

申請日期： 97.6.24	IPC分類 6067 7/14
申請案號： 97118237	

(以上各欄由本局填註)

發明專利說明書

一、 發明名稱	中文	快速傅立葉轉換處理器及其動態調整方法及基數-8之快速傅立葉轉換演算法
	英文	
二、 發明人 (共2人)	姓名 (中文)	1. 李鎮宜 2. 林昱偉
	姓名 (英文)	1. 2.
	國籍 (中英文)	1. 中華民國 TW 2. 中華民國 TW
	住居所 (中文)	1. 新竹市園後街84號4樓 2. 台南市東寧路36號
	住居所 (英文)	1. 2.
三、 申請人 (共1人)	名稱或 姓名 (中文)	1. 國立交通大學
	名稱或 姓名 (英文)	1.
	國籍 (中英文)	1. 中華民國 TW
	住居所 (營業所) (中文)	1. 新竹市大學路1001號 (本地址與前向貴局申請者不同)
	住居所 (營業所) (英文)	1.
	代表人 (中文)	1. 張俊彥
	代表人 (英文)	1.



一、本案已向

國家(地區)申請專利

申請日期

案號

主張專利法第二十四條第一項優先權

無

二、主張專利法第二十五條之一第一項優先權：

申請案號：

無

日期：

三、主張本案係符合專利法第二十條第一項第一款但書或第二款但書規定之期間

日期：

四、有關微生物已寄存於國外：

寄存國家：

寄存機構：

寄存日期：

寄存號碼：

無

有關微生物已寄存於國內(本局所指定之寄存機構)：

寄存機構：

寄存日期：

寄存號碼：

無

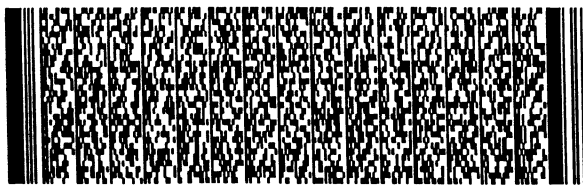
熟習該項技術者易於獲得,不須寄存。



四、中文發明摘要 (發明名稱：快速傅立葉轉換處理器及其動態調整方法及基數-8之快速傅立葉轉換演算法)

本發明提供一種快速傅立葉轉換處理器及其動態調整方法及基數-8之快速傅立葉轉換演算法，其係以陣列預取緩衝器架構的快速傅立葉轉換處理器為基礎，以該陣列預取緩衝器每次之運算大小為區塊大小，根據每一區塊內訊號溢位的情形來動態調整訊號，使訊號不會溢位，以降低運算時所產生的量化誤差；而利用三步驟之基數-8快速傅立葉轉換和重新排程的演算法，係可有系統的錯開複數乘法運算的時間並降低蝴蝶運算單元內的運算複雜度。另外，本發明並提出實現式上述方法及演算法之快速傅立葉轉換處理器。

五、英文發明摘要 (發明名稱：)



六、指定代表圖

(一)、本案代表圖為：第四圖



五、發明說明 (1)

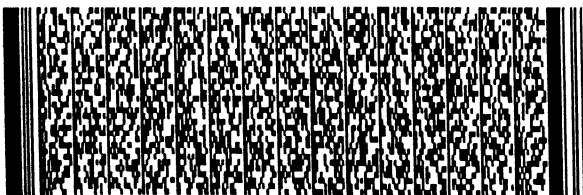
【發明所屬之技術領域】

本發明係有關一種快速傅立葉轉換處理器(Fast Fourier Transform processor, FFT processor)之技術，特別是關於一種快速傅立葉轉換處理器之架構及其動態調整方法及以及基數-8之快速傅立葉轉換演算法。

【先前技術】

在一些特定的無線通訊系統中，需要長點數的快速傅立葉轉換處理器(Fast Fourier Transform processor, FFT processor)的模組，以增加傳輸的頻寬或增加傳輸的效率，如非對稱式數位用戶線路(ADSL)、超高速數位用戶迴路(VDSL)或數位影音廣播系統(DAB/DVB-T)等應用。在數位影音廣播系統中，快速傅立葉轉換處理器(以下簡稱FFT處理器)佔據極大的面積和功率消耗，快速傅立葉轉換的訊號量化雜訊比(signal to quantization noise ratio, SQNR)會隨著快速傅立葉轉換的點數增加而衰減，而為了維持相同的訊號量化雜訊比，高點數FFT處理器所需的字元長度會比低點數FFT處理器所需的字元長度來的多。區塊浮點(block-floating point)是一種動態調整(dynamic scaling)機制，通常應用於減少FFT處理器的量化誤差及所需的字元長度。

第一圖為習知區塊浮點方法之示意圖，在每級運算完畢後，會檢查訊號的最大值是否溢位，並適當的調整大小係數(scale factor)，以避免溢位的情形發生。圈選處是指該點在運算完後發生溢位，而scale factor需向右移以



五、發明說明 (2)

避免溢位。然而，現有的動態調整機制或區塊浮點方法未能對預取緩衝器架構(prefetch buffer based)的快速傅立葉轉換處理器提供最佳化的設計。現今所提出的低硬體複雜度的基數-8快速傅立葉轉換(radix-8 FFT)演算法的方式大都建構在管線式的架構上，在單一記憶體架構的FFT處理器中，只能用串列式的複數乘法器運算來降低處理器的硬體複雜度，但此方式卻犧牲了處理器的效能。

在習知預取緩衝器架構的FFT處理器中，決定溢位的區塊大小係由快速傅立葉轉換的點數來決定，而高基數FFT處理器的硬體複雜度又取決於複數乘法器的數目，因此習知高基數FFT處理器的硬體複雜度極高。

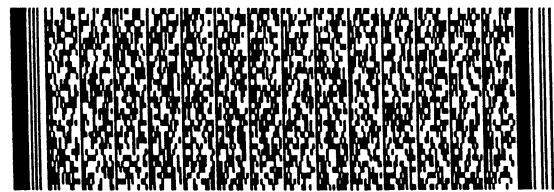
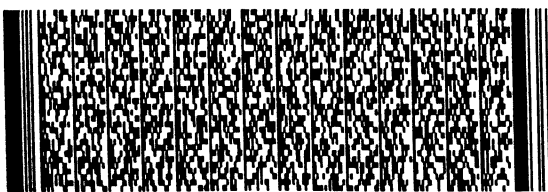
有鑑於此，本發明係針對上述之種種問題，提出一種動態調整之快速傅立葉轉換處理器及方法，以有效克服習知缺失。

【發明內容】

本發明之主要目的，係在提供一種快速傅立葉轉換處理器及其動態調整方法，藉由陣列預取緩衝器的大小來決定溢位的區塊大小，以實現一高訊號量化雜訊比的動態調整機制。

本發明之另一目的，係在提供一種基底-8快速傅立葉轉換演算法，藉由重新排程的方式有效的實現基底-8快速傅立葉轉換，可大幅減少晶片面積，進而達成縮小晶片面積及低功率消耗的功效。

本發明之再一目的，係在提供一種基底-8快速傅立葉



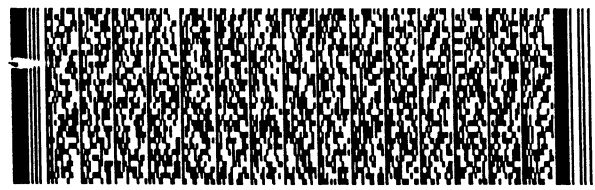
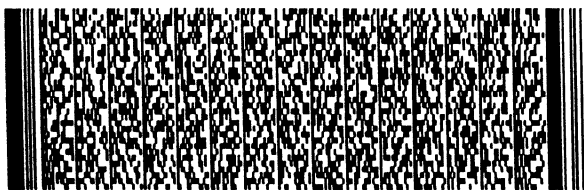
五、發明說明 (3)

轉換演算法，其係可有效減少複數乘法器的數目，以達到低硬體複雜度之目的。

為達到上述之目的，本發明提出一種應用於具有陣列預取緩衝器之快速傅立葉轉換處理器的動態調整方法，此動態調整方法係抓取資料且進行區塊浮點運算，且以該陣列預取緩衝器的大小為基準來決定溢位的區塊大小；當該陣列預取緩衝器內的資料運算完後，根據資料溢位的情形及該區塊大小先動態調整資料的大小，使該資料相對於所屬區塊無溢位後再回存。

另外，本發明提出一種基數-8之快速傅立葉轉換演算法，其係應用於設有複數級(stage)之傅立葉轉換，該基數-8之快速傅立葉轉換演算法係將一基數-8蝴蝶運算單元分解成數個步驟；接著利用重新排程的方式，將原先在蝴蝶運算單元中一次完成執行的數個複數乘法(complex multiplication)分解成該等步驟執行，且將第一步驟執行的部分乘法移至前一級的最末步驟來執行。

本發明並提出實現上述方法之快速傅立葉轉換處理器，其包括一用以控制且處理各元件間之作動的控制單元，控制單元連接一記憶體、一陣列預取緩衝器、一蝴蝶運算器及一正規化單元，記憶體提供儲存資料，以該陣列預取緩衝器為一個區塊，負責向記憶體抓取資料；之後蝴蝶運算器再跟陣列預取緩衝器抓取資料，以做蝴蝶運算，運算完的資料存回陣列預取緩衝器，藉以根據每一次陣列預取緩衝器運算後之資料決定出區塊之大小係數；正規化



五、發明說明 (4)

單元則係在資料被儲存至記憶體前，先依據該大小係數調整該資料之大小，使其於所屬區塊中不會溢位。

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。

【實施方式】

本發明提出一個新的動態調整機制應用於預取緩衝器架構的高點數快速傅立葉轉換處理器。此外，本發明也提出基底-8快速傅立葉轉換演算法配合重新排程的方法，以有效的實現基底-8快速傅立葉轉換。

有別於習知基數-2演算法之乘法複雜的缺失，為了有效節省消耗功率，遂發展出提出基數-8演算法，關於N點FFT ($N = 8^v$)之演算過程如下。

一序列之N-點離散傅立葉轉換(Discrete Fourier Transform, DFT)係定義為

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, k = 0 \dots N-1, \quad (1)$$

其中 $x(n)$ 及 $X(k)$ 係為複數(complex numbe)；複數指數(twiddle factor)為 $W_N = e^{-j2\pi/N}$ 。

