公告本

I235582

發明專利說明書

（本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫）

※ 申請案號：92134464

※ 申請日期：92.12.5　　※IPC 分類：H04L 9/06　H04L 9/28

壹、發明名稱：（中文/英文）

可支援先進加密標準之加解密裝置

APPARATUS FOR SUPPORTING ADVANCED ENCRYPTION STANDARD ENCRYPTION AND DECRYPTION

貳、申 請 人：（共 1 人）

姓名或名稱：（中文/英文）

財團法人工業技術研究院

INDUSTRIAL TECHNOLOGY RESEARCH INSTITUTE

代表人：（中文/英文） 翁政義/ WENG, CHENG-I

住居所或營業所地址：（中文/英文）

新竹縣竹東鎮中興路四段 195 號

No. 195, Sec. 4, Chung Hsing Rd., Chutung, Hsinchu

國　籍：（中文/英文） 中華民國

參、發明人：（共 1 人）

姓　名：（中文/英文）

1. 呂誌忠 LU, CHIH-CHUNG

住居所地址：（中文/英文）

1. 台北縣樹林市東昇里中山路二段 48 號 2 樓

   2nd Floor, 48 Sec. 2 Chung-Sun Road, Dun-Sheng Li, Su-Lin City, Taipei County, Taiwan, R.O.C.

國　籍：（中文/英文）

1. 中華民國

肆、聲明事項：

☐ 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，其日期為： 年 月 日。

◎本案申請前已向下列國家（地區）申請專利 ☐ 主張國際優先權：

【格式請依：受理國家（地區）；申請日；申請案號數 順序註記】

1.

2.

3.

4.

5.

☐ 主張國內優先權(專利法第二十五條之一)：

【格式請依：申請日；申請案號數 順序註記】

1.

2.

☐ 主張專利法第二十六條微生物：

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

☐ 熟習該項技術者易於獲得，不須寄存。

玖、發明說明：

【發明所屬之技術領域】

本發明是有關於一種加解密裝置，且特別是有關於一種可支援先進加密標準(AES)之加解密裝置。

【先前技術】

由於近年來電子商務與線上交易發展得相當蓬勃，因此對於資料加密的要求也益加嚴謹，繼資料加密標準(Data Encryption Standard, DES)後更發展出先進加密標準(Advanced Encryption Standard, AES)的密碼系統，使資料的保密性更上層樓。另一方面，AES 密碼系統屬於對稱式加密系統，也就是說，加解密時所使用的是同一把金鑰，金鑰的長度可以是 128 位元、192 位元或 256 位元等，而明文(Plaintext)及密文(Ciphertext)則只可以是 128 位元，為方便說明起見，下文之明文、密文及金鑰等均以 128 位元的長度為例並說明之。

AES 密碼系統進行加密運算時的演算法：

```
AddRoundKey
    for round=1 to Nr-1
        KeyExpansion
        SubBytes
        ShiftRows
        MixColumes
        AddRoundKey
    end for
    SubBytes
    ShiftRows
```

AddRoundKey

首先會執行「加入循環金鑰」(以下簡稱 AddRoundKey)的步驟，此時系統會將明文與第一把次金鑰(次金鑰為金鑰經過特定運算產生，以下簡稱 SubKey)進行互斥或(XOR)運算後輸出，並進入以下的迴圈；為方便說明起見，暫且將明文與第一把 SubKey 進行互斥或運算後的資料稱為加密輸入資料。迴圈的數目設定為 Nr-1，Nr 的大小則依照 AES 的規定設定之。迴圈內的「金鑰更迭」(以下簡稱 KeyExpansion)運算會根據前一把 SubKey 來產生出後一把 SubKey，也就是在第一次執行迴圈時會根據第一把 SubKey 產生出第二把 SubKey，在第二次執行迴圈時會根據第二把 SubKey 產生出第三把 SubKey，依此類推。接著將加密輸入資料進行「次位元運算」(以下簡稱 SubBytes)、「列移轉運算」(以下簡稱 ShiftRows)及「行混合運算」(以下簡稱 MixColumes)後再與目前的 SubKey(因已執行 KeyExpansion，故此時為第二把 SubKey)進行互斥或運算，再次執行上述步驟直到迴圈結束。迴圈結束後，系統會將此時的資料進行 SubBytes、ShiftRows 及 AddRoundKey 等運算後，完成加密步驟。下文將繼續說明解密運算時的演算法。

AES 密碼系統進行解密運算時的演算法：

```
AddRoundKey
for round=1 to Nr-1
  InvKeyExpansion
  InvShiftRows
  InvSubBytes
  AddRoundKey
  InvMixColumes
end for
```

InvShiftRows

InvSubBytes

AddRoundKey

基本上，解密的運算程序為加密時的逆運算。首先系統會執行 InvAddRoundKey 的步驟，將密文與最後一把，例如是第十把 SubKey 進行位元互斥或(XOR)運算後輸出，並進入以下的迴圈；為方便說明起見，暫且將密文與第十把 SubKey 進行互斥或運算後的資料稱為解密輸入資料。需要注意的是，由於互斥或運算的特性，使得 InvAddRoundKey 的運算與 AddRoundKey 相同，因此在下文的說明中此二者將不再加以區隔而統一以 AddRoundKey 稱之。執行迴圈時，迴圈內的「逆金鑰更迭」(以下簡稱 InvKeyExpansion)運算會根據後一把 SubKey 來產生出前一把 SubKey，也就是在第一次執行迴圈時會根據第十把 SubKey 產生出第九把 SubKey，在第二次執行迴圈時會根據第九把 SubKey 產生出第八把 SubKey，依此類推。接著將解密輸入資料進行「逆次位元運算」(以下簡稱 InvSubBytes)、「逆列移轉運算」(以下簡稱 InvShiftRows)及「逆行混合運算」(以下簡稱 InvMixColumes)後再與目前的 SubKey(因已執行 InvKeyExpansion，故此時為第九把 SubKey)進行互斥或運算，再次執行上述步驟直到迴圈結束。迴圈結束後，系統會將此時的資料進行 InvSubBytes、InvShiftRows 及 AddRoundKey 等運算後，完成解密步驟。

如上文所述，加解密的過程中有五個主要的資料處理步驟，分別為 AddRoundKey, KeyExpansion, SubBytes, ShiftRows, MixColumes，以下將針對這些處理程序一一加以說明；需要注意的是，由於上述資料處理步驟環環相扣，前一步驟的輸出資料(out)即為下一步驟的輸入資料(in)，為簡化圖式及便於說明起

見，將不再另行定義輸入資料及輸出資料的名稱；此外，下文中所有的加號(+)係表示互斥或運算而並非單純的加法運算，爾後不再重複說明。

明文、密文、次金鑰的資料型態為 4×4 的矩陣，矩陣中每個元素(Element)為 8 位元，故此三者均為 128 位元的資料長度。請參照第 1 圖，其繪示了 AddRoundKey 的資料處理情形，很明顯的，這個步驟是將輸入資料(in)與次金鑰(k)進行互斥或運算後求得輸出資料(out)，而後續的運算可繼續針對輸出資料加以處理。由互斥或基本的運算原理可知，將輸出資料與次金鑰進行互斥或運算後即可反推回輸入資料。接著請參照第 2 圖，其繪示 ShiftRows 的資料處理情形。這個步驟是將輸入資料(例如是經 AddRoundKey 運算後的輸出資料)中的每一列規則性地移轉數個位元，例如將第一列向右移轉 0B(Byte)、將第二列向右移轉 1B、將第三列向右移轉 2B、將第四列向右移轉 3B，然後將結果輸出。若加密時如此處理，那麼解密時所進行的 InvShiftRows 就必須反其道而行，也就是將第一列向左移轉 0B、將第二列向左移轉 1B、將第三列向左移轉 2B、將第四列向左移轉 3B。

請參照第 3 圖，其繪示 MixColumns/InvMixColumns 的資料處理情形。MixColumns 的執行方式是將輸入資料(例如是經 ShiftRows 運算後的輸出資料)中的每一行做矩陣相乘的運算後求得輸出資料；反之，若將輸出資料執行反矩陣運算即可反推回輸入資料，亦即執行 InvMixColumns 與 MixColumns 所採用的矩陣互為反矩陣。

請參照第 4 圖，其繪示 SubBytes/InvSubBytes 的資料處理情形。SubBytes 中的主要運算單元稱為 S-Box,每個 S-Box 之輸入資料(圖式中標示為 in, 即第 1 式中的 x)與輸出資料(圖式中

標示為 out, 即第 1 式中的 y)皆為 1byte (8bits)之資料，則

$$y = M * \text{multiplicative\_inverse}(x) + c. \qquad (1)$$

$$M = \begin{pmatrix} 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \end{pmatrix}, \quad \text{常數 } c=[0\ 1\ 1\ 0\ 0\ 0\ 1\ 1]^T.$$

其中乘法反向(multiplicative_inverse)為一很複雜的函數(function)，因此大部份的作法係直接利用查表法求解，所以會有 y=Table_A(x)的查表資料，Table_A 即圖式中的次位元表，也就是 AES 標準內的 S-BOX 表。同樣的，要求得 inverse S-Box 來完成 InvSubBytes function，所以會需要另一個表 x=Table_B(y)，Table_B 即圖式中的逆次位元表，也就是 AES 標準內的 inv-S-BOX 表。但這兩個表勢必會佔去很大的硬體空間，在使用上相當不經濟。

如上文所述，傳統加解密處理步驟除了執行流程不同外，Inverse function 也是個問題，尤其是其中的 SubBytes 跟 InvSubBytes 為兩個查表的 function，在高效率的設計要求下會佔用很大的記憶體空間(2＊16＊256＊8bit)；此外，MixColumns 跟 InvMixColumns 兩個 function 內容為矩陣的乘法，若不能有效地加以整合勢必也會耗費相當的運算資源，是個須要加以考量並重新設計的模組。

【發明內容】

有鑑於此，本發明之目的旨在提供一種可支援先進加密標準之加解密裝置的整合型電路模組，用以選擇性地執行次位元運算

及逆次位元運算(SubBytes/InvSubBytes operaton)，在進行運算時利用共用的對應表資料以節省運算資源以外，藉由簡化的電路結構，能達成兼具整體關鍵路徑(critical paths)減化及低複雜度的需求，使得此運算模組之速度能有所提高。

