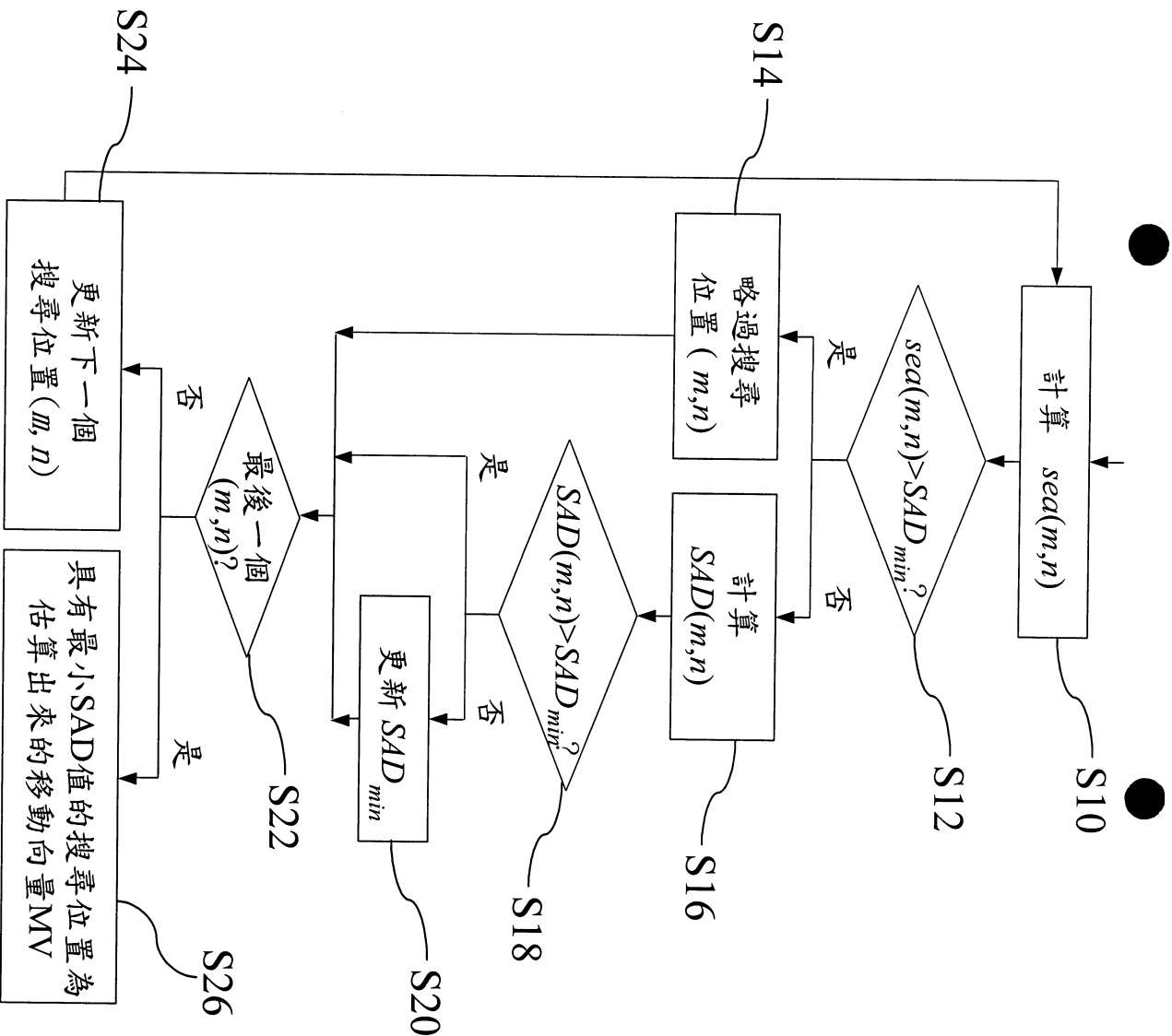
- 一心脈式跳動模組，其係平行計算每個子區塊中的粗特徵；
- 一樹狀加法器，用以平行比對參考區塊及候選區塊的每個粗特徵，且可重複使用來比對該參考區塊與候選區塊的細特徵；
- 至少一樹狀比較器，用以平行搜尋該粗特徵與該參考區塊最相似的M個候選區塊；