首先，令 $n = n_1 + 8n_2$, $k = N/8\hat{k}_1 + k_2$, $n_1, k_1 = 0 \dots 7$, 且 $n_2, k_2 = 0 \dots N/8-1$ ，式(1)可改寫為



五、發明說明 (5)

$$\begin{aligned}
 X(N/8k_1 + k_2) &= \sum_{n_1=0}^7 \sum_{n_2=0}^{N/8-1} x(n_1 + 8n_2) W_N^{(n_1+8n_2)(N/8k_1+k_2)} \\
 &= \sum_{n_1=0}^7 \left\{ \underbrace{\sum_{n_2=0}^{N/8-1} x(n_1 + 8n_2) W_{N/8}^{n_2 k_2}}_{N/8 \text{ point DFT}} \underbrace{W_N^{n_1 k_2}}_{\text{twiddle factor}} \right\} W_8^{n_1 k_1}.
 \end{aligned} \tag{2}$$

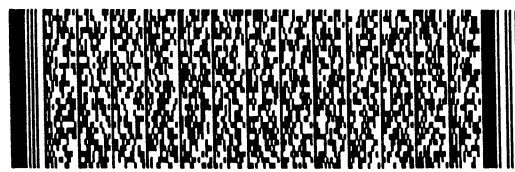
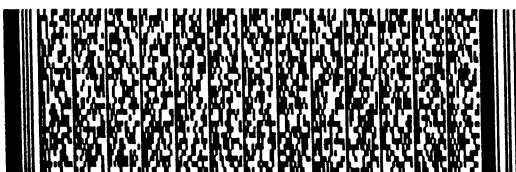
其中

$$BU_{N/8}(n_1, k_2) = \sum_{n_2=0}^{N/8-1} x(n_1 + 8n_2) W_{N/8}^{n_2 k_2}. \tag{3}$$

式(2)可被視為2維DFT，藉由將N/8-點DFT分解成一8-點DFT且遞歸v-1次，其中v係等於，藉此可完成N-點分時(decimation in time, DIT)基數-8FFT演算法。

在式(2)中，該8-點DFT係為基礎運算單元，其係稱為蝴蝶運算(butterfly)，在硬體架構中則稱為FFT運算器的蝴蝶運算單元(butterfly unit)，如第二(a)圖所示。由圖中清楚可知，在該8-點DFT之實施例中，在一蝴蝶運算單元內係設有七個複數乘法器(complex multiplier)，其係相當於28個實數乘法器。為了更有效率地執行基數-8FFT演算法，本發明進一步將基數-8演算法中之蝴蝶運算分解成三個步驟且使用基數-2將其映射至基數-8演算法之蝴蝶運算中。當使用一3維線性指數映射時， n_1 及 k_1 可定義為

$$\begin{aligned}
 n_1 &= \gamma_1 + 2\gamma_2 + 4\gamma_3 \quad \gamma_1, \gamma_2, \gamma_3 \in \{0, 1\}. \\
 k_1 &= 4\nu_1 + 2\nu_2 + 1\nu_3 \quad \nu_1, \nu_2, \nu_3 \in \{0, 1\}.
 \end{aligned} \tag{4}$$



五、發明說明 (6)

在式(4)之條件下，則式(2)成為

$$\begin{aligned}
 & X\left(\frac{N}{2}v_1 + \frac{N}{4}v_2 + \frac{N}{8}v_3 + k_2\right) \\
 &= \sum_{r_3=0}^1 \sum_{r_2=0}^1 \sum_{r_1=0}^1 \{BU_{N/8}(\gamma_1, \gamma_2, \gamma_3, k_2) W_N^{(n+2r_2+4r_3)k_2} W_8^{4nr_1} W_8^{2nr_2} W_8^{4r_2v_2} W_8^{(n+2r_2)v_3} W_8^{4r_3v_3}\} \\
 &= \sum_{r_1=0}^1 \sum_{r_2=0}^1 \sum_{r_3=0}^1 \left\{ \underbrace{\left\{ TU_{N/8}(\gamma_1, \gamma_2, \gamma_3, k_2) W_2^{nr_1} W_4^{nr_2} W_2^{r_2v_2} \right\}}_{1st \ step} \right\} \underbrace{W_8^{(n+2r_2)v_3} W_2^{r_3v_3}}_{2nd \ step} \underbrace{\phantom{W_8^{(n+2r_2)v_3} W_2^{r_3v_3}}}_{3rd \ step} \phantom{W_8^{(n+2r_2)v_3} W_2^{r_3v_3}} \quad (5)
 \end{aligned}$$

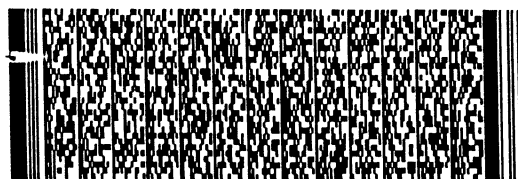
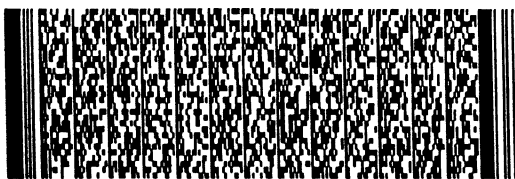
其中

$$TU_{N/8}(\gamma_1, \gamma_2, \gamma_3, k_2) = BU_{N/8}(\gamma_1, \gamma_2, \gamma_3, k_2) W_N^{(n+2r_2+4r_3)k_2} \quad (6)$$

在式(5)中，係可使用基數-2指數映射來將-8-點DFT分成三步驟。如第二(b)圖所示，其顯示三步驟之基數-8FFT的蝴蝶狀態，由於在第三步驟時之簡易乘法(trivial multiplication)及可容易由六移位器及四加法器來實現，故基數-8演算法可大幅減少複數乘法(complex multiplication)之次數。其中關於乘法器以移位器及加法器實現之相關文獻請參閱Lihong Jia等人所發表之"A new VLSI-oriented FFT algorithm and implementation" (參附件一)，於此不再贅述。

在3-步驟基數-8演算法中之複數乘法的原始時間排程如第三(a)圖所示，其中T1、T2及T3各代表每一步驟的時槽，而該矩形代表每一時槽中之複數乘法。

為了縮小複數乘法器之數目，本發明提出在3-步驟基



五、發明說明 (7)

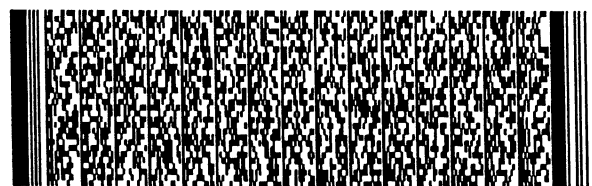
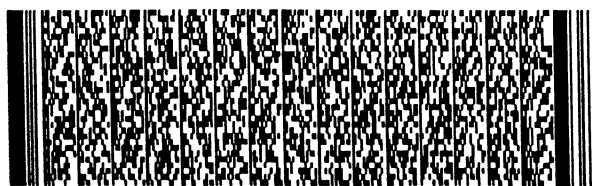
數-8 FFT 演算中進行重新排程之演算法，其係應用於點數為 $8n$ 的快速傅立葉轉換，此演算法提供系統性的方式來移動某些複數指數至前一級(previous stage)，且平衡該蝴蝶之三時槽中的複數乘法，如第四圖所示，其中之黑點係指在該點乘以該複數指數，而所有隔離的複數指數係置於該些蝴蝶之第三步驟。因此，在該蝴蝶的每一時槽中，最多僅有四個複數乘法，且於重新排程後，在3-步驟基數-8演算中只需要使用四個複數乘法器。第三(b)圖及第三(c)圖顯示重新排程後之複數乘法的時程。

請參第五圖，在重新排程後，由於該些複數指數係同時位於蝴蝶之第一及第三步驟，因此某些平衡運算模式需要被加入。在重新排程之步驟後，更包括加入二平衡運算模式之步驟，其係在蝴蝶運算中，利用第一平衡運算模式乘以下一級中第一步驟的複數指數，且利用第二平衡運算模式乘以前一級中最末步驟的複數指數；其中，如圖所示，在此蝴蝶運算中，模式A及模式B係乘以第一步驟之複數指數；而其他二運算模式，模式C及模式D，則乘以第三步驟的複數指數。

為了使運算器中之八筆資料在蝴蝶之每一步驟中以相同模式運算，以減少運算複雜度，本發明提出一 N -點FFT重新排程之演算法，其係依據FFT之級別與蝴蝶運算之群組數目來決定被移動之群組及其所移至之階級(stage)。

首先，定義：

1. N 點FFT之級別(stage)係從1至 L ()。



五、發明說明 (8)

2. 屬於第L級中之群組的數目係從0至。
3. 在第L級中的每一群組的蝴蝶數係0至 $8(L-1)-1$ 。
4. BU_1係為蝴蝶之第一步驟中的運算模式，且BU_3係為蝴蝶之第三步驟中的運算模式。

該N-點FFT之重新排程演算如下，請配合第六圖之流程圖所示：

對於(從1至L之各階級)

開始

若($1 \leq \text{階級} \leq L-1$)

開始

若(群組的數目為偶數)

BU_3=mode C;

反之

BU_3=mode D;

結束

若($2 \leq \text{階級} \leq L$)

開始

若(蝴蝶數等於或少於每一群組前半部的蝴蝶數)

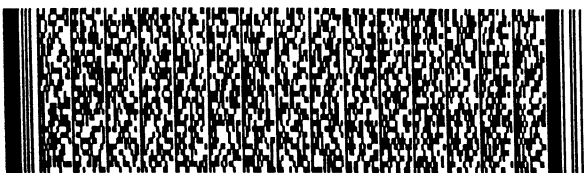
BU_1=mode A;

反之

BU_1=mode B;

結束

結束

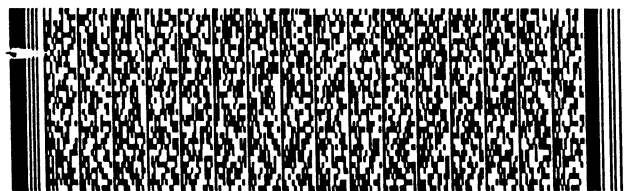
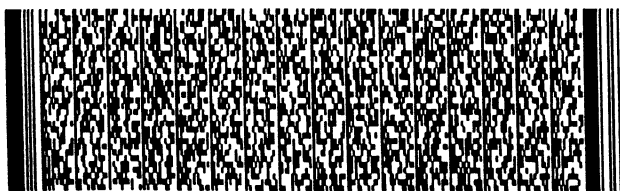