本發明之另一目的旨在提供一種可支援先進加密標準之循環運算模組(round module)，整合次位元運算及逆次位元運算、列移轉(ShiftRows)/逆列移轉(InvShiftRows)運算，及行混合/逆行混合運算等於同一模組中，用以選擇性地執行加密及解密的循環運算(a round)；藉由使用此循環運算模組，先進加密標準之加解密裝置之硬體實作能符合高運算速度及兼具低複雜度的需求。

根據上述之發明目的，本發明提出一種可支援先進加密標準(Advanced Encryption Standard, AES)之整合型次位元(SubBytes) /逆次位元(InvSubBytes)運算裝置，用以針對一輸入資料碼，選擇性地進行次位元和逆次位元運算後輸出一欲求之輸出資料碼，此整合型次位元/逆次位元運算裝置包括：

一第一矩陣運算器，用以針對此輸入資料碼進行一第一矩陣運算，並輸出此第一矩陣運算之結果；一第一互斥或(exclusive-OR)運算模組，用以針對此輸入資料碼進行一第一互斥或運算，並輸出此第一互斥或運算之結果；　一第一多工器，與此第一矩陣運算器及此第一互斥或運算模組耦接，此第一多工器係依據一選擇信號，自此第一矩陣運算之結果及此第一互斥或運算之結果二者間擇一輸出，以作為此第一多工器之輸出資料碼；一查表運算裝置，與此第一多工器耦接，用以依據此第一多工器之輸出資料碼，進行一查表運算後輸出一查表資料碼；一第二矩陣運算器，用以針對此查表資料碼進行一第二矩陣運算，並

輸出此第二矩陣運算之結果；一第二互斥或運算模組，用以針對此查表資料碼進行一第二互斥或運算，並輸出此第二互斥或運算之結果；以及一第二多工器，與此第二矩陣運算模組及此第二互斥或運算模組耦接，此第二多工器係依據此選擇信號自此第二矩陣運算之結果及此第二互斥或運算之結果二者間擇一輸出，以作為此第二多工器之輸出資料碼；其中，此第二多工器之輸出資料碼即為此欲求之輸出資料碼。

此整合型次位元/逆次位元運算裝置，係於此選擇信號代表需要進行加密時，進行次位元運算，其中此第一多工器選擇此第一互斥或運算之結果，此第二多工器選擇此第二互斥或運算之結果。當此選擇信號代表需要進行解密時，此整合型次位元/逆次位元運算裝置進行逆次位元運算，其中此第一多工器選擇此第一矩陣運算之結果，此第二多工器選擇此第二矩陣運算之結果。

根據上述之另一發明目的，本發明提出一種可支援先進加密標準之循環運算模組(round module)，用以依據一輸入資料碼及一次金鑰選擇性地進行加密/解密運算後產生一輸出資料碼，此循環運算模組包括：

一互斥或閘，用以將此輸入資料碼與此次金鑰進行互斥或運算後產生此互斥或閘之輸出碼；一第一多工器，與此互斥或閘耦接，此第一多工器具有一第一輸入端及一第二輸入端，此第一輸入端係用以接收一待解密資料碼且此第二輸入端用以接收此互斥或閘之輸出碼，其中，此第一多工器係依據一選擇信號自此待解密資料碼與此互斥或閘之輸出碼二者間擇一輸出此第一多工器之輸出碼；一次位元/逆次位元運算裝置，耦接至此第一多工器，用以將此第一多工器之輸出碼進行次位元/逆次位元運算後輸出一替代運算輸出碼；一列移轉(ShiftRows)/逆列移轉

(InvShiftRows)運算裝置，耦接至此次位元/逆次位元運算裝置，用以將此替代運算輸出碼進行列移轉/逆列移轉運算後輸出一移轉運算輸出碼；一第二多工器，與此互斥或閘及此列移轉/逆列移轉運算裝置耦接，此第二多工器具有一第一輸入端及一第二輸入端，此第一輸入端係用以接收此互斥或閘之輸出碼且此第二輸入端用以接收此移轉運算輸出碼，其中，此第二多工器係依據此選擇信號自此互斥或閘之輸出碼與此移轉運算輸出碼二者間擇一輸出此第二多工器之輸出碼；一行混合/逆行混合運算裝置，與此第二多工器耦接，用以將此第二多工器之輸出碼進行行混合/逆行混合運算後輸出一混合運算輸出碼；一第三多工器，與此第二多工器及此行混合/逆行混合運算裝置耦接，此第三多工器具有一第一輸入端及一第二輸入端，此第一輸入端係用以接收此第二多工器之輸出碼且此第二輸入端用以接收此混合運算輸出碼，其中，此第三多工器係依據一加解密盡判斷信號自此第二多工器之輸出碼與此混合運算輸出碼二者間擇一輸出，且此第三多工器之輸出碼即為此待解密資料碼；一第四多工器，與此第三多工器及此列移轉/逆列移轉運算裝置耦接，此第四多工器具有一第一輸入端及一第二輸入端，此第一輸入端係用以接收此移轉運算輸出碼且此第二輸入端用以接收此待解密資料碼，其中，此第四多工器係依據此選擇信號自此移轉運算輸出碼與此待解密資料碼二者間擇一輸出此第四多工器之輸出碼；以及一第五多工器，與此第四多工器及此互斥或閘耦接，此第五多工器具有一第一輸入端及一第二輸入端，此第一輸入端係用以接收此第四多工器之輸出碼且此第二輸入端用以接收此互斥或閘之輸出碼，其中，此第五多工器係依據一回合盡判斷信號自此第四多工器之輸出碼與此互斥或閘之輸出碼二者間擇一輸出，以作為此第五多工

器之輸出碼；其中，此第五多工器之輸出碼即為此輸出資料碼。

此次位元/逆次位元運算裝置包括：一第一矩陣運算器，用以針對此第一多工器之輸出碼進行一第一矩陣運算，並輸出此第一矩陣運算之結果；一第一互斥或(exclusive-OR)運算模組，用以針對此第一多工器之輸出碼進行一第一互斥或運算，並輸出此第一互斥或運算之結果；一第一選擇器，與此第一矩陣運算器及此第一互斥或運算模組耦接，此第一選擇器係依據此選擇信號，自此第一矩陣運算之結果及此第一互斥或運算之結果二者間擇一輸出，以作為此第一選擇器之輸出碼；一查表運算裝置，耦接至此第一選擇器，用以依據此第一選擇器之輸出碼，進行一查表運算後輸出一查表資料碼；一第二矩陣運算器，用以針對此查表資料碼進行一第二矩陣運算，並輸出此第二矩陣運算之結果；一第二互斥或運算模組，用以針對此查表資料碼進行一第二互斥或運算，並輸出此第二互斥或運算之結果；以及一第二選擇器，與此第二矩陣運算器及此第二互斥或運算模組耦接，此第二選擇器係依據此選擇信號自此第二矩陣運算之結果及此第二互斥或運算之結果二者間擇一輸出，以作為此替代運算輸出碼。

又根據另一發明目的，本發明提出一種先進加密標準之加解密裝置，用以選擇性的針對一輸入資料碼進行先進加密標準之加密或解密的動作，以輸出一輸出資料碼，此加解密裝置包括：

一循環運算裝置，與此金鑰儲存裝置耦接，用以撰擇性地進行加密及解密之一在進行中所需之循環運算，依據輸入此循環運算裝置之一輸入碼及一次金鑰以輸出一循環運算之輸出碼；一次金鑰更迭運算裝置，與此循環運算裝置耦接，用以撰擇性地產生進行加密及解密之一時，循環運算所需之此次金鑰，其中此次金鑰係基於輸此次金鑰更迭運算裝置之一已知次金鑰而得之一待

解次金鑰；以及一金鑰儲存裝置，與此循環運算裝置及此次金鑰更迭運算裝置耦接，用以作次金鑰的暫存及分配，以便此次金鑰更迭運算裝置及此循環運算裝置進行循環運算。另外，此循環運算裝置包括一次位元/逆次位元運算裝置，其結構如上所述。

此外，此金鑰儲存裝置接收此循環運算輸出碼及此次金鑰更迭運算裝置輸出之此次金鑰；此次金鑰更迭運算裝置之此已知次金鑰及此循環運算裝置之此輸入碼係為此金鑰儲存裝置所輸出。此金鑰儲存裝置暫存此輸入資料碼，進行金鑰之分配及暫存，接收此循環運算裝置及此次金鑰更迭運算裝置之輸出以進行循環運算，並輸出此輸出資料碼。

為讓本發明之上述目的、特徵、和優點能更明顯易懂，下文特舉一較佳實施例，並配合所附圖式，作詳細說明如下。

【實施方式】

實施例一

實施例一主要是將次位元(SubBytes) 運算與逆次位元(InvSubBytes)運算加以整合，並利用適當的硬體予以實現。在說明之前，我們先將第(1)式再複習一次：

$$y = M * \text{multiplicative\_inverse}(x) + c. \qquad (1)$$

$$M = \begin{pmatrix} 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \end{pmatrix}, \quad \text{常數 } c=[0\ 1\ 1\ 0\ 0\ 0\ 1\ 1]^T.$$

若加密與解密分別採用兩個不同的表將佔去很大的硬體空間，因此接下來我們再繼續推導，由第(1)式可得：

$$x = multiplicative\_inverse^{-1}\ (M^{-1} * (y + c)). \quad (2)$$

因為 multiplicative_inverse()與 $multiplicative\_inverse^{-1}$()是相等的，所以第(2)式可改寫為：

$$x=multiplicative\_inverse(M^{-1} *(y + c)). \quad (3)$$

經由反矩陣運算的推導後我們得到：

$$M' = M^{-1} = \begin{pmatrix} 01010010 \\ 00101001 \\ 10010100 \\ 01001010 \\ 00100101 \\ 10010010 \\ 01001001 \\ 10100100 \end{pmatrix} \quad (4)$$

所以第(3)式可記為：

$$x=multiplicative\_inverse(M'*(y+c)). \quad (5)$$

由第(1)式與第(5)式我們發現兩個 equation 可共用同樣的對應表(即 multiplicative_inverse())，所以我們可將 S-box 與 inverse S-box 加以整合，以降低硬體需求。