- 一控制裝置，其係控制該心脈式跳動模組、該樹狀加法器及該樹狀比較器之作動；以及
- 至少一記憶體，用以儲存該參考區塊與候選區塊的所有資料。
- 8．如申請專利範圍第7項所述之硬體架構，其中該心脈式跳動模組內的每個處理單元係負責計算該區塊中的一個粗特徵。
- 9．如申請專利範圍第7項所述之硬體架構，其中該樹狀比較器係將粗特徵與該參考區塊最相似的M個候選區塊之相似度與其對應的移動向量記錄在暫存器中，再比較輸入之候選區塊的相似度與該M個候選區塊之相似度，找出此(M+1)個候選區塊中與該參考區塊最不相似者；若該最不相似者屬於該暫存器中的某一個候選區塊，則將其替換成該輸入之候選區塊；若有超過一個於該暫存器中的候選區塊同時與該參考區塊最不相似，則只選擇其中一者，將其替換成該輸入之候選區塊。



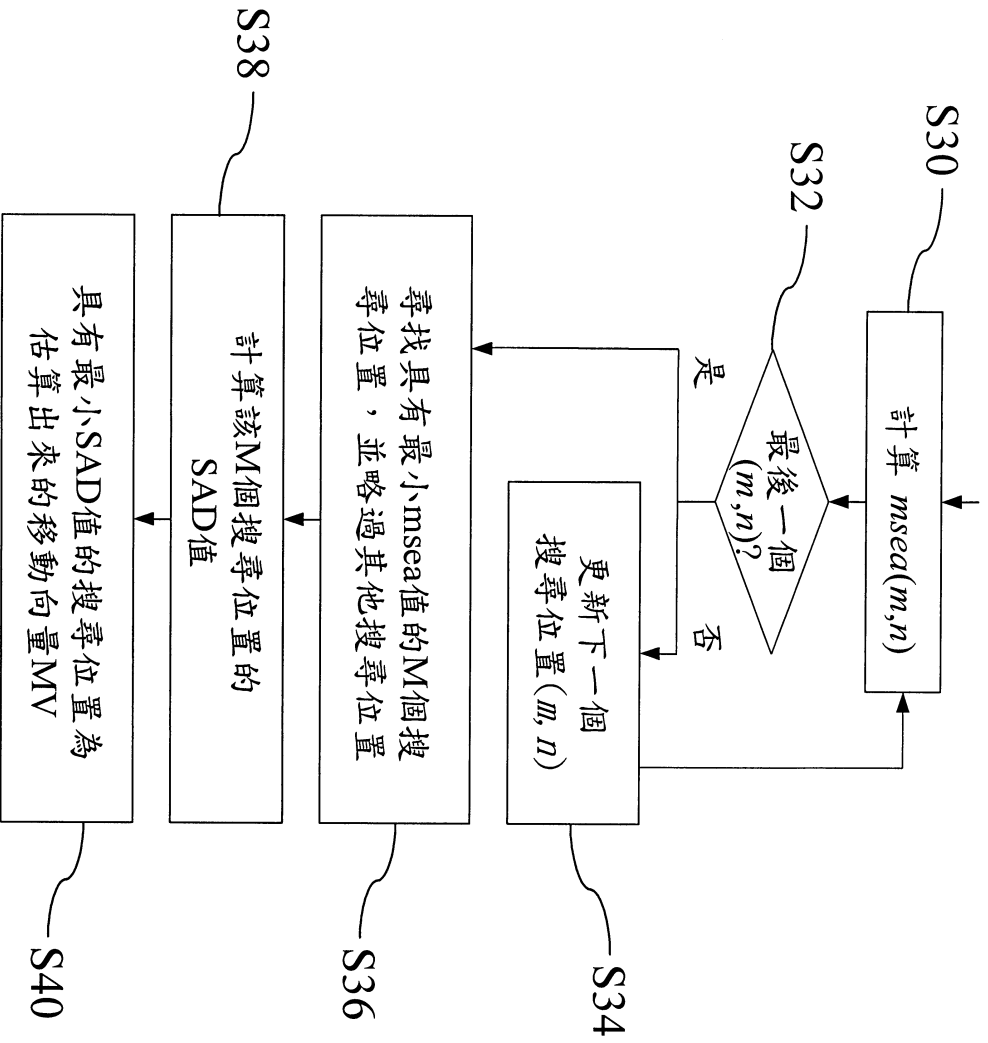
## 六、申請專利範圍

10. 如申請專利範圍第7項或第9項所述之硬體架構，其中該M值係介於1~63之間。