五、發明說明 (9)

動態調整方法

為了維持定點(fixed point)之FFT的資料正確性，FFT處理器之內部字元長度(wordlength)通常係大於輸入資料之字元長度，以達到較高之訊雜比(signal to noise ratio, SNR)，特別是在一長點數之FFT中。區塊浮點(block-floating point, BFP)是一種動態調整(dynamic scaling)方式，通常應用於減少FFT處理器的量化誤差及所需的字元長度。在習知的區塊浮點中，在開始第N+1級之計算前，最大值係被偵測出，以此值決定大小係數，且所有計算的結果係以第N級中之一大小係數(scale factor)來調整。

本發明之快速傅立葉轉換處理器之動態調整方法係應用於預取緩衝器之快速傅立葉轉換器，本發明之動態調整方法所適用之硬體架構如第七圖所示，一預取緩衝器10係連接於一記憶體12及一蝴蝶運算單元(butterfly unit, BU)14之間。以下將以前述架構來輔助說明本發明之動態調整方法，此方法包括下列步驟：首先，由預取緩衝器10抓取記憶體12中之一筆資料，接著由蝴蝶運算單元14進行區塊浮點運算，此時係以該預取緩衝器10的大小為基準，決定溢位的區塊大小；當每一次區塊之資料運算完成時，即根據該資料的溢位量決定所屬區塊之大小係數，以便後續可依據該大小係數調整該區塊內資料之大小，使資料無溢位後再回存至記憶體12。其中，調整資料大小之方式係藉由位移小數點之位置而為之。



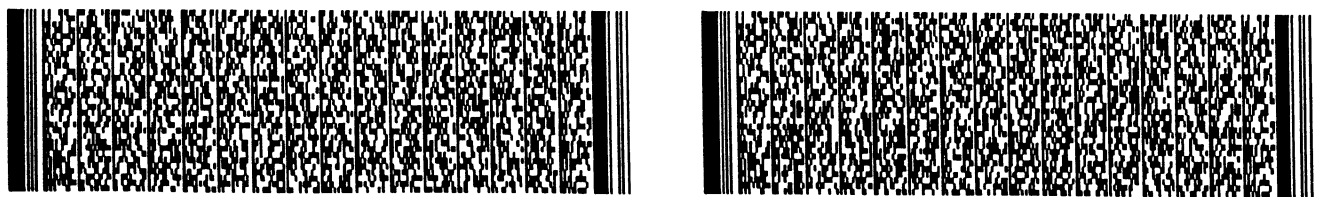
五、發明說明 (10)

本發明的區塊浮點方法可藉由預取緩衝架構之FFT處理器來執行，其藉由在FFT演算法中增加大小係數(scale factor)及區塊數目二參數，以有效改善訊號雜訊比(SQNR)。第八圖為16-點FFT中具有四點之區塊大小的實施例，當每一次區塊之資料運算完成時，即可決定出區塊之大小係數，且在開始運算下一次區塊前，將前一次區塊中之資料依據該大小係數來調整大小，使其不會溢位。其中，所有大小係數係被儲存在一表中，以便在下次資料運算時可使用。

請再參第九圖所示，其係為定點(fixed-point)、習知區塊浮點及本方法此三種方式處理訊號之實驗成效，由於SQNR係與輸入資料數值相關，本發明之實驗建立一8K模式之DVB-T系統平台，且所有資料係由此系統平台產生，該區塊之大小為64點，由實驗結果明顯可知，本方法可有效降低量化錯誤率，且在相同的字元長度下，比其他二種方式具有更高之SQNR。

在了解本發明對於快速傅立葉轉換處理器之動態調整方法及其功效之後，接下來，將以一硬體實施例說明可實現該方法及功效之具體架構。

如第十圖所示，為本發明之快速傅立葉轉換處理器的結構方塊圖，一快速傅立葉轉換(FFT)處理器20包括一控制單元22，控制單元22連接有一記憶體24、一陣列預取緩衝器26、四複數乘法器28、一蝴蝶運算器30及一正規化單元32，以利用控制單元22控制且處理各元件間之作動。記

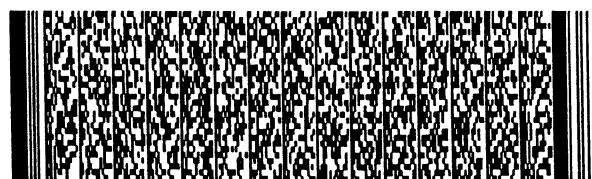
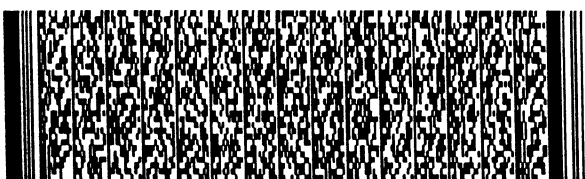


五、發明說明 (11)

憶體24係提供儲存資料，並以該陣列預取緩衝器26於不同運算時點之大小為一區塊，進而依不同時點產生複數不同之區塊，且以每一次陣列預取緩衝器26的運算大小為一區塊，陣列預取緩衝器26負責由記憶體24抓取資料；該四複數乘法器28係連接於陣列預取緩衝器26與蝴蝶運算器30之間，且連接一複數指數儲存單元40，複數乘法器28係用以由複數指數儲存單元40讀出所乘之複數指數值後，將陣列預取緩衝單元26內之資料依據該複數指數進行乘法運算，而後再將資料傳送至蝴蝶運算器30；蝴蝶運算器30係由四個蝴蝶運算單元42組成，用以將經乘法運算後之資料進行蝴蝶運算，且將運算後之資料存回陣列預取緩衝器26中所對應區塊，當陣列預取緩衝器的資料運算完畢後，決定出區塊之大小係數，每一次區塊之大小係數則儲存在一記憶表38中；正規化單元32則係在每一區塊資料運算完成而被儲存至記憶體24之前來調整(scale)資料，亦即當每一次陣列預取緩衝器26內之資料運算完成時，陣列預取緩衝器26係可決定出區塊之大小係數，且在下一區塊之資料開始運算前，利用正規化單元32將前一區塊中之資料依據大小係數來進行調整(scaling)，使其不會溢位。

另外，在該等乘法器28與蝴蝶運算器30之間設有二緩衝器34，用來暫存資料，減少陣列預取緩衝器26的讀取次數；且有一共用匯流排36係連接陣列預取緩衝器26、蝴蝶運算器30及正規化單元32。

在本發明中，藉由在FFT處理器20中使用三階(three-

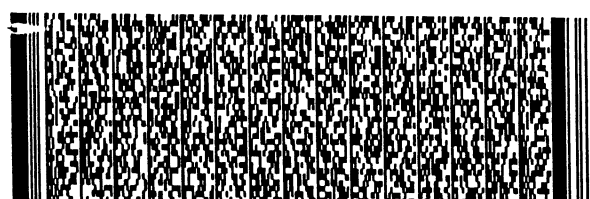
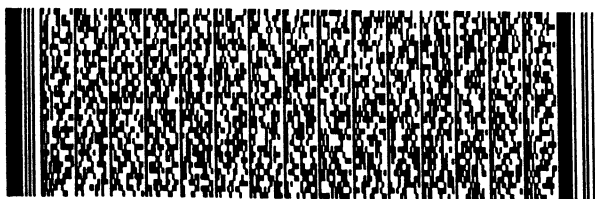


五、發明說明 (12)

level) 記憶體架構，係可增加資料處理效率。第一階 (first level) 係記憶體24，其係為主記憶體，分成八個資料庫(bank)，以允許多路資料可同時被存取且該記憶體24之大小係為8K點；陣列預取緩衝器26係為第二階，其係設有64點以進行基數-八之演算；第三階係為二緩衝器34，且每一緩衝器34係為8點(8 point)者。透過在此三階記憶體中適當之重新排程，單埠記憶體可被使用作為第一及第二階(level)且不會有任何效能降低。故在此設計中，藉由使用動態調整方式，實數部分及虛數部分之字元長度係為11字元(bits)。

蝴蝶運算單元42係為FFT處理器20中之核心單元，該蝴蝶運算單元42包括一簡易乘法器(trivial multiplier)，其係用以處理 $-j, W_8^1, W_8^3$ 及一複數加法器/減法器；複數指數儲存單元40通常為唯讀記憶體(ROM)，其係用以儲存FFT複數指數，只有1/8週期的餘弦及正弦波形係儲存在ROM中，而其他週期波形可藉由儲存的數值來重新建構。當資料讀出或寫入於該些緩衝器34，資料係乘以複數指數，且該等緩衝器34內之資料在藉由蝴蝶運算單元42運算時需要三個週期(cycles)，以執行3-步驟基數-8演算法。

其中，陣列預取緩衝器26之架構如第十一(a)圖及第十一(b)圖所示，第0至7行係為第1級之八個蝴蝶，第0至7列係為第2級之八個蝴蝶，配合第四圖所示，於進行每一

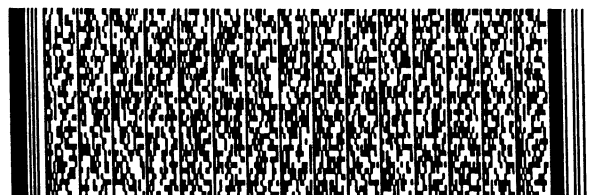
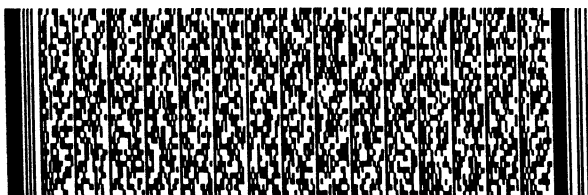


五、發明說明 (13)