請參照第 5A 圖，其繪示採用同一對應表以實現一種可支援先進加密標準(Advanced Encryption Standard, AES)之整合型次位元/逆次位元運算裝置方塊圖。如圖所示，整合型次位元/逆次位元運算裝置 500A 包括矩陣運算器 510、多工器 520、乘法反向運算裝置 530、矩陣運算器 540 及多工器 550，其中乘法反向運算裝置 530 可執行 data=multiplicative_inverse(addr)的運算，一般是利用查表的方式加以實現，即依據輸入資料查表後將查表資料輸出。矩陣運算器 510 負責執行 (in+c)*M'的運算，矩陣運算器 540 則負責執行 data*M+c 的運算，矩陣 M 及 M^{-1}的型態如上文所述。

當需要執行 SubBytes 運算時，選擇信號 ec 設定為一定值，

例如 1，以代表需要進行加密動作。此時輸入資料 in 經由多工器 520 饋入乘法反向運算裝置 530，經過查表後，反向運算裝置 530 將查表資料 data (即 multiplicative_inverse(in))輸出。而後矩陣運算器 540 再針對查表資料進行 out=data*M+c 的運算，其結果經由多工器 550 選擇並輸出；如此，SubBytes 運算即告完成。

另一方面，當需要執行 InvSubBytes 運算時，選擇信號 ec 設定另一定值，例如為 0，以代表需要進行解密動作。此時，輸入資料 in 饋入矩陣運算器 510 以進行(in+c)*M'的運算，運算結果再經由多工器 520 饋入乘法反向運算裝置 530 查表後產生查表資料，此結果最後經由多工器 550 輸出；如此，InvSubBytes 運算即告完成。

很明顯的，整合型次位元/逆次位元運算裝置 500A 同時具有 S-box 與 inverse S-box 的功能，但只需要一個表，在實現 Subbytes 與 InvSubBytes 兩個 function 上的硬體需求降到了原來的 57%，有著非常明顯的改良。

接下來，我們可以再把第 5A 圖右端的多工器 550 路徑做一些改進。第一步，先將前一級輸出到此多工器 550 之下方輸入之直接連線部分，可以插入一個運算模組

x = multiplicative_inverse(M"*(y+c")) (5.1)

其中，y 為 input，x 為 output，

$$M'' = \begin{pmatrix} 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \end{pmatrix} \qquad c'' = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

但不影響第 5A 圖之運算結果，接下來我們定義新的運算模組：

x = multiplicative_inverse(M(e)*(y+c(e))) (5.2)

其中

$$M(e)=\begin{pmatrix} 1eeee000 \\ 01eeee00 \\ 001eeee0 \\ 0001eeee \\ e0001eee \\ ee0001ee \\ eee0001e \\ eeee0001 \end{pmatrix} \qquad c(e)=\begin{pmatrix} 0 \\ e \\ e \\ 0 \\ 0 \\ 0 \\ e \\ e \end{pmatrix}$$

將第 5A 圖右端多工器 550 及矩陣運算器 540 部分用此運算模組取代，我們可得新的 Modified Inverse-optional S-box 模組。

上述針對第 5A 圖之電路結構之改進，是從減少元件的數量的角度來思考的，並沒有明顯的減少 critical path 及模組元件的複雜度。

接下來，我們跟據本發明的技術思想，進一步將整合型次位元/逆次位元運算裝置 500A 的設計作改良，以達到減少 critical path 及模組元件複雜度的結果，讓整體加解密的運算速度提高。

首先，我們把第 5A 圖左邊之+c 與$*M^{-1}$運算動作之順序對調，也就是

$$(in+c)*M^{-1}= in*M^{-1} + c*M^{-1} = in*M^{-1} + c'$$

其中 $c' = c*M^{-1}$；

接著，由於在 AES standard 中，+ 號代表位元之 XOR 操作，所以：

$$in = in + c' + c';$$

藉上述兩種方式，第 5A 圖之可以改變成第 5B 圖之整合型次位元/逆次位元運算裝置 500B 的架構，但其整體操作及結果不變；

再來，我們將第 5B 圖左邊多工器前相同之+c'運算移到多工器後面作，而第 5B 圖右下方之路徑，因為

$$data = data * M * M^{-1},$$

我們可以在第 5B 圖右下方之路徑插入*M 與 $*M^{-1}$ 運算動作，同樣不影響計算結果，所以可以得到如第 5C 圖所示之整合型次位元/逆次位元運算裝置 500C；

同理，第 5C 圖右邊多工器後面之兩個*M 運算移到多工器前與乘法反向運算裝置 530 之間來操作，可得如第 5D 圖之整合型次位元/逆次位元運算裝置 500D 電路；

最後，我們將第 5D 圖中間如虛線框內所示之三個運算操作之輸出入結果事先算好，並存於一新的對照表中，最後架構如第 5E 圖所示。

第 5E 圖是一種可支援先進加密標準之整合型次位元/逆次位元運算裝置 500E，用以針對一輸入資料碼 in，選擇性地進行次位元和逆次位元運算後輸出一欲求之輸出資料碼 out，此整合型次位元/逆次位元運算裝置 500E 包括：第一矩陣運算器 561、第一互斥或(exclusive-OR)運算模組 565、第一多工器 520、查表運算裝置 590、第二矩陣運算器 571、第二互斥或運算模組 575，以及第二多工器 550。

第一矩陣運算器 561，用以針對此輸入資料碼 in 進行第一矩陣運算，如本例中的$*M^{-1}$運算，並輸出此第一矩陣運算之結果。第一互斥或(exclusive-OR)運算模組 565，用以針對此輸入資料碼 in 進行第一互斥或運算，如本例中的+c'運算，並輸出此第一互斥或運算之結果。第一多工器 520，與此第一矩陣運算器 561 及此第一互斥或運算模組 565 耦接，此第一多工器 520 係依據選擇信號 ec，自此第一矩陣運算之結果及此第一互斥或運算之結果二者間擇一輸出，以作為此第一多工器之輸出資料碼。查表運算裝置 590，與此第一多工器 520 耦接，用以依據此第一多工器之

輸出資料碼，進行一查表運算後輸出一查表資料碼 data。第二矩陣運算器 571，用以針對此查表資料碼 data 進行第二矩陣運算，如本例中的 $*M^{-1}$ 運算，並輸出此第二矩陣運算之結果。第二互斥或運算模組 575，用以針對此查表資料碼進 data 行第二互斥或運算，如本例中的+c 運算，並輸出此第二互斥或運算之結果。第二多工器 530，與此第二矩陣運算模組 571 及此第二互斥或運算模組 575 耦接，此第二多工器 550 係依據此選擇信號 ec 自此第二矩陣運算之結果及此第二互斥或運算之結果二者間擇一輸出，以作為此第二多工器之輸出資料碼；其中，此第二多工器之輸出資料碼即為此欲求之輸出資料碼 out。

此整合型次位元/逆次位元運算裝置 500E，係於此選擇信號 ec 代表需要進行加密時，進行次位元運算，其中此第一多工器 520 選擇此第一互斥或運算之結果，此第二多工器 550 選擇此第二互斥或運算之結果。當此選擇信號 ec 代表需要進行解密時，此整合型次位元/逆次位元運算裝置 500E 進行逆次位元運算，其中此第一多工器 520 選擇此第一矩陣運算之結果，此第二多工器 550 選擇此第二矩陣運算之結果。

在本實施例一中：第一矩陣運算與第二矩陣運算實質上相同，即 $*M^{-1}$ 運算。而且，此第一互斥或運算包含一運算元，c'運算元，此運算元之值係基於此第一矩陣運算(如 M^{-1} 運算)及此第二互斥或運算(如+c 運算)所得者。再來，此查表運算模組 590 包含一對照表，此對照表係基於一乘法反向運算(multiplicative inverse operation)、此第一互斥或運算，及此第一矩陣運算所得者。在本例中，此對照表係基於+c'、Mulplicative_inverse()，及*M 運算而得者。

最後我們將第 5E 圖之架構與原來的第 5A 圖做比較，主要有

兩項改進：

(1)整體之關鍵路徑(critical paths)變短，加電路運算之速度可以提昇；

(2)*M^{-1}運算比*M運算，複雜度較低，因為矩陣M^{-1}內之元素1的個數只有矩陣M之元素1的3/5，所以複雜度也可降低；

綜合此二優點，我們可以得到比第5A圖之架構更快速更低複雜度的整合型次位元/逆次位元運算裝置500E。

實施例二

實施例二提供一種整合型加解密運算的演算法及利用循環運算在硬體上的實現，加解密運算流程為：

```
if(ec==0) for (i=0;i<Nr;i++)
      Inv_Opt_keyexpansion(key,1); //inverse key
for (i=0;i<=Nr;i++)
{     addroundkey;
      if (i==Nr) break;
      Inv_Opt_keyexpansion(key,ec);
      if (ec==1)
      {    Inv_Opt_subbytes(ec);
           Inv_Opt_shiftrows(ec);
           if (i<(Nr-1)) Inv_Opt_mixcolumns(ec);
      } else
      {    if (i>0) Inv_Opt_mixcolumns(ec);
           Inv_Opt_subbytes(ec);
           Inv_Opt_shiftrows(ec);
      }
```

}

其中，Nr 是循環運算的次數(number of rounds)，在進行 128 位元之 AES 加解密時，Nr 之值為 10；在進行 192 位元及 256 位元之 AES 加解密時，Nr 分別為 12 及 14。

接著請參照第 9 圖，其繪示依照本發明之實施例二所提供的一種可支援先進加密標準之循環運算裝置，用以支援上述先進加密標準之加解密演算流程。循環運算裝置 900 包括互斥或閘 90、次位元/逆次位元運算裝置 95、列移轉/逆列移轉運算裝置 97、行混合/逆行混合運算裝置 99 及數個多工器 910、920、930、940 及 950，其中次位元/逆次位元運算裝置 95 的實作方式係如第 5E 圖所示者。