11. 如申請專利範圍第7項所述之硬體架構，其中更可在該樹狀加法器後再加上四個樹狀比較器，以及在該控制單元方面做微小更動，即可支援先進預測模式。
12. 如申請專利範圍第7項所述之硬體架構，其中該粗特徵係為連續消除演算值及多階層連續消除演算值其中之一者。
13. 如申請專利範圍第7項所述之硬體架構，其中該細特徵差異係為絕對差值總和。
14. 如申請專利範圍第7項所述之硬體架構，其中該M個候選區塊係為具有最小粗特徵的M個搜尋位置。

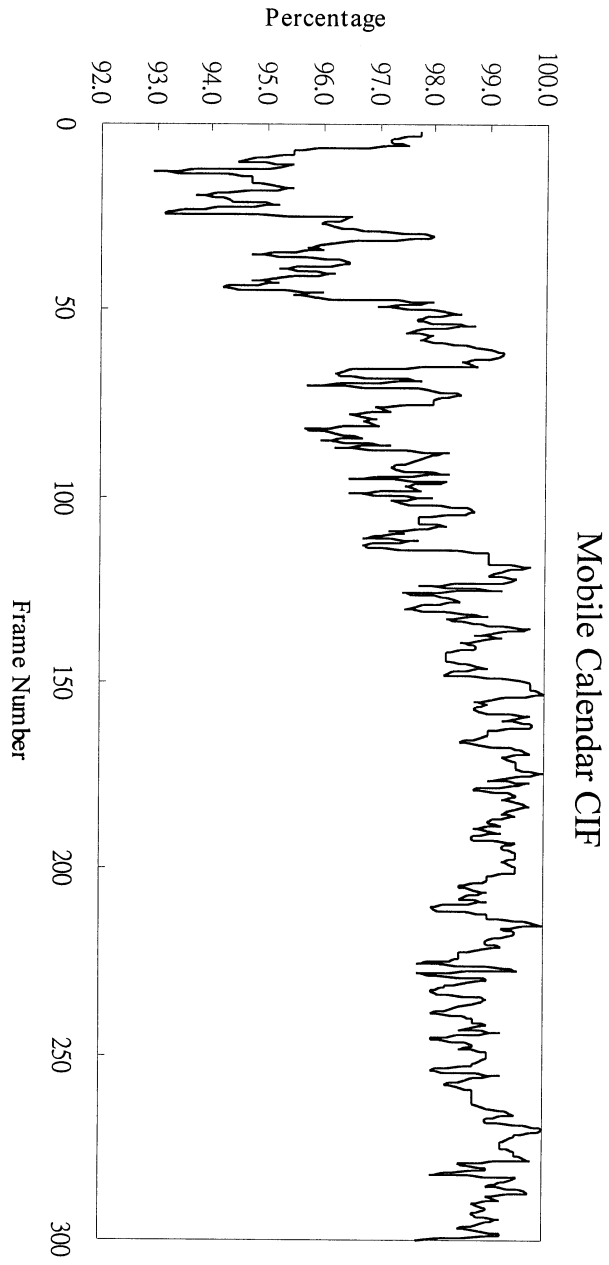




第一圖



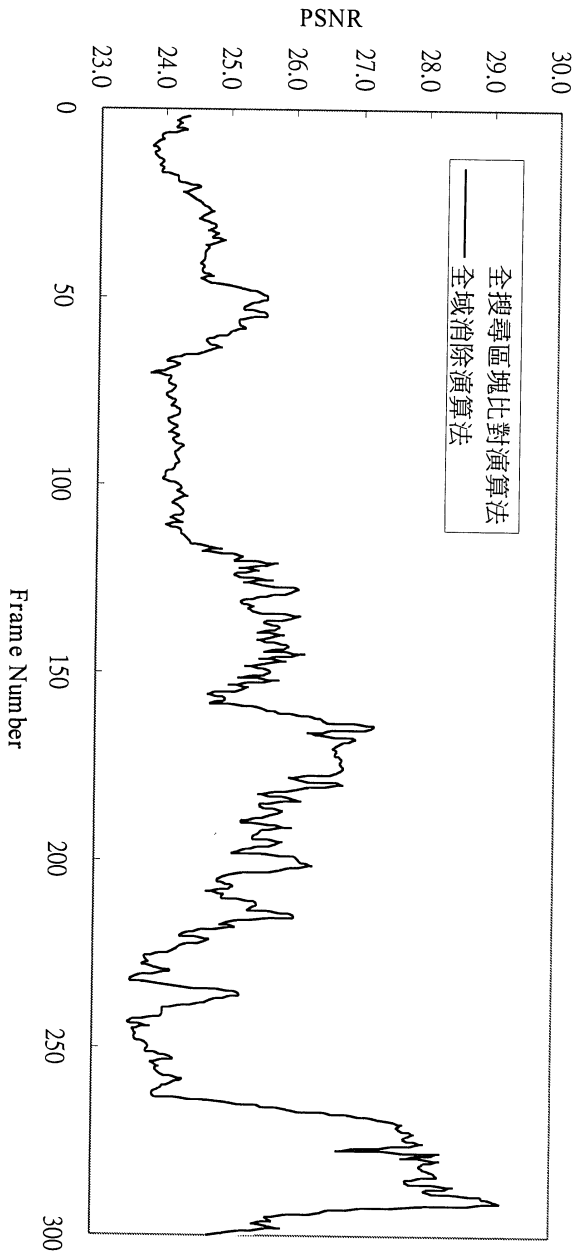
第二圖