次讀/寫時，在陣列預取緩衝器26中之八組資料係同時以水平或垂直方向被讀出或寫入；在此架構中，共用匯流排36係可減少晶片實作上繞線之複雜度。

當資料依序被下載到記憶體26後，FFT處理器20便開始利用3-步驟基數-8演算法來執行64點FFT之運算。在第1級(stage)時，在陣列預取緩衝器26的資料依序以行之方向被讀出，如第十一(a)圖所示，經過乘法器28的運算，資料存入緩衝器34並和蝴蝶運算器30運算，所有計算後之資料係被回存進陣列預取緩衝器26中之相同位址處；在第2級(stage 2)時，在陣列預取緩衝器26的資料依序以列的方向被讀出，且該資料運算流程同第1級，只是資料在緩衝器34運算完後，係被送到正規化單元32，係等待正規化單元32進行後續之調整動作。當第0列資料已被下載至正規化單元32，新資料將會從記憶體24下載至第0列，如第十一(b)圖所示；在下一64點中，由於更新資料係為列方向，第1級之方向將會改變成列方向。

第十二圖顯示若使用單埠記憶體時，緩衝器內之資料以BU重新排程且以預取緩衝器交換之關係圖，如圖所示，白色矩形係代表在該等緩衝器34中資料之運算時間；灰色矩形則代表在陣列預取緩衝器26中交換資料或下載至正規化單元32所花之時間。由結果明顯可知，在此排程中係無閒置(stall)產生；同理，在64-點FFT之第2級時，資料係從第一階下載至第二階，且在64-點FFT之第1級時，該資料係再從正規化單元32儲存至第一階。因此，單一埠記憶

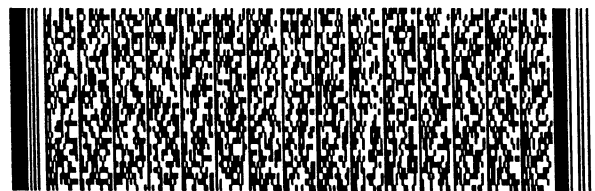
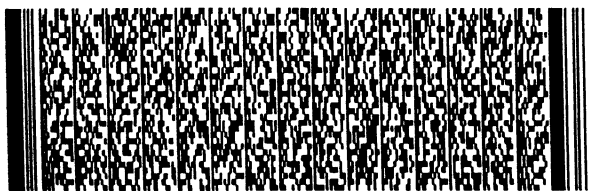


五、發明說明 (14)

體係可被使用而不會導致任何效能降低。因此在模式為八千點的數位影音廣播系統模擬下，本發明可始實部及虛部所需的字元長度比定點時所需各少四個位元，在八千點的快速傅立葉轉換中，上述動態調整方法約可節省記憶體六萬四千位元。

本發明以陣列預取緩衝器架構的快速傅立葉轉換處理器為基礎，根據每一區塊內訊號溢位的情形做動態調整，不同以往的是以陣列預取緩衝器的大小作為區塊大小，來決定區塊內的值是否需要溢位，由於決定溢位的區塊大小比傳統的方式小，因此能提高訊號量化雜訊比，並降低運算時所產生的量化誤差。另外，利用三步驟之基數-8快速傅立葉轉換和重新排程的演算法，將蝴蝶運算單元內所需運算的複數乘法重新排程，避免所有複數乘法在同一時間運算，以此方式來減少複數乘法器的個數，不僅具有低硬體複雜度之優點，且可大幅減少晶片面積，進而達成縮小晶片面積及低功率消耗的功效。

以上所述係藉由實施例說明本發明之特點，其目的在使熟習該技術者能瞭解本發明之內容並據以實施，而非限定本發明之專利範圍，故，凡其他未脫離本發明所揭示之精神所完成之等效修飾或修改，仍應包含在以下所述之申請專利範圍中。



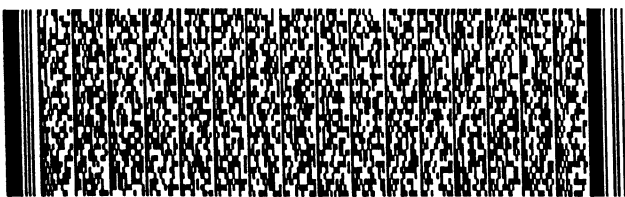
圖式簡單說明

圖式說明：

- 第一圖為習知區塊浮點方法之示意圖。
- 第二(a)圖為基數-8 FFT演算法的蝴蝶示意圖。
- 第二(b)圖為本發明之3-步驟基數-8 FFT演算法的蝴蝶示意圖。
- 第三(a)圖為本發明之複數乘法於重新排程前的時間排程圖。
- 第三(b)圖及第三(c)圖分別為本發明之複數乘法於重新排程後的時間排程實施例。
- 第四圖為本發明重新排程演算法之蝴蝶及時程圖。
- 第五圖為本發明於重新排程方法中之蝴蝶運算單元的運作模式示意圖。
- 第六圖為本發明之重新排程流程圖。
- 第七圖為本發明之動態調整方法所適用之硬體架構。
- 第八圖為本發明之區塊浮點方法的區塊示意圖。
- 第九圖為本發明與定點、習知區塊浮點之SNR測試結果。
- 第十圖為本發明之FFT處理器的結構方塊圖。
- 第十一(a)圖及第十一(b)圖分別為本發明之陣列預取緩衝器在第1級及第2級時之運作示意圖。
- 第十二圖為本發明之陣列預取緩衝器內資料的時間排程圖。

圖號說明：

- | | |
|-----------|--------|
| 10 預取緩衝器 | 12 記憶體 |
| 14 蝴蝶運算單元 | |



圖式簡單說明

- | | | | |
|----|-----------------|----|----------|
| 20 | 快速傅立葉轉換(FFT)處理器 | | |
| 22 | 控制單元 | 24 | 記憶體 |
| 26 | 陣列預取緩衝器 | 28 | 複數乘法器 |
| 30 | 蝴蝶運算器 | 32 | 正規化單元 |
| 34 | 緩衝器 | 36 | 共用匯流排 |
| 38 | 記憶表 | 40 | 複數指數儲存單元 |
| 42 | 蝴蝶運算單元 | | |



六、申請專利範圍

1. 一種快速傅立葉轉換處理器之動態調整方法，其係應用於具有一陣列預取緩衝器之快速傅立葉轉換處理器，該動態調整方法包括下列步驟：

抓取資料且進行區塊浮點運算，以該陣列預取緩衝器的大小為基準，決定溢位的區塊大小；以及

當該陣列預取緩衝器內的資料運算完後，根據資料溢位的情形及該區塊大小先動態調整該資料的大小，使該資料相對於所屬區塊無溢位後再回存該資料。

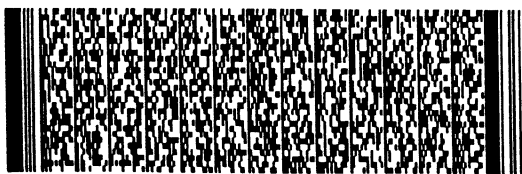
2. 如申請專利範圍第1項所述之快速傅立葉轉換處理器之動態調整方法，其中，動態調整該資料大小之步驟，係每次當該陣列預取緩衝器之資料運算完成時，根據該資料的溢位量決定該區塊之大小係數，以依據該大小係數調整該區塊內資料之大小，使該資料無溢位。

3. 如申請專利範圍第2項所述之快速傅立葉轉換處理器之動態調整方法，其係在開始運算下一次區塊之資料前，動態調整前一次區塊內資料之大小。

4. 如申請專利範圍第1項所述之快速傅立葉轉換處理器之動態調整方法，其中，調整該資料大小之方式係藉由位移小數點之位置。

5. 一種基數-8之快速傅立葉轉換演算法，應用於設有複數級(stage)之傅立葉轉換，該基數-8之快速傅立葉轉換演算法包括下列步驟：

將一基數-8蝴蝶運算器中每一該級之運算分解成數個步驟(step)；以及



六、申請專利範圍

利用重新排程的方式，將原先在該蝴蝶運算器同一級中一次完成執行的數個複數乘法(complex multiplication)分解成該等步驟執行，且將第一步驟執行的部分乘法移至前一級的最末步驟來執行。

6. 如申請專利範圍第5項所述之基數-8之快速傅立葉轉換演算法，其係將基數-2映射至基數-8演算法之蝴蝶運算中。

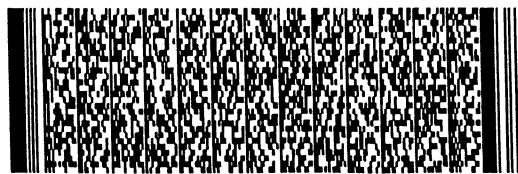
7. 如申請專利範圍第5項所述之基數-8之快速傅立葉轉換演算法，其中，進行該重新排程的步驟時，係移動下一級之第一步驟的複數指數，使其同時存在於前一級之最末步驟中。

8. 如申請專利範圍第5項所述之基數-8之快速傅立葉轉換演算法，其中，在重新排程之步驟後，更包括加入二平衡運算模式之步驟，其係在蝴蝶運算中，利用第一平衡運算模式乘以下一級中第一步驟的複數指數，且利用第二平衡運算模式乘以前一級中最末步驟的複數指數。

9. 如申請專利範圍第8項所述之基數-8之快速傅立葉轉換演算法，其中，該第一及第二平衡運算模式分別包含數模式。

10. 如申請專利範圍第5項所述之基數-8之快速傅立葉轉換演算法，其中，該重新排程之方式係依據該等級之級別與蝴蝶運算之群組數目來決定被移動之群組及其所移至之階級(stage)。

11. 一種快速傅立葉轉換處理器，包括：



六、申請專利範圍

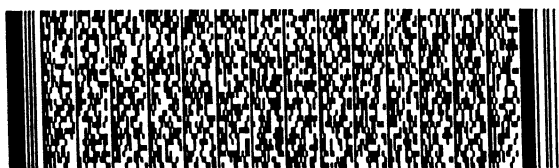
- 一 控制單元，用以控制且處理各元件間之作動；
- 一 記憶體，其係連接該控制單元，提供儲存資料；
- 一 陣列預取緩衝器，其係負責由該記憶體抓取資料，並以該陣列預取緩衝器於不同運算時點之大小為一區塊，進而依不同時點產生複數不同之區塊；
- 數個乘法器，其係連接該陣列預取緩衝器，以將該資料進行乘法運算；
- 一 蝴蝶運算器，其係連接該等乘法器，用以將該等區塊中之資料進行蝴蝶運算，且將運算後之資料存回所屬區塊，藉以由該陣列預取緩衝器根據運算後之資料決定出每一該區塊之大小係數；以及
- 一 正規化單元，其係在資料被儲存至該記憶體前，先依據該大小係數調整該資料之大小，使其於所屬區塊中不會溢位。