當需要進行加密運算時，令選擇信號 ec 為 1，以改變循環運算裝置 900 的運作組態以進行加密動作。首先，輸入資料碼 in(此時為明文)與一 SubKey 饋入互斥或閘 90 執行 AddRoundKey 運算之後將結果輸出。接著經多工器 910，此 AddRoundKey 運算結果被饋入次位元/逆次位元運算裝置 95 以進行次位元運算。接著將次位元運算後的結果饋入列移轉/逆列移轉運算裝置 97 進行列移轉運算。之後，列移轉運算後的結果經多工器 920 饋入行混合/逆行混合運算裝置 99 以進行行混合運算。行混合運算前與運算後的資料分別饋入多工器 930 的輸入端(0)與輸入端(1)，多工器 930 則依據「加解密盡判斷信號」自兩輸入資料中擇一輸出。加解密盡判斷信號係對應到上述之加解密演算法之判斷方式，對於 128 位元之 AES 加解密來說，Nr 為 10，則此信號係可依據下列之判斷式或經由電路實作而產生：

~((ec&(i==4'd9))|(~ec&(i==4'd0)))

因此當加解密盡判斷信為 1 時多工器 930 會將行混合/逆行混合

運算裝置 99 的輸出資料輸出；當加解密盡判斷信為 0 時多工器 930 則會將多工器 920 的輸出資料輸出，為方便說明起見，茲將多工器 930 的輸出資料稱為待解密資料 93。如圖所示，待解密資料 93 係同時饋入多工器 910 之輸入端(0)及多工器 940 之輸入端(1)，此時多工器 940 可將其輸出至多工器 950 之輸入端(0)。多工器 950 之輸入端(1)係接收互斥或閘 90 之輸出資料，且多工器 950 係依據「回合盡判斷信號」自兩輸入資料中擇一輸出。回合盡判斷信號係由判斷循環運算之次數是否已到達 Nr 而產生，在此例中，可記為(i==4'd10)。因此當回合盡判斷信號為 0 時多工器 950 會將多工器 940 的輸出資料輸出，此多工器 940 的輸出資料會被饋入循環運算裝置 900 作為下一循環運算(next round)之輸入資料碼(in)。此外，經由 Inv_Opt_keyexpansion(key,ec)，執行 Key Expansion 的運算，以產生出下一 SubKey。依照上述演算法之迴圈設計，循環運算裝置 900 重複上述 AddRoundKey、SubBytes、ShiftRows、MixColumns 等步驟，進行一系列的循環運算，直到 i==4'd9 時，多工器 930 直接將多工器 920 的輸出資料經多工器 940、多工器 950 輸出，以作為下一個輸入資料碼 in；而後 i==4'd10，此輸入資料碼 in 與 SubKey 進行 AddRoundKey 運算後直接由多工器 950 輸出，加密程序即告結束，其中，此輸出之結果即為卻求之密文。

當需要進行解密運算時，令選擇信號 ec 為 0，以改變循環運算裝置 900 的運作組態以進行解密動作。首先，輸入資料 in(此時為密文)與 SubKey 饋入互斥或閘 90 執行 AddRoundKey 運算之後將結果輸出。接著經多工器 920，此 AddRoundKey 運算之結果被饋入行混合/逆行混合運算裝置 99 以進行逆行混合運算。逆行混合運算前與運算後的資料分別被饋入多工器 930 的輸入端(0)

與輸入端(1)，多工器 930 則依據加解密盡判斷信號自兩輸入資料中擇一輸出，其中加解密盡判斷信號的型態如上文所述。當加解密盡判斷信號為 1 時，多工器 930 會將行混合/逆行混合運算裝置 99 的輸出資料輸出；當加解密盡判斷信號為 0 時多工器 930 會將多工器 920 的輸出資料輸出。多工器 920 的輸出即待解密資料 93，待解密資料 93 可經多工器 910 饋入次位元/逆次位元運算裝置 95 以進行逆次位元運算，接著將逆次位元運算後的結果饋入列移轉/逆列移轉運算裝置 97 進行逆列移轉運算後經多工器 940 將結果輸出。另一方面，多工器 950 可依據回合盡判斷信號自兩輸入資料中擇一輸出，當回合盡判斷信號為 0 時多工器 950 會將多工器 940 的輸出資料輸出。此多工器 940 的輸出資料會被饋入循環運算裝置 900 作為下一循環運算(next round)之輸入資料碼(in)。此外，經由 Inv_Opt_keyexpansion(key,ec)，執行 Key Expansion 的運算，以產生出下一 SubKey。並依照上述演算法之迴圈設計，循環運算裝置 900 重複上述 AddRoundKey、InvMixColumns、InvSubBytes 及 InvShiftRows 等步驟，進行一系列的循環運算，直到 i==4'd9 時，多工器 930 便將多工器 920 的輸出資料經多工器 910 饋入次位元/逆次位元運算裝置 95，其結果之後饋入列移轉/逆列移轉運算裝置 97 並加以運算後，透過多工器 940 及多工器 950 加以輸出，以作為下一個輸入資料碼 in；而後 i==4'd10，此輸入資料 in 與 SubKey 進行 AddRoundKey 運算後直接由多工器 950 輸出，解密程序即告結束，其中，此輸出之結果即為卻求之明文。

實施例三

基於上述的循環運算裝置，我們依據本發明，提出 AES 之加

解密裝置的架構，用以選擇性的進行AES加密或解密的動作。請參考第10圖，此AES加解密裝置1000，包括次金鑰更迭運算裝置800、循環運算裝置900及金鑰儲存裝置1100。金鑰儲存裝置1100包含三個記憶裝置1110、1120及1130，例如是暫存器(registers)，它們分別用以存放Data、Key，及backup key，整體架構如第10圖所示。當中，din是代表輸入的資料碼，dout是代表輸出的資料碼。

金鑰儲存裝置1100與循環運算裝置900及次金鑰更迭運算裝置800耦接，用以作次金鑰的暫存及分配，以便次金鑰更迭運算裝置800及循環運算裝置900進行循環運算。金鑰儲存裝置1100提供循環運算裝置900輸入資料碼in，接收及暫存循環運算裝置900所輸出的輸出資料碼out於記憶裝置1110；金鑰儲存裝置1100亦用於提供次金鑰更迭運算裝置800輸入資料碼in，接收及次金鑰更迭運算裝置800所輸出的輸出資料碼out於記憶裝置1120中，其中此次金鑰更迭運算裝置800所輸出的輸出資料碼out，即subkey，會饋入到循環運算裝置900之key端，以作為循環運算裝置900之次金鑰之用。

當需要進行加密運算時，令選擇信號ec為1，以改變AES加解密裝置1000之運作組態以進行加密動作。此外，由上述的實施例一及實施例二可知，循環運算裝置900及循環運算裝置900之次位元/逆次位元運算裝置95亦因此改變其運作組態以進行加密動作。循環運算裝置900之count端，即用以輸入目前循環運算的次數。而din代表明文，而dout是經過AES加解密裝置1000加密後的密文。

當需要進行解密運算時，令選擇信號ec為0，以改變AES加解密裝置1000之運作組態以進行解密動作。此時，din代表密

文，而 dout 是經過 AES 加解密裝置 1000 解密後的明文。

因為作加密或解密所用之 subkey 順序剛好相反，所以在開始加解密之前，有必要作 subkey 之備份工作，以利加解密之順利進行。subkey 之備份規則如表(1)所示，當將要執行的動作(加密或解密)與上一次動作相同時，作 Reg:Key <= Reg:KeyU 之動作；而不同時則作 Reg:KeyU <= Reg:Key。在 key register 中 subkey 在每個循環運算(each round)之變化情形如表(2)所示，其中係以 AES-128 為例。如此，在每一次的加解密完成後，sub_key0 與 sub_key10 會存在於兩個 key register 中，下一次要加密或解密，就可以很方便的選取所需的 sub_key。

表(1): subkey 之備份規則

開始(Start)	金鑰轉移程序(Key transfer process)
Current_ec == previous_ec	Reg:Key <= Reg:KeyU
Current_ec != previous_ec	Reg:KeyU <= Reg:Key

表(2)：每個循環運算(round)中，key resigter 中之 sub key 變化情形。

	加密		解密	
循環運算	Reg: Key	Reg: KeyU	Reg: Key	Reg: KeyU
開始 (金鑰備份，key backup)	sub_key_0	sub_key_0	sub_key_10	sub_key_10
1	sub_key_1	sub_key_0	sub_key_9	sub_key_10
2	sub_key_2	sub_key_0	sub_key_8	sub_key_10
3	sub_key_3	sub_key_0	sub_key_7	sub_key_10
4	sub_key_4	sub_key_0	sub_key_6	sub_key_10

5	sub_key_5	sub_key_0	sub_key_5	sub_key_10
6	sub_key_6	sub_key_0	sub_key_4	sub_key_10
7	sub_key_7	sub_key_0	sub_key_3	sub_key_10
8	sub_key_8	sub_key_0	sub_key_2	sub_key_10
9	sub_key_9	sub_key_0	sub_key_1	sub_key_10
10，(結束)	sub_key_10	sub_key_0	sub_key_0	sub_key_10

接下來，我們針對第 9 圖中的行混合/逆行混合運算裝置 99，及第 10 圖中的次金鑰更迭運算裝置 800，提出可行的實施方式。

本例中，主要是將行混合(MixColumns)運算與逆行混合(InvMixColumns)運算加以整合，並利用適當的硬體予以實現行混合/逆行混合運算裝置。在 MixColumns 與 InvMixColumns 的運算過程中，兩個 function 之主要運算為：

$$outx = [2\ 3\ 1\ 1]*[a\ b\ c\ d]^T \quad (6)$$

$$outy = [14\ 11\ 13\ 9]*[a\ b\ c\ d]^T \quad (7)$$

在數學上我們作以下的拆解：

$$outx = 2(a + b) + b + (c + d) \quad (8)$$

$$outy = 4(2(a + b) + 2(c + d) + (a + c)) + 2(a+b) + b + (c + d). \quad (9)$$

第(8)式與第(9) 式之運算過程如表(3)所示，由前面五個步驟可以求得 outx，然後在加上 5 個步驟可以求得 outy，所以前面 5 個步驟的硬體可以共用，設計出來的硬體第 6 圖所繪示，將兩 function 之重覆部份整合，可減少不必要之硬體浪費。

步驟	操作方式
1	w1 = a + b

2	w2 = a + c
3	w3 = c + d
4	w4 = 2 * w1
5	outx = b + w3 + w4
6	w5 = 2 * w3
7	w6 = w2 + w4 + w5
8	w7 = 2 * w6
9	w8 = 2 * w7
10	outy = w8 + outx

表(3)