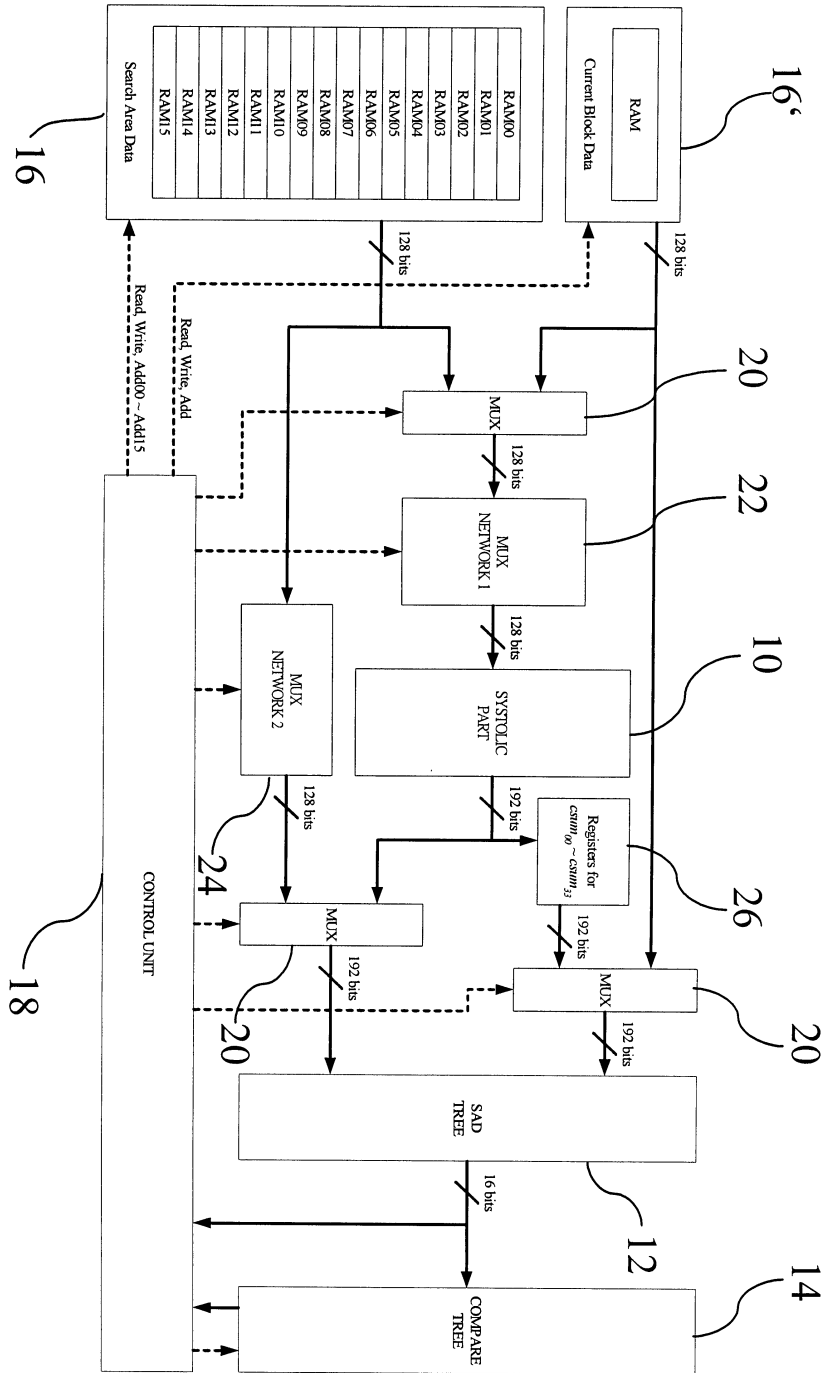
第三圖



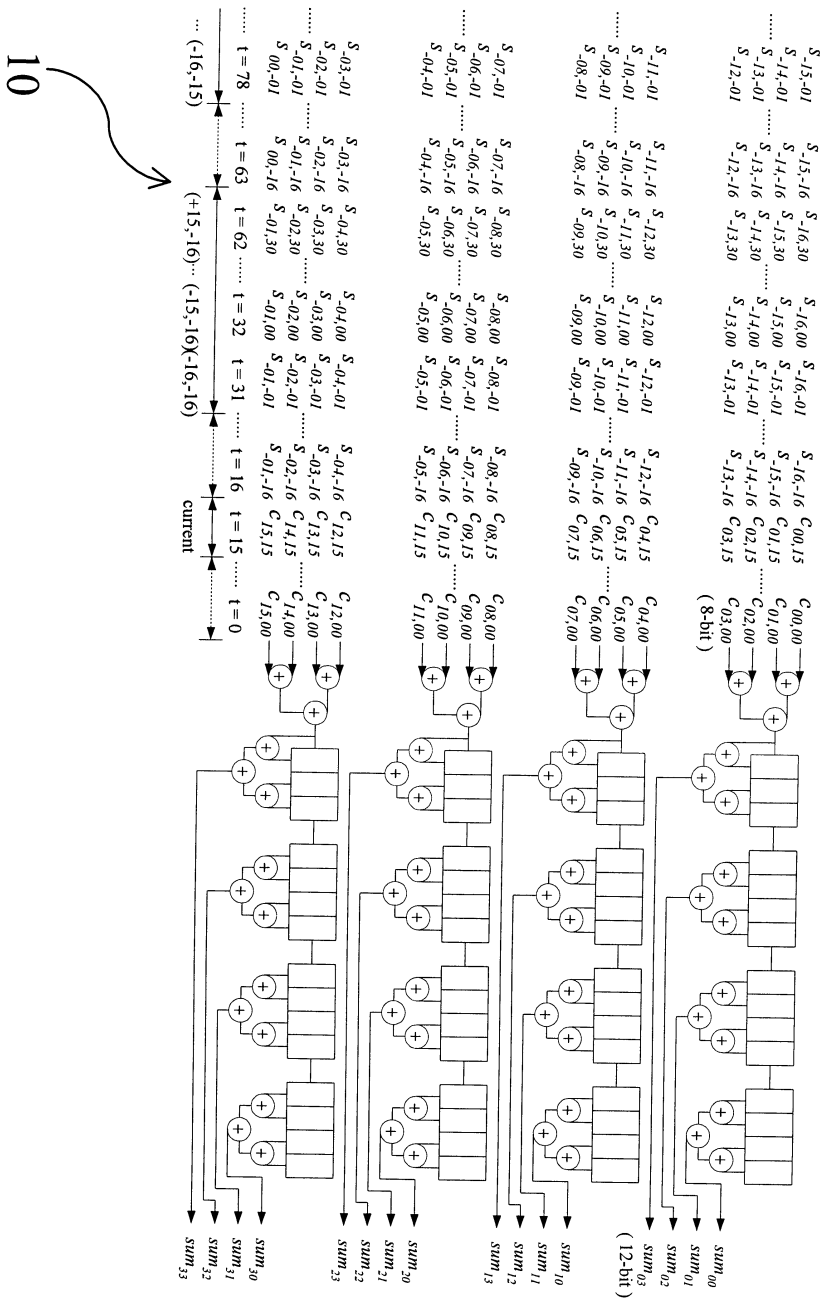
Mobile Calendar CIF



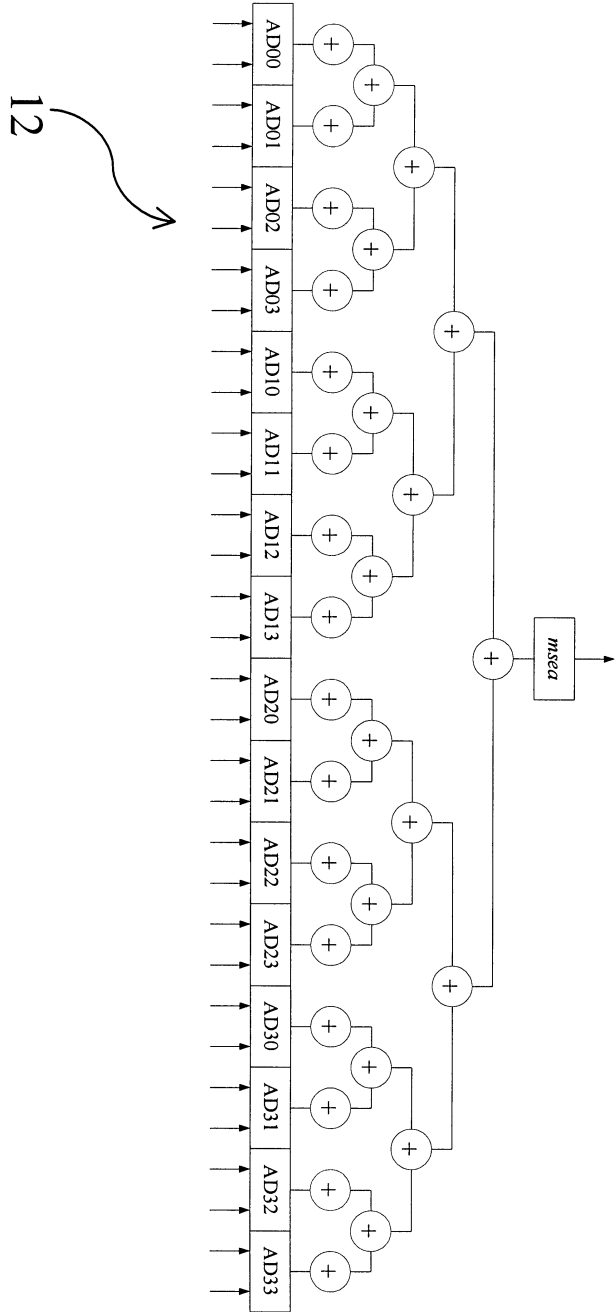
第四圖