12. 如申請專利範圍第11項所述之快速傅立葉轉換處理器，其中，當該陣列預取緩衝器內之資料運算完成時，其係可決定出所屬之該區塊之大小係數，且在下一區塊之資料開始運算前，利用該正規化單元將前一區塊中之資料依據進行調整(scaling)，使其不會溢位。

13. 如申請專利範圍第11項所述之快速傅立葉轉換處理器，其中，該蝴蝶運算器係由複數蝴蝶運算單元組成。

14. 如申請專利範圍第11項所述之快速傅立葉轉換處理器，該乘法器為複數乘法器。

15. 如申請專利範圍第11項所述之快速傅立葉轉換處理



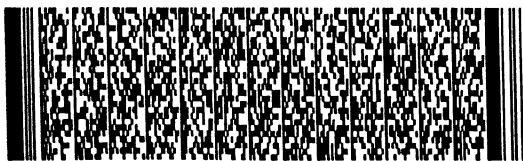
六、申請專利範圍

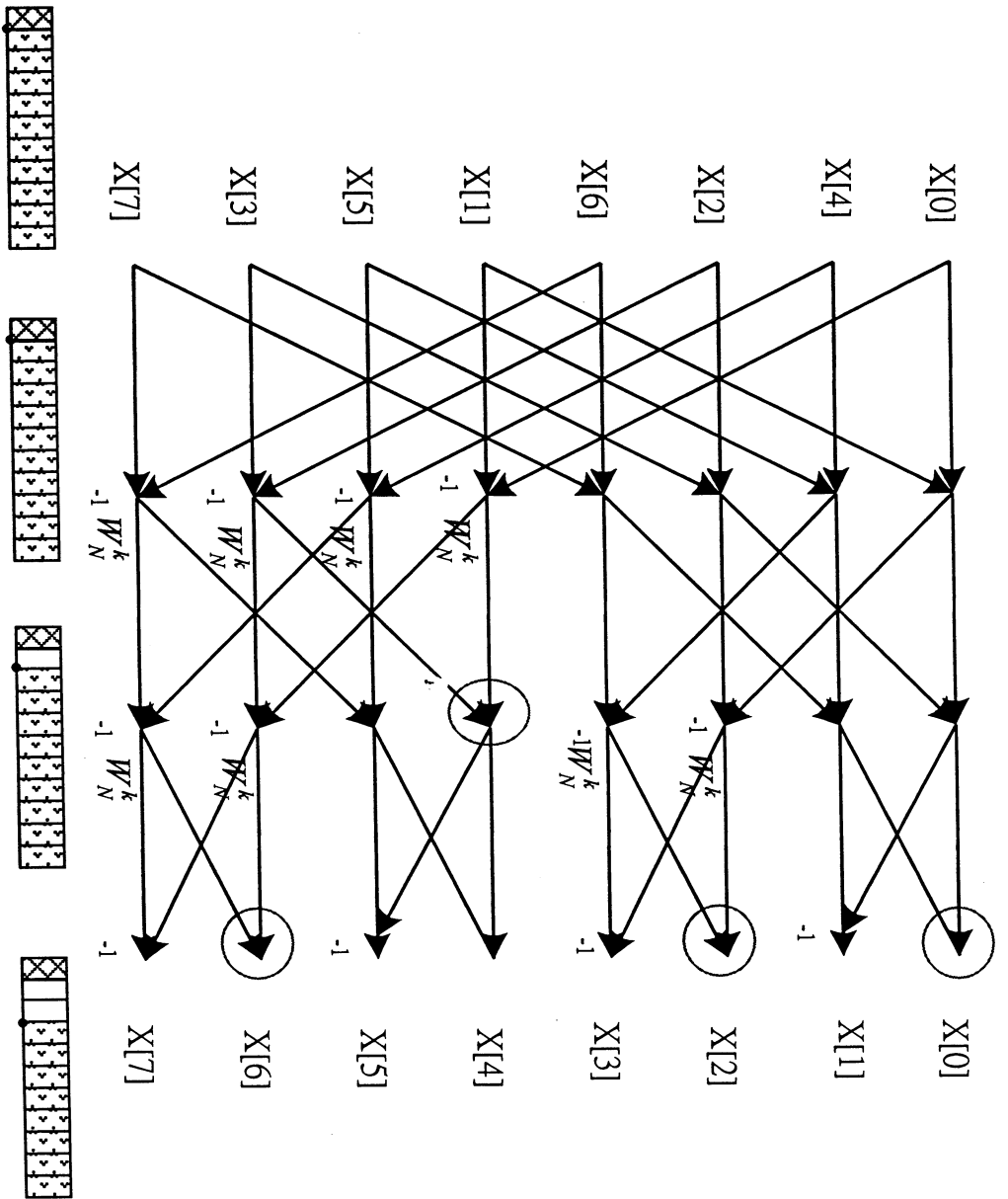
器，其中，每一該區塊之大小係數係儲存在一記憶表中。

16. 如申請專利範圍第11項所述之快速傅立葉轉換處理器，其中，在該等乘法器與該蝴蝶運算器之間更設有至少一緩衝器。

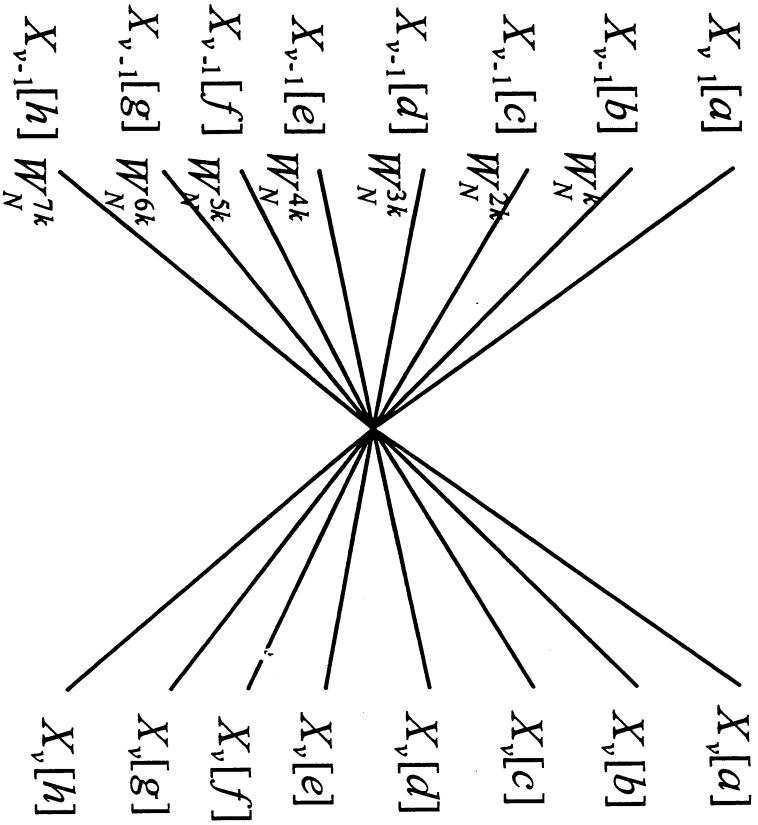
17. 如申請專利範圍第11項所述之快速傅立葉轉換處理器，其中，更包括一共用匯流排，提供連接該陣列預取緩衝器、該蝴蝶運算器及該正規化單元。