請參照第 6 圖，其繪示依照本發明之所提供的一種可支援先進加密標準之整合型行混合/逆行混合運算裝置方塊圖。如圖所示，整合型行混合/逆行混合運算裝置 600 包括多個互斥或閘及倍增器，互斥或閘用來將兩輸入資料進行互斥或運算後輸出，倍增器則用來將輸入資料乘 2 後輸出。為了簡化說明起見，以下僅針對整合型行混合/逆行混合運算裝置 600 的運作方式加以說明。

MixColumns 與 InvMixColumns 的運算是針對輸入資料中的每一行資料進行矩陣乘法的運算，若輸入資料的資料型態為 4×4 的矩陣，那麼每一行資料中會有 4 個元素資料(element)，為便於說明起見，可將這 4 個元素資料依序標記為資料(a)、資料(b)、資料(c)與資料(d)，並分別與圖式中的 a, b, c, d 對應。請同時參照表(3)，首先說明 MixColumns 的運算步驟：步驟 1 可利用互斥或閘 61 將資料(a)與資料(b)進行互斥或運算後再將資料 W1

輸出而予以實現。步驟 2 可利用互斥或閘 62 將資料(a)與資料(c)進行互斥或運算後再將資料 W2 輸出而予以實現。步驟 3 可利用互斥或閘 63 將資料(c)與資料(d)進行互斥或運算後再將資料 W3 輸出而予以實現。步驟 4 可利用倍增器 621 將互斥或閘 61 之輸出資料 W1 乘 2 後再將資料 W4 輸出而予以實現。步驟 5 可先利用互斥或閘 64 將資料(b)與資料 W3 進行互斥或運算後，再利用互斥或閘 65 將互斥或閘 64 的輸出資料與倍增器 621 之輸出資料 W4 進行互斥或運算後輸出，其中互斥或閘 65 之輸出資料即為整合型行混合/逆行混合運算裝置 600 針對行資料進行行混合運算後之結果。

接著說明執行 InvMixColumns 運算時的資料處理步驟：InvMixColumns 運算的前五個步驟與 MixColumns 運算相同，步驟 6 可利用倍增器 622 將互斥或閘 63 之輸出資料 W3 乘 2 後再將資料 W5 輸出而予以實現。步驟 7 可先利用互斥或閘 66 將資料 W2 與資料 W5 進行互斥或運算後，再利用互斥或閘 67 將互斥或閘 66 的輸出資料與倍增器 621 之輸出資料 W4 進行互斥或運算後將資料 W6 輸出而予以實現。步驟 8 可利用倍增器 623 將互斥或閘 67 之輸出資料 W6 乘 2 後再將資料 W7 輸出而予以實現；步驟 9 可利用倍增器 624 將倍增器 623 之輸出資料 W7 乘 2 後再將資料 W8 輸出而予以實現。步驟 10 可利用互斥或閘 68 將互斥或閘 65 之輸出資料與資料 W8 進行互斥或運算後輸出，其中互斥或閘 68 之輸出資料即為整合型行混合/逆行混合運算裝置 600 針對行資料進行逆行混合運算後之結果。

需要注意的是，由於前五個步驟可為 MixColumns 與 InvMixColumns 所共用，因此可大幅減少不必要的硬體浪費。

本例是提供一種次金鑰更迭運算裝置，可依據目前所輸入的

SubKey 來決定輸出為上一把 SubKey 或下一把 SubKey，而目前所輸入的 SubKey 稱之為已知次金鑰，所欲輸出的 SubKey 稱之為待解次金鑰，首先將說明運算原理。請參照第 7A 圖，其繪示依據輸入的 SubKey 輸出下一把 SubKey 的資料處理方法示意圖。目前的 SubKey 記為 SubKey(i)，下一把 SubKey 記為 SubKey(i+1)；SubKey 的資料型態為 4×4 的矩陣，因此具有 4 組行資料，k[3:0]為行資料(1)，k[7:4]為行資料(2)，k[11:8]為行資料(3)，k[15:12]為行資料(4)，每一行資料中包含 4 個元素，若每一元素的大小為 8 位元，則 SubKey 的長度即為 128 位元。首先將 SubKey(i)的行資料(4)經行資料轉換器 750 轉換為特殊行資料 752 後輸出，行資料轉換器 750 之動作會先將 input 資料作一個 rotate byte right 的動作，之後其第一個 byte 會再跟一個回合常數 Rcon[i]做互斥或運算，然後將 4 個 byte 的結果輸出。在回合常數 Rcon[i]方面，i 為回合數，而不同的回合數有不同的 Rcon 值，依據 AES 標準的定義：Rcon[0]=1, Rcon[i]=Xtime(Rcon[i-1])。接著，特殊行資料 752 與 SubKey(i)的行資料(1)藉互斥或閘 71 進行互斥或運算後即可得到 SubKey(i+1)的行資料(1)。很明顯地，SubKey(i)的行資料(2)與 SubKey(i+1)的行資料(1)藉互斥或閘 72 進行互斥或運算後即可得到 SubKey(i+1)的行資料(2)，SubKey(i)的行資料(3)與 SubKey(i+1)的行資料(2)藉互斥或閘 73 進行互斥或運算後即可得到 SubKey(i+1)的行資料(3)，SubKey(i)的行資料(4)與 SubKey(i+1)的行資料(3)藉互斥或閘 74 進行互斥或運算後即可得到 SubKey(i+1)的行資料(4)。

接著請參照第 7B 圖，其繪示依據輸入的 SubKey 輸出上一把 SubKey 的資料處理方法示意圖。首先將 SubKey(i+1)的行資料(3)

與 SubKey(i+1)的行資料(4)藉互斥或閘 74 進行互斥或運算以得到 SubKey(i)的行資料(4)，而後將 SubKey(i)的行資料(4)經行資料轉換器 750 轉換為特殊行資料 752 後饋入互斥或閘 71 並與 SubKey(i+1)的行資料(1)進行互斥或運算以得到 SubKey(i)的行資料(1)。很明顯地，SubKey(i+1)的行資料(1)與 SubKey(i+1)的行資料(2)藉互斥或閘 72 進行互斥或運算後即可得到 SubKey(i)的行資料(2)，SubKey(i+1)的行資料(2)與 SubKey(i+1)的行資料(3)藉互斥或閘 73 進行互斥或運算後即可得到 SubKey(i)的行資料(3)。

接著請參照第 8 圖，其繪示依照本發明所提供的次金鑰更迭運算裝置方塊圖。次金鑰更迭運算裝置 800 包括行資料轉換器 750、數個互斥或閘及數個多工器。輸入資料 in 為目前的 SubKey(即已知次金鑰)，輸出資料 out 為上一把或下一把 SubKey(即待解次金鑰)，當選擇信號 ec 為 1 時待解次金鑰為下一把 SubKey，當選擇信號 ec 為 0 時待解次金鑰為上一把 SubKey。次金鑰更迭運算裝置 800 包括互斥或閘 71, 72, 73, 74、多工器 710, 720, 730, 740 以及行資料轉換器 750，每一多工器都具有輸入端(0)及輸入端(1)，並依據選擇信號 ec 的值自兩輸入資料中擇一輸出，各元件間的耦接關係如圖中所繪示。已知次金鑰之行資料(1)係饋入互斥或閘 71 與多工器 710 之輸入端(0)，已知次金鑰之行資料(2)係饋入互斥或閘 72 與多工器 720 之輸入端(0)，已知次金鑰之行資料(3)係饋入互斥或閘 73 與多工器 730 之輸入端(0)，已知次金鑰之行資料(4)係饋入互斥或閘 74 與多工器 740 之輸入端(1)。另一方面，互斥或閘 71 之輸出資料為待解次金鑰之行資料(1)並饋入多工器 710 之輸入端(1)，互斥或閘 72 之輸出資料為待解次金鑰之行資料(2)並饋入多工器 720 之輸

入端(1)，互斥或閘 73 之輸出資料為待解次金鑰之行資料(3)並饋入多工器 730 之輸入端(1)，互斥或閘 74 之輸出資料為待解次金鑰之行資料(4)並饋入多工器 740 之輸入端(0)。下文將分別說明以次金鑰更迭運算裝置 800 實現 KeyExpansion 及 InvKeyExpansion 運算的情形。

KeyExpansion 運算(依據輸入的 SubKey 輸出下一把 SubKey)：

令選擇信號 ec 為 1，已知次金鑰的行資料(4)可透過多工器 740 經行資料轉換器 750 轉換為特殊行資料 752 後輸出，而後，特殊行資料 752 與已知次金鑰的行資料(1)藉互斥或閘 71 進行互斥或運算後即可得到下一把 SubKey 的行資料(1)。很明顯地，下一把 SubKey 的行資料(1)可經多工器 710 饋入互斥或閘 72 並與已知次金鑰的行資料(2)進行互斥或運算後得到下一把 SubKey 的行資料(2)，下一把 SubKey 的行資料(2)可經多工器 720 饋入互斥或閘 73 並與已知次金鑰的行資料(3)進行互斥或運算後得到下一把 SubKey 的行資料(3)，下一把 SubKey 的行資料(3)可經多工器 730 饋入互斥或閘 74 並與已知次金鑰的行資料(4)進行互斥或運算後得到下一把 SubKey 的行資料(4)。

InvKeyExpansion 運算(依據輸入的 SubKey 輸出上一把 SubKey)：

令選擇信號 ec 為 0，首先將已知次金鑰的行資料(3)經多工器 730 饋入互斥或閘 74 並與已知次金鑰的行資料(4)進行互斥或運算以得到上一把 SubKey 的行資料(4)，而後將上一把 SubKey 的行資料(4)經多工器 740 輸出並饋入行資料轉換器 750 將其轉換為特殊行資料 752 後饋入互斥或閘 71 並與已知次金鑰的行資料(1)進行互斥或運算以得到上一把 SubKey 的行資料(1)。很明

顯地，已知次金鑰的行資料(1)可經多工器 710 饋入互斥或閘 72 並與已知次金鑰的行資料(2)進行互斥或運算以得到上一把 SubKey 的行資料(2)，已知次金鑰的行資料(2) 可經多工器 720 饋入互斥或閘 73 並與已知次金鑰的行資料(3)進行互斥或運算以得到上一把 SubKey 的行資料(3)。