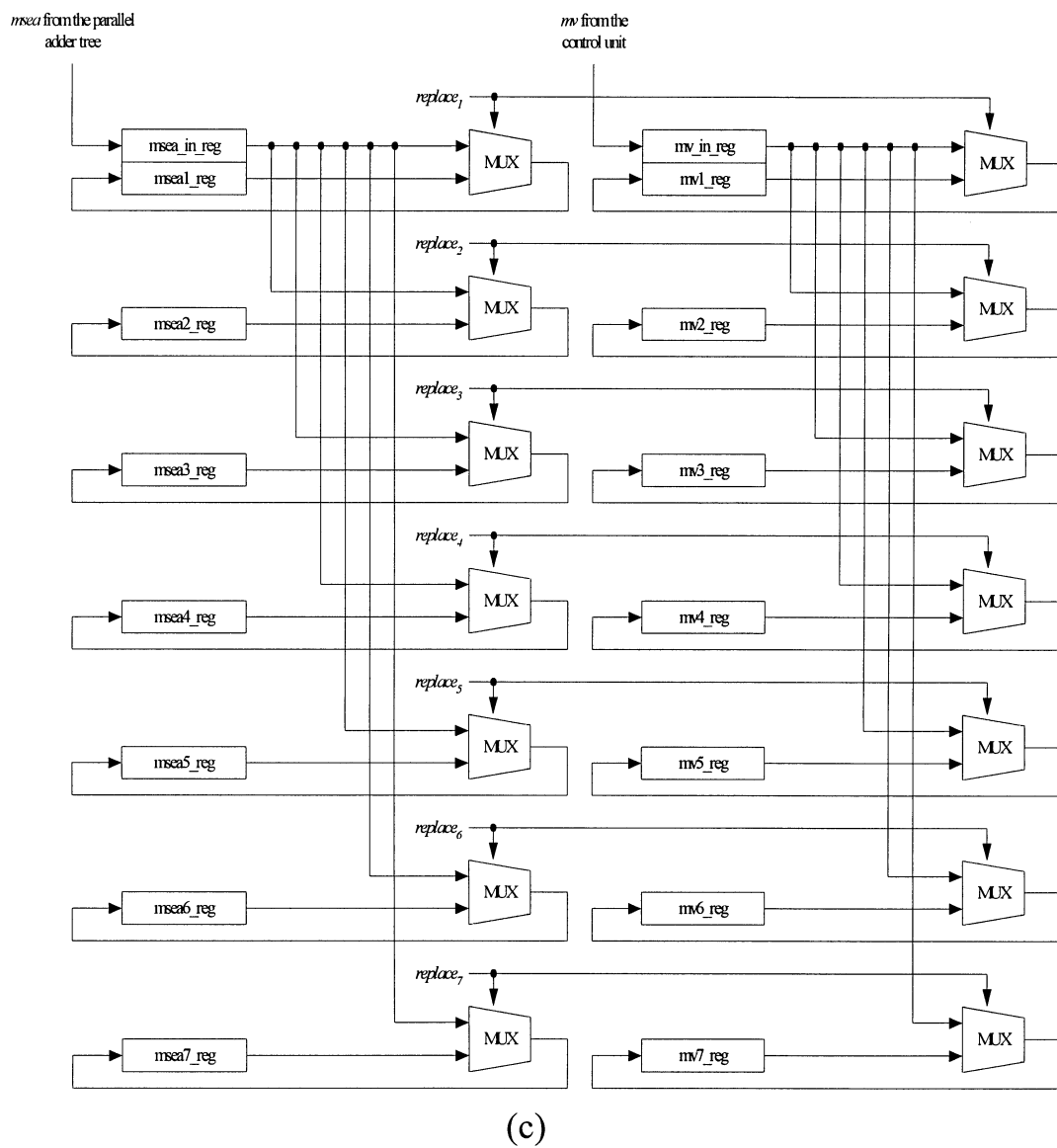
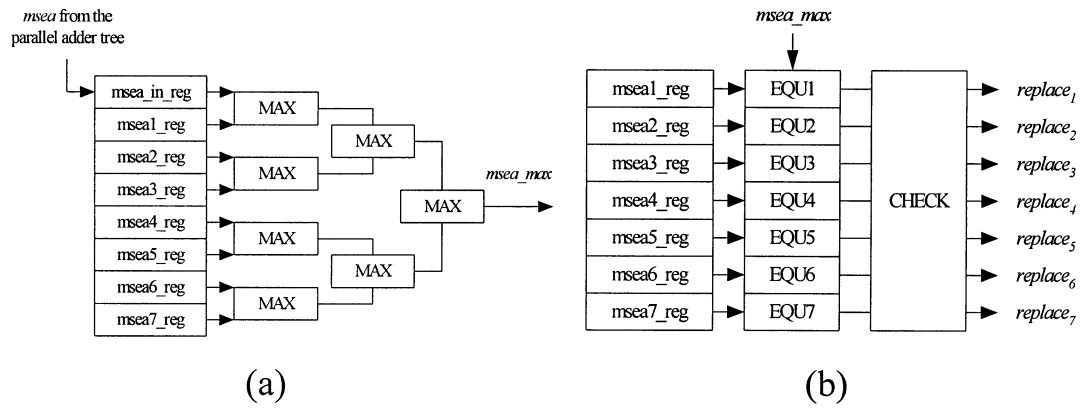
第五圖



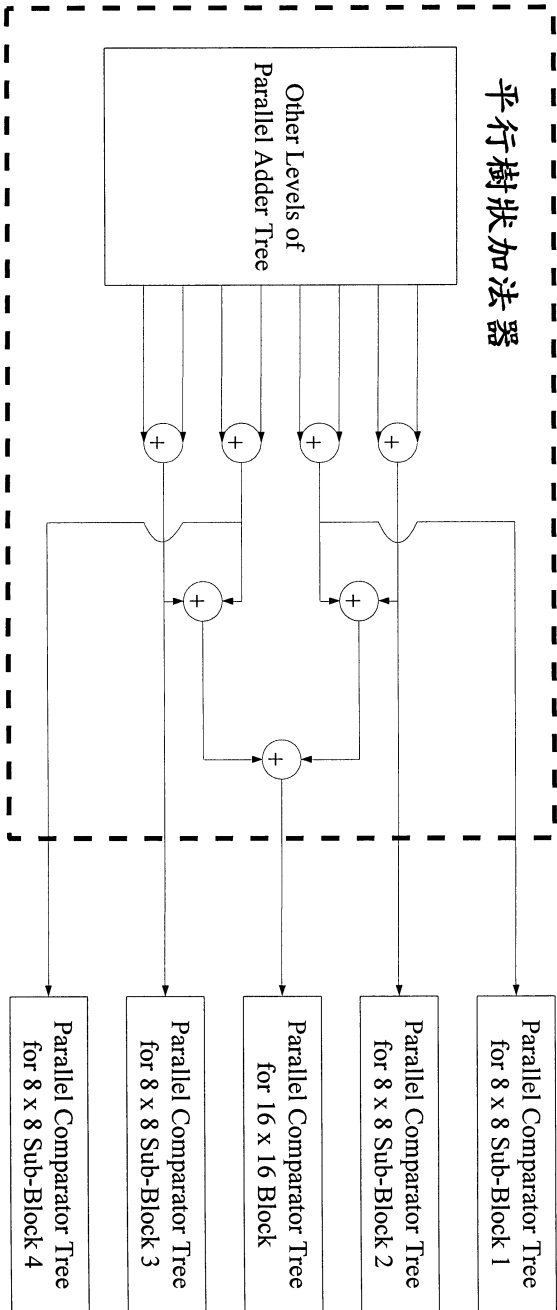
第六圖



第七圖



第八圖



第九圖