18. 如申請專利範圍第11項所述之快速傅立葉轉換處理器，更包括一複數指數儲存單元。

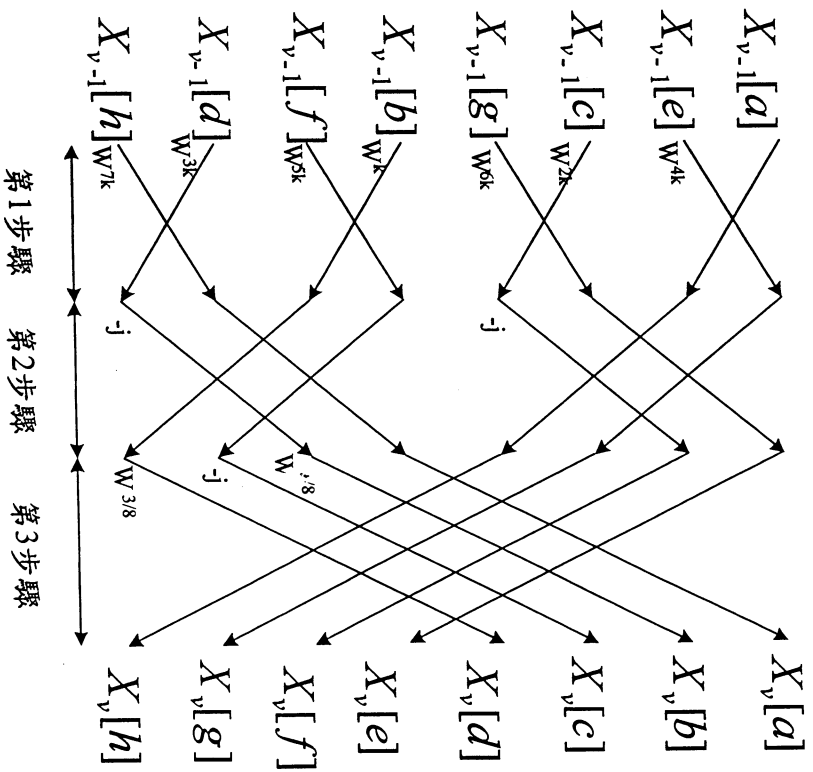




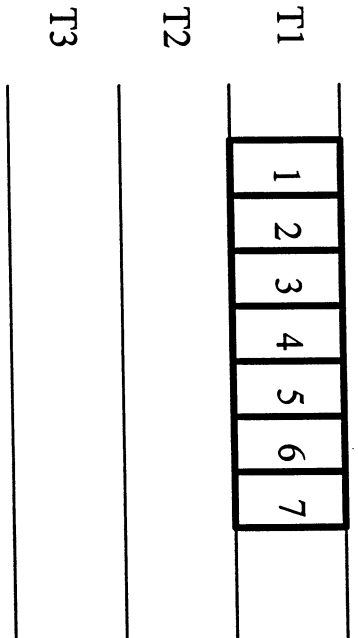
第一圖



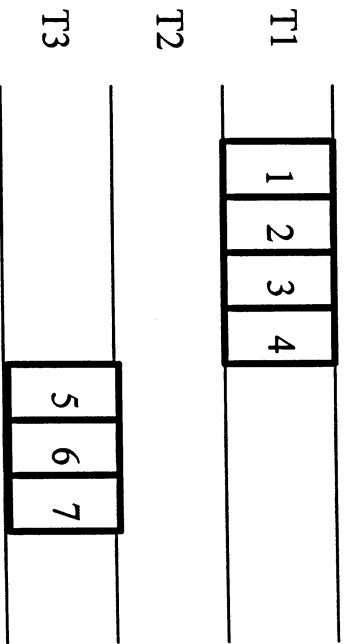
第二(a)圖



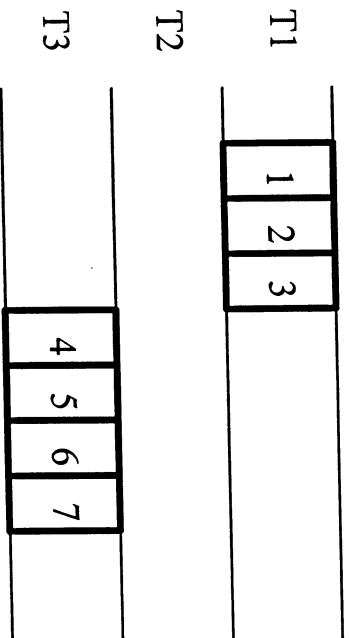
第二(b)圖



第三(a)圖