本發明上述實施例所揭露之可支援先進加密標準之簡化後之整合型次位元/逆次位元運算裝置，具有以下優點：

在執行 SubBytes 與 InvSubBytes 運算時可共用查表資料以節省運算資源，而且因為採用了本發明之簡化電路架構，整體之關鍵路徑(critical paths)變短，其電路運算之複雜度也低，速度因此可以提昇。

也因此，支援先進加密標準之循環運算裝置，及 AES 加解密裝置，也可獲得以上的優點，再者，循環運算裝具有整合型的 MixColumns 與 InvMixColumns 運算硬體以節省運算資源。故此，整體來說，AES 加解密裝置的運算資源因此得到節省而且電路複雜度也降低，運算速度也可增加。

綜上所述，雖然本發明已以一較佳實施例揭露如上，然其並非用以限定本發明，任何熟習此技藝者，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。

【圖式簡單說明】

第 1 圖繪示 AddRoundKey 的資料處理情形。

第 2 圖繪示 ShiftRows 的資料處理情形。

第 3 圖繪示 MixColumns/InvMixColumns 的資料處理情形。

第 4 圖繪示 SubBytes/InvSubBytes 的資料處理情形。

第 5A 圖繪示一種可支援先進加密標準之整合型次位元/逆次位元運算裝置方塊圖。

第 5B 至 5D 圖繪示依照本發明之精神以減化第 5A 圖之整合型次位元/逆次位元運算裝置。

第 5E 圖繪示依照本發之實施例一所提供的一種可支援先進加密標準之整合型次位元/逆次位元運算裝置之方塊圖。

第 6 圖繪示依照本發明所提供的一種可支援先進加密標準之整合型行混合/逆行混合運算裝置方塊圖。

第 7A 圖繪示依據輸入的 SubKey 輸出下一把 SubKey 的資料處理方法示意圖。

第 7B 圖繪示依據輸入的 SubKey 輸出上一把 SubKey 的資料處理方法示意圖。

第 8 圖繪示依照本發明之次金鑰更迭運算裝置方塊圖。

第 9 圖繪示依照本發明之實施例二所提供的一種可支援先進加密標準之循環運算裝置之方塊圖。

第 10 圖繪示依照本發明之實施例三所提供的一種先進加密標準之加解密裝置。

圖式標號說明

61, 62, 63, 64, 65, 66, 67, 68：互斥或閘

71, 72, 73, 74：互斥或閘

90：互斥或閘

93：待解密資料

95：次位元/逆次位元運算裝置

97：列移轉/逆列移轉運算裝置

99：行混合/逆行混合運算裝置

500A, 500B, 500C, 500D：整合型次位元/逆次位元運算裝置

500E：整合型次位元/逆次位元運算裝置

510：矩陣運算器

520：多工器

530：乘法反向運算裝置

540：矩陣運算器

550：多工器

561：第一矩陣運算器

565：第一互斥或運算模組

571：第二矩陣運算器

575：第二互斥或運算模組

590：查表運算裝置

600：整合型行混合/逆行混合運算裝置

621, 622, 623, 624：倍增器

710, 720, 730, 740：多工器

750：行資料轉換器

752：特殊行資料

800：次金鑰更迭運算裝置

900：循環運算裝置

910, 920, 930, 940, 950：多工器

1000：AES加解密裝置

1100：金鑰儲存裝置

1110, 1120, 1130：記憶裝置

ec：選擇信號

伍、中文發明摘要：

一種可支援先進加密標準之加解密裝置，可將次位運算與逆次位元運算加以整合，包括第一矩陣運算器、第一互斥或運算模組、第一多工器、查表運算裝置、第二矩陣運算器、第二互斥或運算模組，及第二多工器，第一多工器自第一矩陣運算器和第一互斥或運算模組之輸出間擇一輸出給查表運算裝置，第二多工器自第二矩陣運算器和第二互斥或運算模組之輸出間擇一輸出。查表運算裝置採用一共用對應表以節省運算資源。此外，上述元件的耦接方式減少了整體關鍵路徑及複雜度，使得此運算模組之速度能有所提高。

陸、英文發明摘要：

An apparatus for supporting advanced encryption standard encryption and decryption combines bytes substitution and inverse bytes substitution operations, and includes first and second matrix operation devices, first and second exclusive-OR operation modules, first and second multiplexers, and a table-look-up device. The first multiplexer selects one from the outputs of the first matrix operation device and first exclusive-OR operation module. The second multiplexer selects one from the outputs of the second matrix operation device and second exclusive-OR operation module. The table-look-up device applies a common look-up table so as to save operation resources. In addition, the elements of the encryption apparatus are connected in a way such that the entire critical paths and complexity are reduced, thus improving the speed of the apparatus.

柒、指定代表圖：

(一)本案指定代表圖為：第 5E 圖。

(二)本代表圖之元件代表符號簡單說明:

500E：整合型次位元/逆次位元運算裝置

520：多工器

550：多工器

561：第一矩陣運算器

565：第一互斥或運算模組

571：第二矩陣運算器

575：第二互斥或運算模組

590：查表運算裝置

ec：選擇信號

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

拾、申請專利範圍：

1. 一種可支援先進加密標準(Advanced Encryption Standard, AES)之整合型次位元(SubBytes)/逆次位元(InvSubBytes)運算裝置，用以針對一輸入資料碼，選擇性地進行次位元和逆次位元運算後輸出一欲求之輸出資料碼，該整合型次位元/逆次位元運算裝置包括：

一第一矩陣運算器，用以針對該輸入資料碼進行一第一矩陣運算，並輸出該第一矩陣運算之結果；

一第一互斥或(exclusive-OR)運算模組，用以針對該輸入資料碼進行一第一互斥或運算，並輸出該第一互斥或運算之結果；

一第一多工器，與該第一矩陣運算器及該第一互斥或運算模組耦接，該第一多工器係依據一選擇信號，自該第一矩陣運算之結果及該第一互斥或運算之結果二者間擇一輸出，以作為該第一多工器之輸出資料碼；

一查表運算裝置，與該第一多工器耦接，用以依據該第一多工器之輸出資料碼，進行一查表運算後輸出一查表資料碼；

一第二矩陣運算器，耦接至該查表運算裝置，用以針對該查表資料碼進行一第二矩陣運算，並輸出該第二矩陣運算之結果；

一第二互斥或運算模組，用以針對該查表資料碼進行一第二互斥或運算，並輸出該第二互斥或運算之結果；以及

一第二多工器，與該第二矩陣運算模組及該第二互斥或運算模組耦接，該第二多工器係依據該選擇信號自該第二矩陣運算之結果及該第二互斥或運算之結果二者間擇一輸出，以作為該第二多工器之輸出資料碼；

其中，該第二多工器之輸出資料碼即為該欲求之輸出資料碼。

2. 如申請專利範圍第 1 項所述之可支援先進加密標準之整

合型次位元/逆次位元運算裝置，係於該選擇信號代表需要進行加密時，進行次位元運算，其中該第一多工器選擇該第一互斥或運算之結果，該第二多工器選擇該第二互斥或運算之結果。

3. 如申請專利範圍第 1 項所述之可支援先進加密標準之整合型次位元/逆次位元運算裝置，係於該選擇信號代表需要進行解密時，進行逆次位元運算，其中該第一多工器選擇該第一矩陣運算之結果，該第二多工器選擇該第二矩陣運算之結果。

4. 如申請專利範圍第 1 項所述之可支援先進加密標準之整合型次位元/逆次位元運算裝置，其中該第一矩陣運算與第二矩陣運算實質上相同。

5. 如申請專利範圍第 1 項所述之可支援先進加密標準之整合型次位元/逆次位元運算裝置，其中該第一互斥或運算包含一運算元，該運算元之值係基於該第一矩陣運算及該第二互斥或運算所得者。

6. 如申請專利範圍第 1 項所述之可支援先進加密標準之整合型次位元/逆次位元運算裝置，其中該查表運算模組包含一對照表，該對照表係基於一乘法反向運算(multiplicative inverse operation)、該第一互斥或運算，及該第一矩陣運算所得者。

7. 一種可支援先進加密標準之循環運算模組(round module)，用以依據一輸入資料碼及一次金鑰選擇性地進行加密/解密運算後產生一輸出資料碼，該循環運算模組包括：

一互斥或閘，用以將該輸入資料碼與該次金鑰進行互斥或運算後產生該互斥或閘之輸出碼；

一第一多工器，與該互斥或閘耦接，該第一多工器具有一第一輸入端及一第二輸入端，該第一輸入端係用以接收一待解密資料碼且該第二輸入端用以接收該互斥或閘之輸出碼，其中，該第

一多工器係依據一選擇信號自該待解密資料碼與該互斥或閘之輸出碼二者間擇一輸出該第一多工器之輸出碼；

一次位元/逆次位元運算裝置，耦接至該第一多工器，用以將該第一多工器之輸出碼進行次位元/逆次位元運算後輸出一替代運算輸出碼，該次位元/逆次位元運算裝置包括：

一第一矩陣運算器，用以針對該第一多工器之輸出碼進行一第一矩陣運算，並輸出該第一矩陣運算之結果；

一第一互斥或(exclusive-OR)運算模組，用以針對該第一多工器之輸出碼進行一第一互斥或運算，並輸出該第一互斥或運算之結果；

一第一選擇器，與該第一矩陣運算器及該第一互斥或運算模組耦接，該第一選擇器係依據該選擇信號，自該第一矩陣運算之結果及該第一互斥或運算之結果二者間擇一輸出，以作為該第一選擇器之輸出碼；

一查表運算裝置，耦接至該第一選擇器，用以依據該第一選擇器之輸出碼，進行一查表運算後輸出一查表資料碼；

一第二矩陣運算器，用以針對該查表資料碼進行一第二矩陣運算，並輸出該第二矩陣運算之結果；

一第二互斥或運算模組，用以針對該查表資料碼進行一第二互斥或運算，並輸出該第二互斥或運算之結果；以及

一第二選擇器，與該第二矩陣運算器及該第二互斥或運算模組耦接，該第二選擇器係依據該選擇信號自該第二矩陣運算之結果及該第二互斥或運算之結果二者間擇一輸出，以作為該替代運算輸出碼；