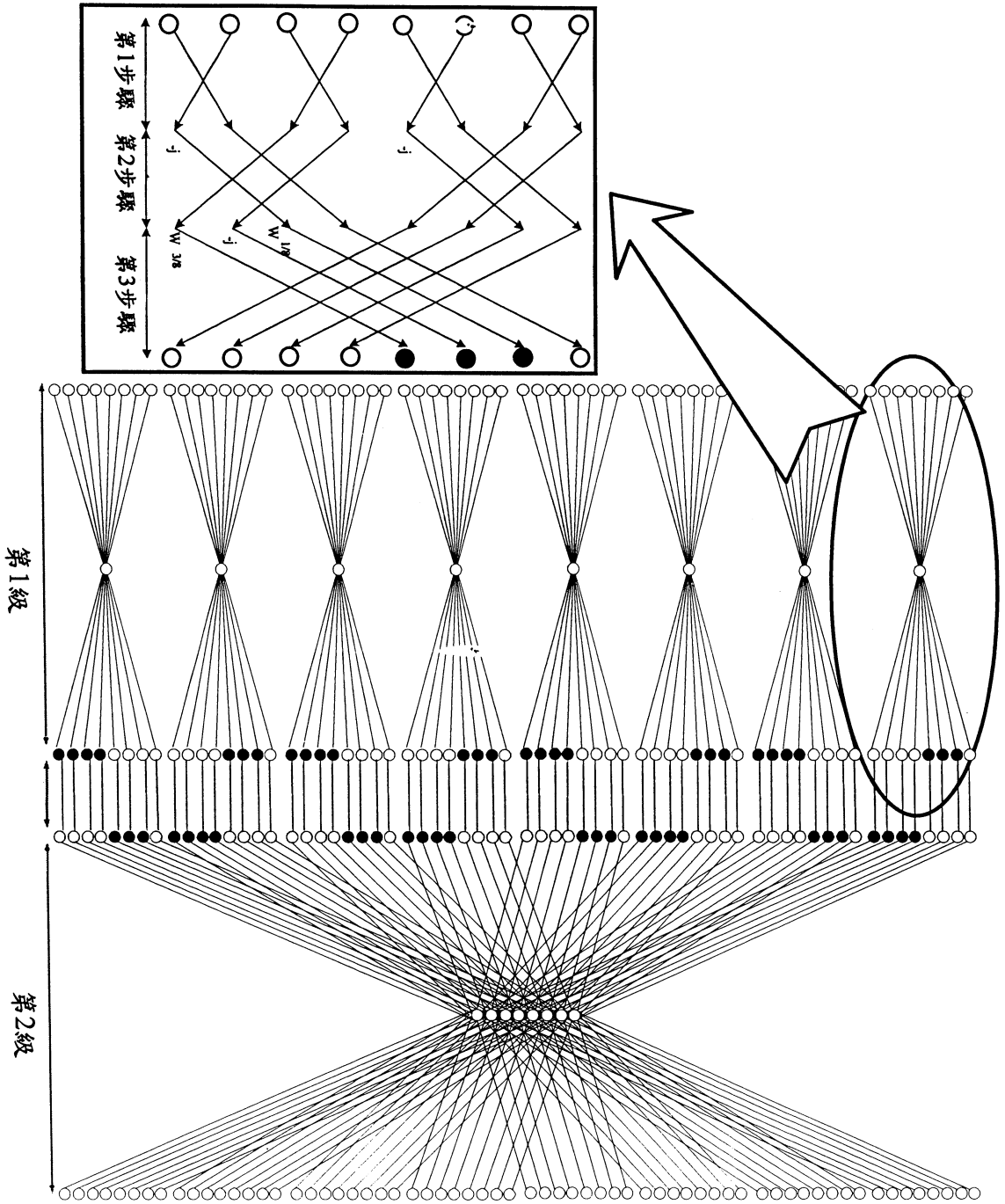


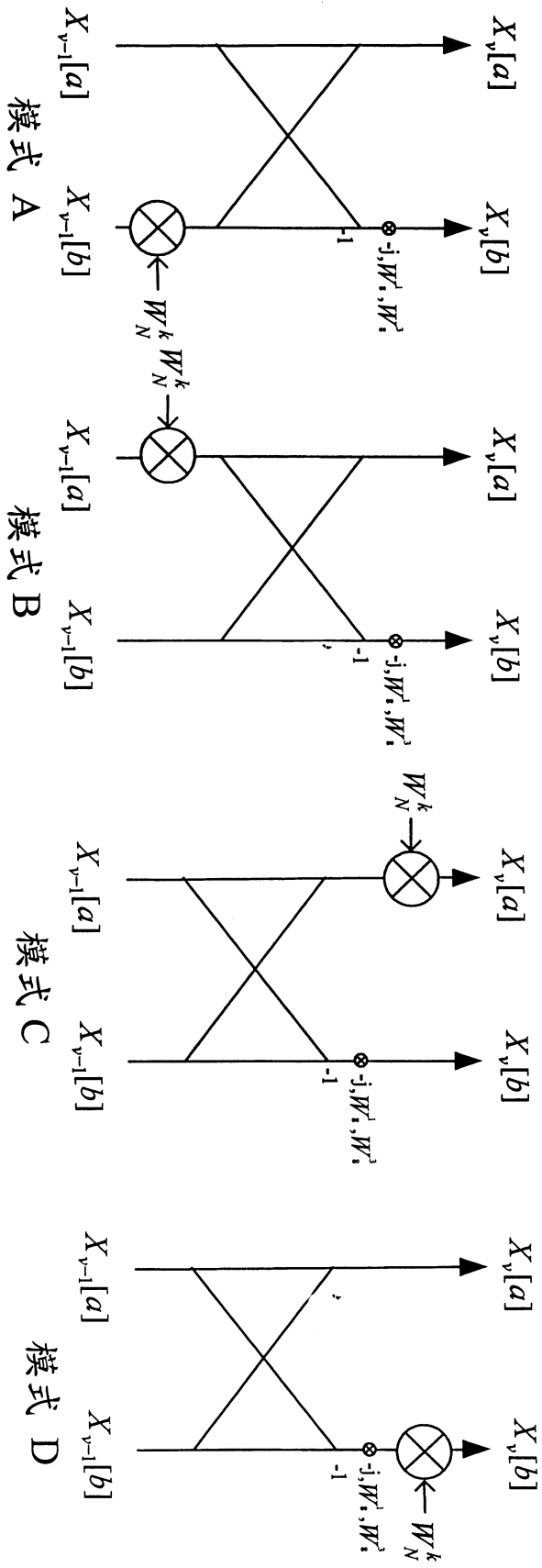
第三(b)圖



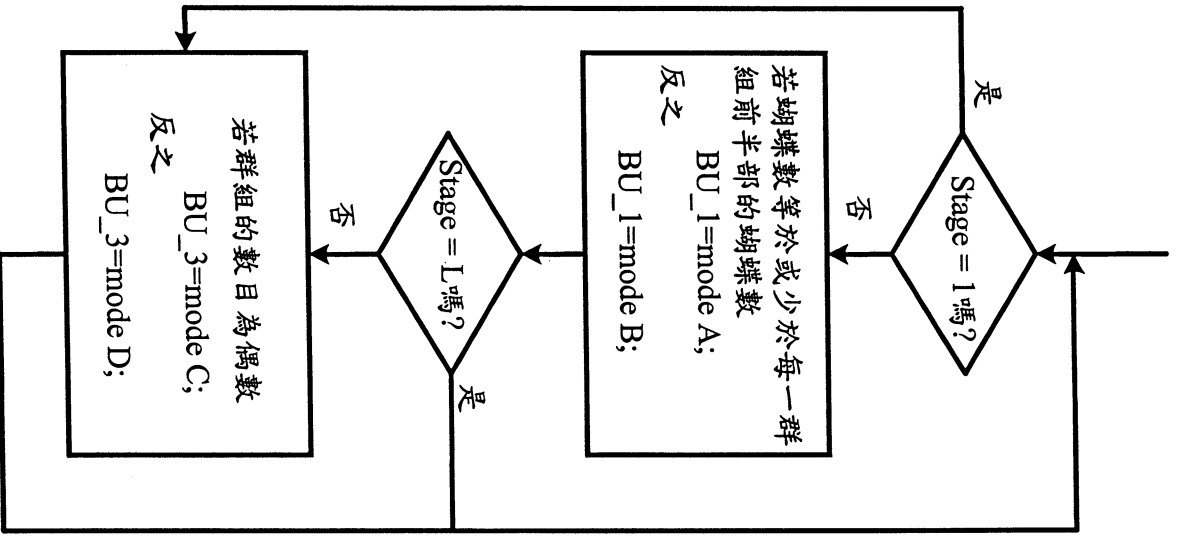
第三(c)圖

第四圖

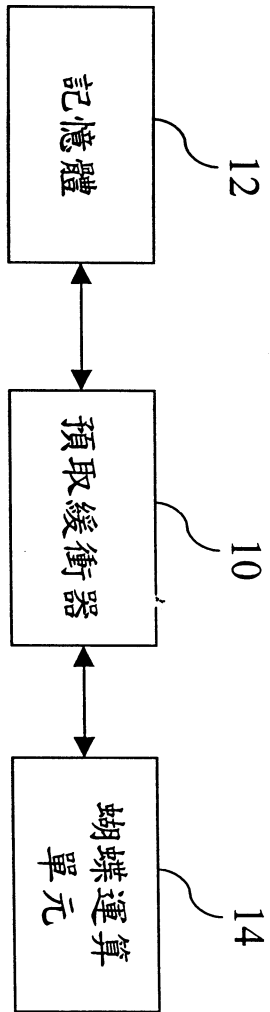




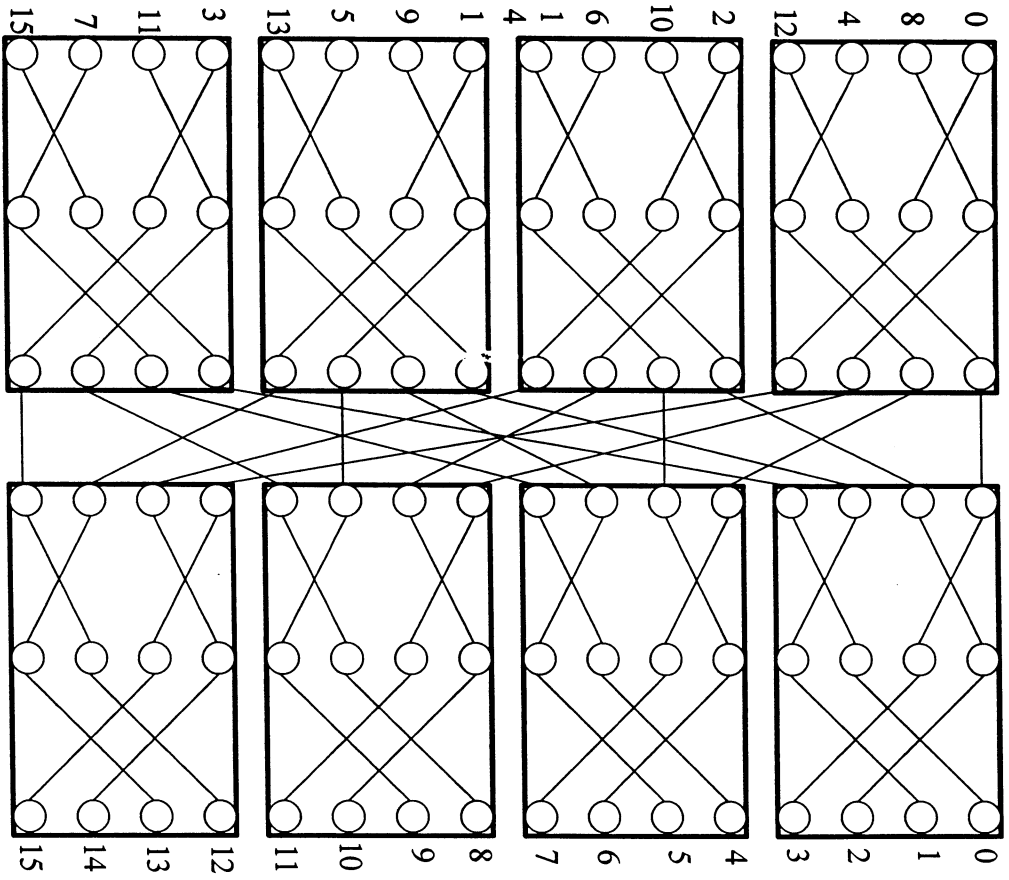
第五圖



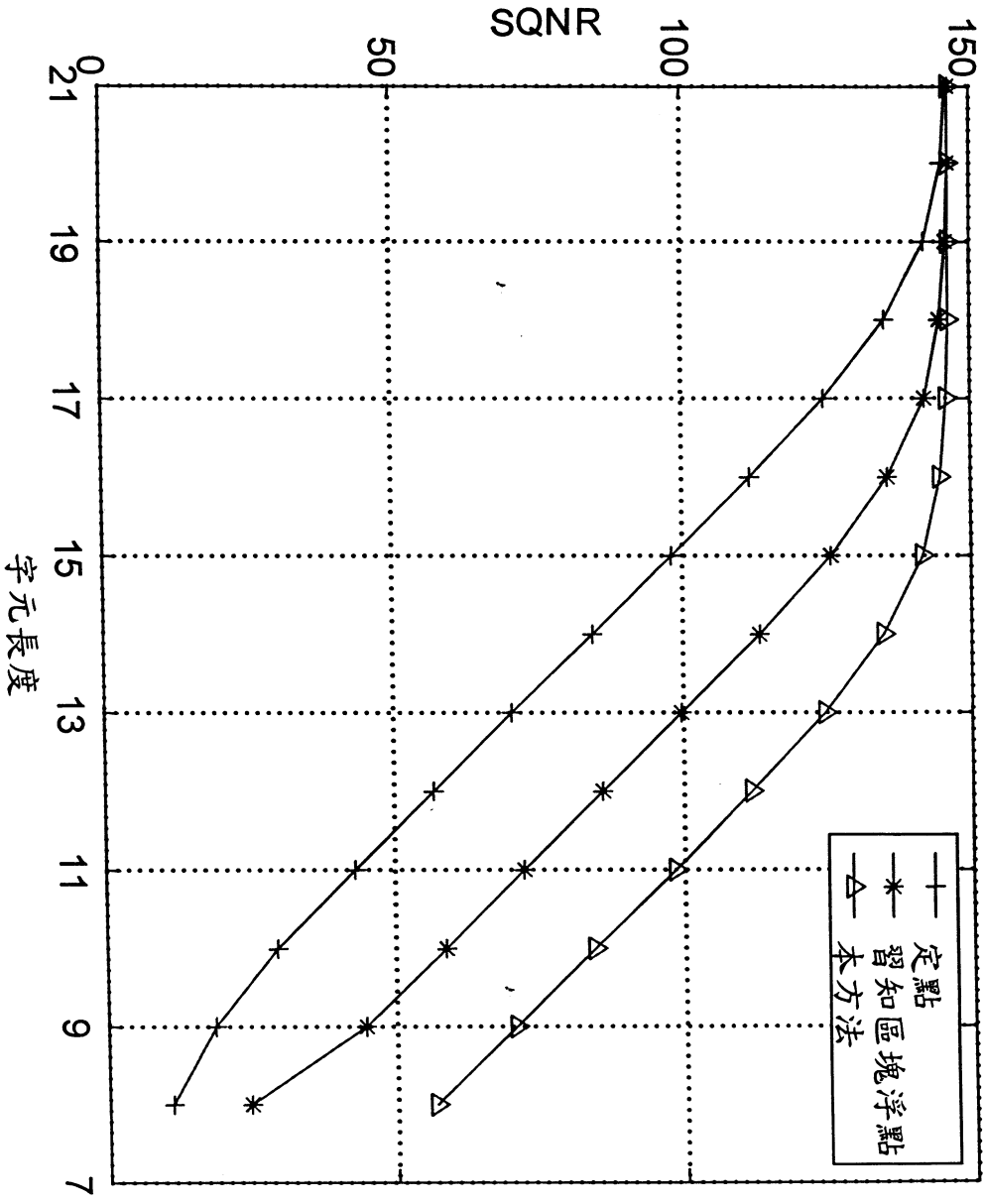