一列移轉(ShiftRows)/逆列移轉(InvShiftRows)運算裝置，耦接至該次位元/逆次位元運算裝置，用以將該替代運算輸

出碼進行列移轉/逆列移轉運算後輸出一移轉運算輸出碼；

一第二多工器，與該互斥或閘及該列移轉/逆列移轉運算裝置耦接，該第二多工器具有一第一輸入端及一第二輸入端，該第一輸入端係用以接收該互斥或閘之輸出碼且該第二輸入端用以接收該移轉運算輸出碼，其中，該第二多工器係依據該選擇信號自該互斥或閘之輸出碼與該移轉運算輸出碼二者間擇一輸出該第二多工器之輸出碼；

一行混合/逆行混合運算裝置，與該第二多工器耦接，用以將該第二多工器之輸出碼進行行混合/逆行混合運算後輸出一混合運算輸出碼；

一第三多工器，與該第二多工器及該行混合/逆行混合運算裝置耦接，該第三多工器具有一第一輸入端及一第二輸入端，該第一輸入端係用以接收該第二多工器之輸出碼且該第二輸入端用以接收該混合運算輸出碼，其中，該第三多工器係依據一加解密盡判斷信號自該第二多工器之輸出碼與該混合運算輸出碼二者間擇一輸出，且該第三多工器之輸出碼即為該待解密資料碼；

一第四多工器，與該第三多工器及該列移轉/逆列移轉運算裝置耦接，該第四多工器具有一第一輸入端及一第二輸入端，該第一輸入端係用以接收該移轉運算輸出碼且該第二輸入端用以接收該待解密資料碼，其中，該第四多工器係依據該選擇信號自該移轉運算輸出碼與該待解密資料碼二者間擇一輸出該第四多工器之輸出碼；以及

一第五多工器，與該第四多工器及該互斥或閘耦接，該第五多工器具有一第一輸入端及一第二輸入端，該第一輸入端係用以接收該第四多工器之輸出碼且該第二輸入端用以接收該互斥或閘之輸出碼，其中，該第五多工器係依據一回合盡判斷信號自該

第四多工器之輸出碼與該互斥或閘之輸出碼二者間擇一輸出，以作為該第五多工器之輸出碼；

其中，該第五多工器之輸出碼即為該輸出資料碼。

8. 如申請專利範圍第 7 項所述之可支援先進加密標準之循環運算模組，其中，當該選擇信號代表需要進行加密時，該次位元/逆次位元運算裝置進行次位元運算，其中該第一選擇器選擇該第一互斥或運算之結果，該第二選擇器選擇該第二互斥或運算之結果。

9. 如申請專利範圍第 7 項所述之可支援先進加密標準之循環運算模組，其中，當該選擇信號代表需要進行解密時，該次位元/逆次位元運算裝置進行逆次位元運算，其中該第一選擇器選擇該第一矩陣運算之結果，該第二選擇器選擇該第二矩陣運算之結果。

10. 如申請專利範圍第 7 項所述之可支援先進加密標準之循環運算模組，其中該第一矩陣運算與第二矩陣運算實質上相同。

11. 如申請專利範圍第 7 項所述之可支援先進加密標準之循環運算模組，其中該第一互斥或運算包含一運算元，該運算元之值係基於該第一矩陣運算及該第二互斥或運算所得者。

12. 如申請專利範圍第 7 項所述之可支援先進加密標準之循環運算模組，其中該查表運算模組包含一對照表，該對照表係基於一乘法反向運算(multiplicative inverse operation)、該第一互斥或運算，及該第一矩陣運算所得者。

13. 一種先進加密標準之加解密裝置，用以選擇性的針對一輸入資料碼進行先進加密標準之加密或解密的動作，以輸出一輸出資料碼，該加解密裝置包括：

一循環運算裝置，與該金鑰儲存裝置耦接，用以撰擇性地進行加密及解密之一在進行中所需之循環運算，依據輸入該循環運算裝置之一輸入碼及一次金鑰以輸出一循環運算之輸出碼；

一次金鑰更迭運算裝置，與該循環運算裝置耦接，用以撰擇性地產生進行加密及解密之一時，循環運算所需之該次金鑰，其中該次金鑰係基於輸該次金鑰更迭運算裝置之一已知次金鑰而得之一待解次金鑰；以及

一金鑰儲存裝置，與該循環運算裝置及該次金鑰更迭運算裝置耦接，用以作次金鑰的暫存及分配，以便該次金鑰更迭運算裝置及該循環運算裝置進行循環運算；

其中，該循環運算裝置包括一次位元/逆次位元運算裝置，該次位元/逆次位元運算裝置用以將基於該循環運算裝置之該輸入碼及該次金鑰之一運算輸入碼進行次位元/逆次位元運算後輸出一替代運算輸出碼，該次位元/逆次位元運算裝置包括：

一第一矩陣運算器，用以針對該運算輸入碼進行一第一矩陣運算，並輸出該第一矩陣運算之結果；

一第一互斥或(exclusive-OR)運算模組，用以針對該運算輸入碼進行一第一互斥或運算，並輸出該第一互斥或運算之結果；

一第一選擇器，與該第一矩陣運算器及該第一互斥或運算模組耦接，該第一選擇器係依據該選擇信號，自該第一矩陣運算之結果及該第一互斥或運算之結果二者間擇一輸出，以作為該第一選擇器之輸出碼；

一查表運算裝置，耦接至該第一選擇器，用以依據該第一選擇器之輸出碼，進行一查表運算後輸出一查表資料碼；

一第二矩陣運算器，用以針對該查表資料碼進行一第

二矩陣運算，並輸出該第二矩陣運算之結果；

一第二互斥或運算模組，用以針對該查表資料碼進行一第二互斥或運算，並輸出該第二互斥或運算之結果；以及

一第二選擇器，與該第二矩陣運算器及該第二互斥或運算模組耦接，該第二選擇器係依據該選擇信號自該第二矩陣運算之結果及該第二互斥或運算之結果二者間擇一輸出，以作為該替代運算輸出碼；

其中，該金鑰儲存裝置接收該循環運算輸出碼及該次金鑰更迭運算裝置輸出之該次金鑰；該次金鑰更迭運算裝置之該已知次金鑰及該循環運算裝置之該輸入碼係為該金鑰儲存裝置所輸出；該金鑰儲存裝置暫存該輸入資料碼，進行金鑰之分配及暫存，接收該循環運算裝置及該次金鑰更迭運算裝置之輸出以進行循環運算，並輸出該輸出資料碼。

14. 如申請專利範圍第 13 項所述之先進加密標準之加解密裝置，其中該次金鑰更迭運算裝置，用以依據該已知次金鑰求得該待解次金鑰，其中該已知次金鑰與該待解次金鑰係兩相鄰之次金鑰，且該已知次金鑰與該待解次金鑰各具有一行資料(1)、一行資料(2)、一行資料(3)及一行資料(4)，該次金鑰更迭運算裝置包括：

一行資料轉換器，用以將一行資料轉換為一特殊行資料後輸出；

一互斥或閘(1)，用以將該已知次金鑰之行資料(1)與該特殊行資料進行互斥或運算後輸出，其中該互斥或閘(1)之輸出資料即為該待解次金鑰之行資料(1)；

一多工器(1)，與該互斥或閘(1)耦接，該多工器(1)具有一第一輸入端 及一輸入端(1)，該第一輸入端係用以接收該已知次

金鑰之行資料(1)且該輸入端(1)用以接收該待解次金鑰之行資料(1)，該多工器(1)係依據一選擇信號自該已知次金鑰之行資料(1)與該待解次金鑰之行資料(1)二者間擇一輸出；

一互斥或閘(2)，用以將該已知次金鑰之行資料(2)與該多工器(1)之輸出資料進行互斥或運算後輸出，其中該互斥或閘(2)之輸出資料即為該待解次金鑰之行資料(2)；

一多工器(2)，與該互斥或閘(2)耦接，該多工器(2)具有一第一輸入端 及一輸入端(1)，該多工器(2)之第一輸入端係用以接收該已知次金鑰之行資料(2)且該多工器(2)之輸入端(1)用以接收該待解次金鑰之行資料(2)，該多工器(2)係依據該選擇信號自該已知次金鑰之行資料(2)與該待解次金鑰之行資料(2)二者間擇一輸出；

一互斥或閘(3)，用以將該已知次金鑰之行資料(3)與該多工器(2)之輸出資料進行互斥或運算後輸出，其中該互斥或閘(3)之輸出資料即為該待解次金鑰之行資料(3)；

一多工器(3)，與該互斥或閘(3)耦接，該多工器(3)具有一輸入端(0) 及一輸入端(1)，該多工器(3)之輸入端(0)係用以接收該已知次金鑰之行資料(3)且該多工器(3)之輸入端(1)用以接收該待解次金鑰之行資料(3)，該多工器(3)係依據該選擇信號自該已知次金鑰之行資料(3)與該待解次金鑰之行資料(3)二者間擇一輸出；

一互斥或閘(4)，用以將該已知次金鑰之行資料(4)與該多工器(3)之輸出資料進行互斥或運算後輸出，其中該互斥或閘(4)之輸出資料即為該待解次金鑰之行資料(4)；以及

一多工器(4)，與該互斥或閘(4)耦接，該多工器(4)具有一輸入端(0) 及一輸入端(1)，該多工器(4)之輸入端(1)係用以接

收該已知次金鑰之行資料(4)且該多工器(4)之輸入端(0)用以接收該待解次金鑰之行資料(4)，該多工器(4)係依據該選擇信號自該已知次金鑰之行資料(4)與該待解次金鑰之行資料(4)二者間擇一輸出，且該多工器(4)之輸出資料即為該行資料。

15. 如申請專利範圍第 13 項所述之先進加密標準之加解密裝置，其中，當該選擇信號代表需要進行加密時，該次位元/逆次位元運算裝置進行次位元運算，其中該第一選擇器選擇該第一互斥或運算之結果，該第二選擇器選擇該第二互斥或運算之結果。

16. 如申請專利範圍第 13 項所述之先進加密標準之加解密裝置，其中，當該選擇信號代表需要進行解密時，該次位元/逆次位元運算裝置進行逆次位元運算，其中該第一選擇器選擇該第一矩陣運算之結果，該第二選擇器選擇該第二矩陣運算之結果。