第六圖



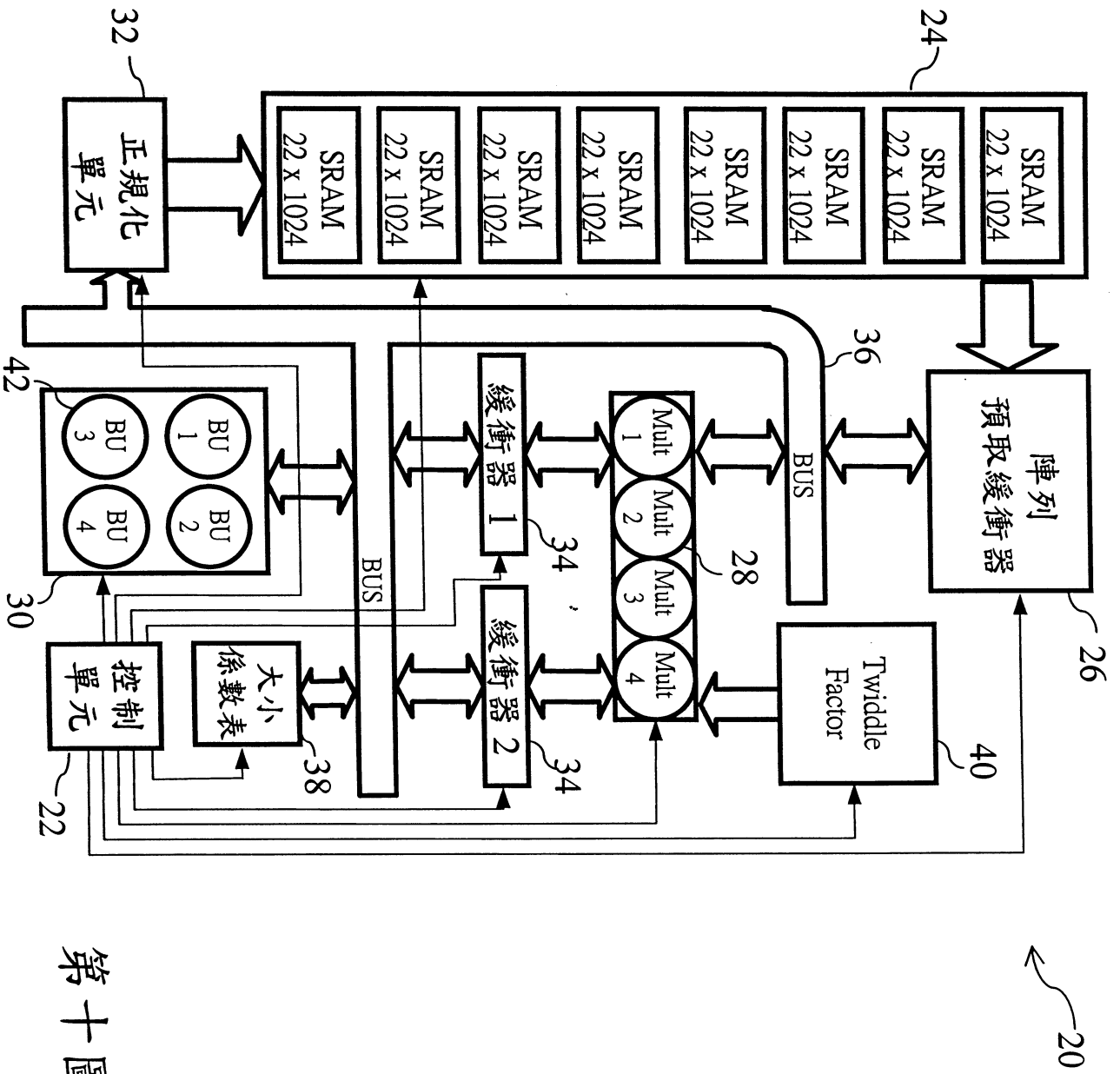
第七圖



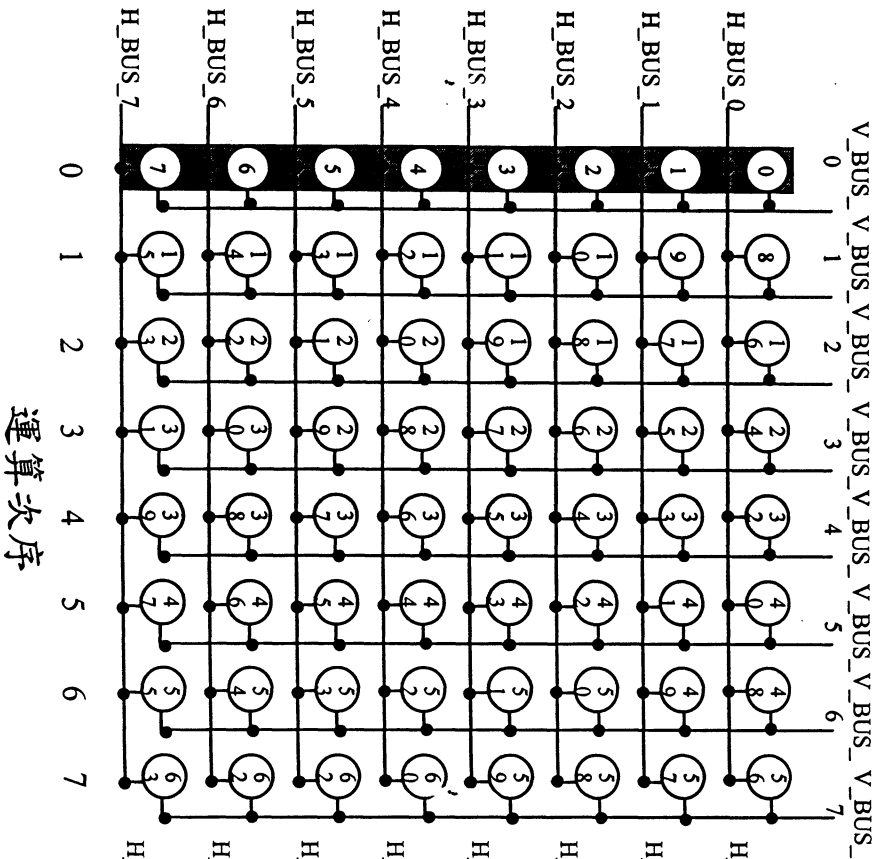
第八圖



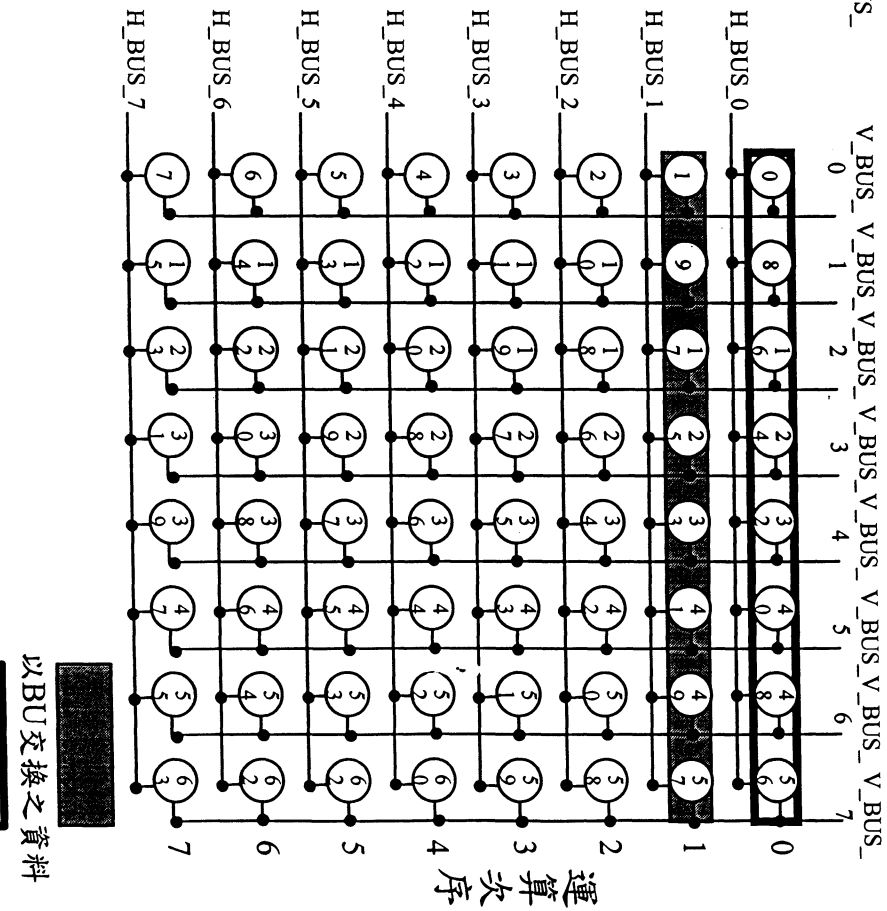
第九圖



第十圖

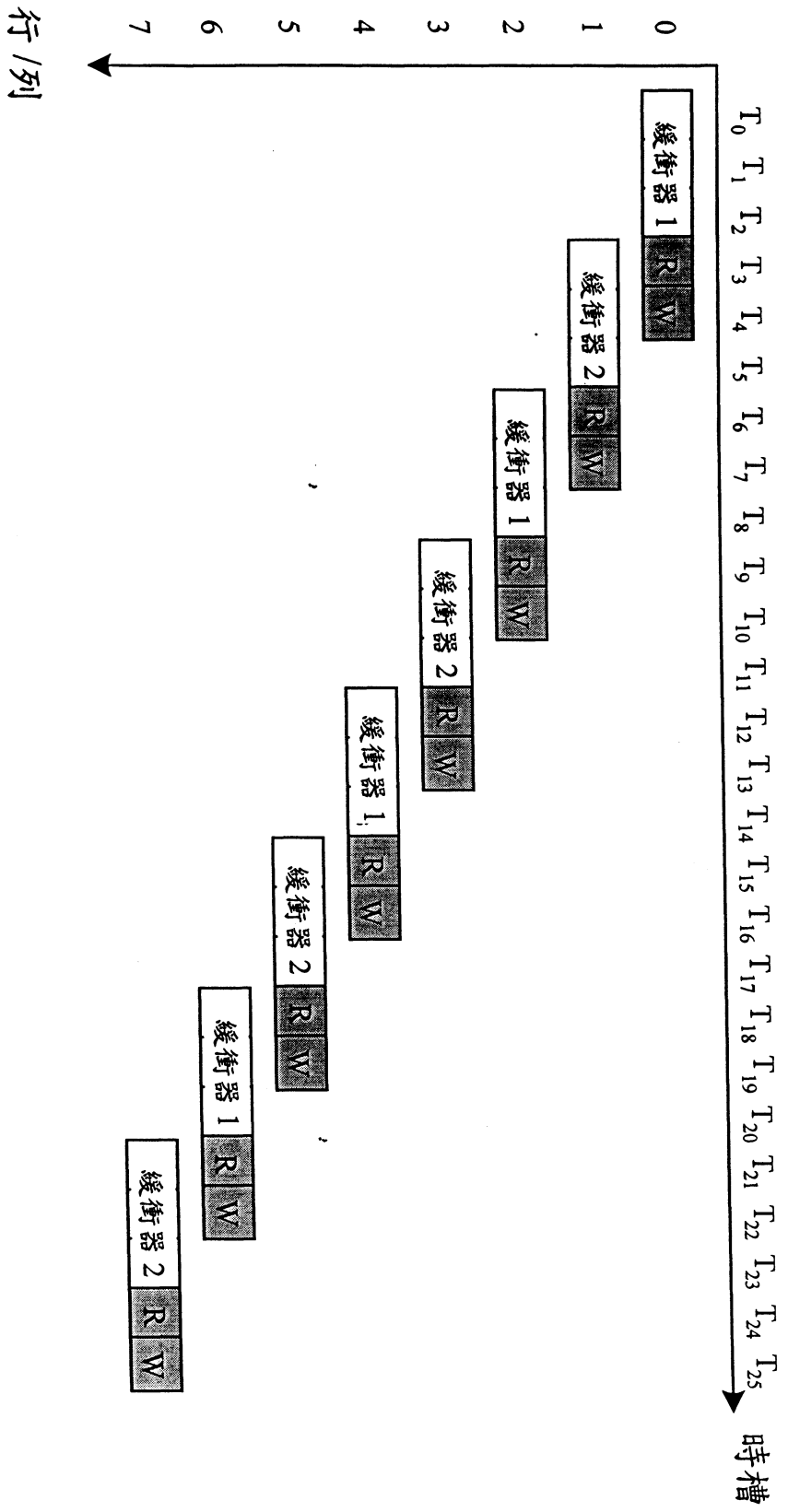


第十一(a)圖



第十一(b)圖

以BU交換之資料
 以記憶體交換之資料



第十二圖