17. 如申請專利範圍第 13 項所述之先進加密標準之加解密裝置，其中該第一矩陣運算與第二矩陣運算實質上相同。

18. 如申請專利範圍第 13 項所述之先進加密標準之加解密裝置，其中該第一互斥或運算包含一運算元，該運算元之值係基於該第一矩陣運算及該第二互斥或運算所得者。

19. 如申請專利範圍第 13 項所述之先進加密標準之加解密裝置，其中該查表運算模組包含一對照表，該對照表係基於一乘法反向運算(multiplicative inverse operation)、該第一互斥或運算，及該第一矩陣運算所得者。

in0	in4	in8	in12
in1	in5	in9	in13
in2	in6	in10	in14
in3	in7	in11	in15

⊕

k0	k4	k8	k12
k1	k5	k9	k13
k2	k6	k10	k14
k3	k7	k11	k15

=

out0	out4	out8	out12
out1	out5	out9	out13
out2	out6	out10	out14
out3	out7	out11	out15

in0	in4	in8	in12
in1	in5	in9	in13
in2	in6	in10	in14
in3	in7	in11	in15

=

k0	k4	k8	k12
k1	k5	k9	k13
k2	k6	k10	k14
k3	k7	k11	k15

⊕

out0	out4	out8	out12
out1	out5	out9	out13
out2	out6	out10	out14
out3	out7	out11	out15

第 1 圖(習知技藝)

in0	in4	in8	in12
in1	in5	in9	in13
in2	in6	in10	in14
in3	in7	in11	in15

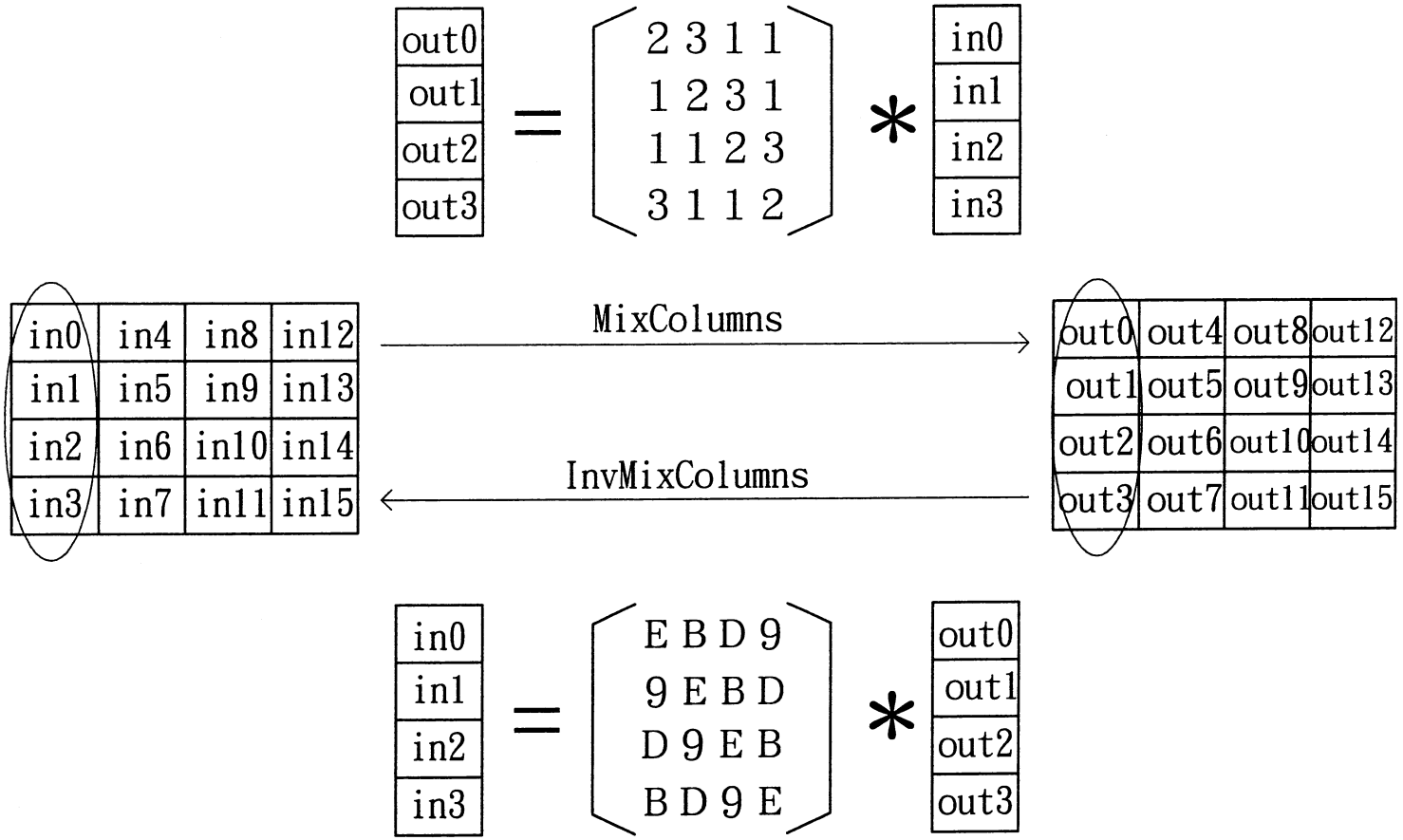
>>0B →
>>1B →
>>2B →
>>3B →

in0	in4	in8	in12
in13	in1	in5	in9
in10	in14	in2	in6
in7	in11	in15	in3

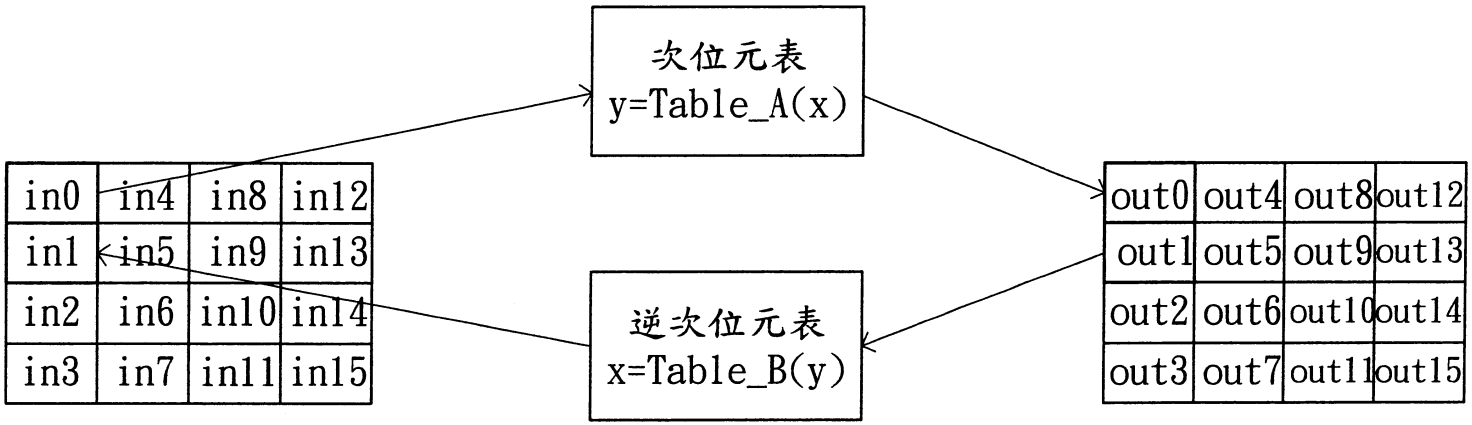
=

out0	out4	out8	out12
out1	out5	out9	out13
out2	out6	out10	out14
out3	out7	out11	out15

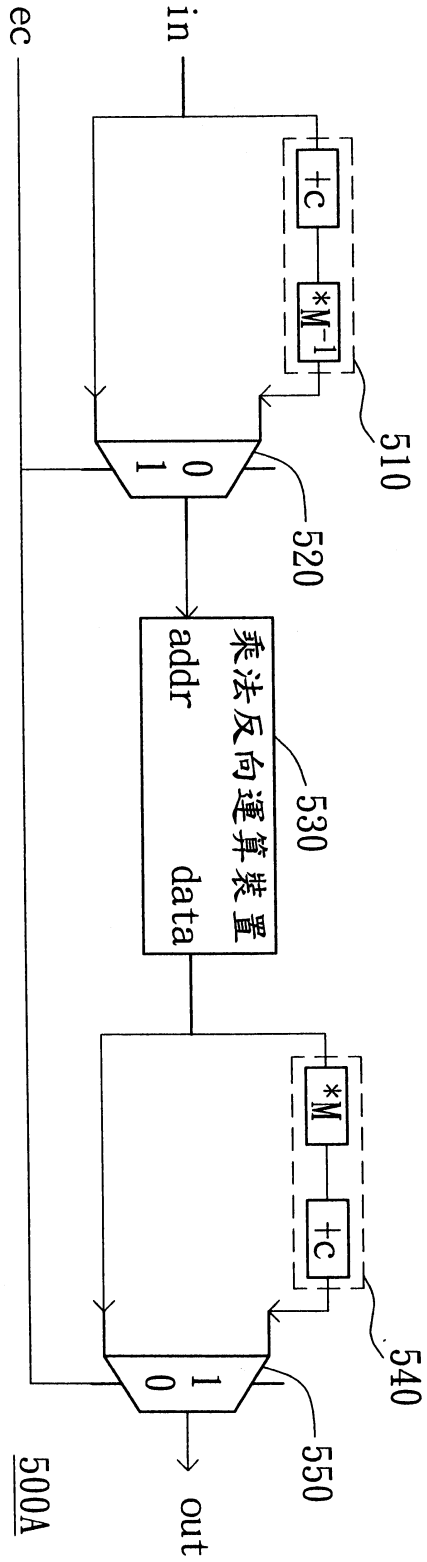
第 2 圖(習知技藝)


out0
out1
out2
out3
=
2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2
*
in0
in1
in2
in3
in0 in4 in8 in12
in1 in5 in9 in13
in2 in6 in10 in14
in3 in7 in11 in15
MixColumns
InvMixColumns
out0 out4 out8 out12
out1 out5 out9 out13
out2 out6 out10 out14
out3 out7 out11 out15
in0
in1
in2
in3
=
E B D 9
9 E B D
D 9 E B
B D 9 E
*
out0
out1
out2
out3

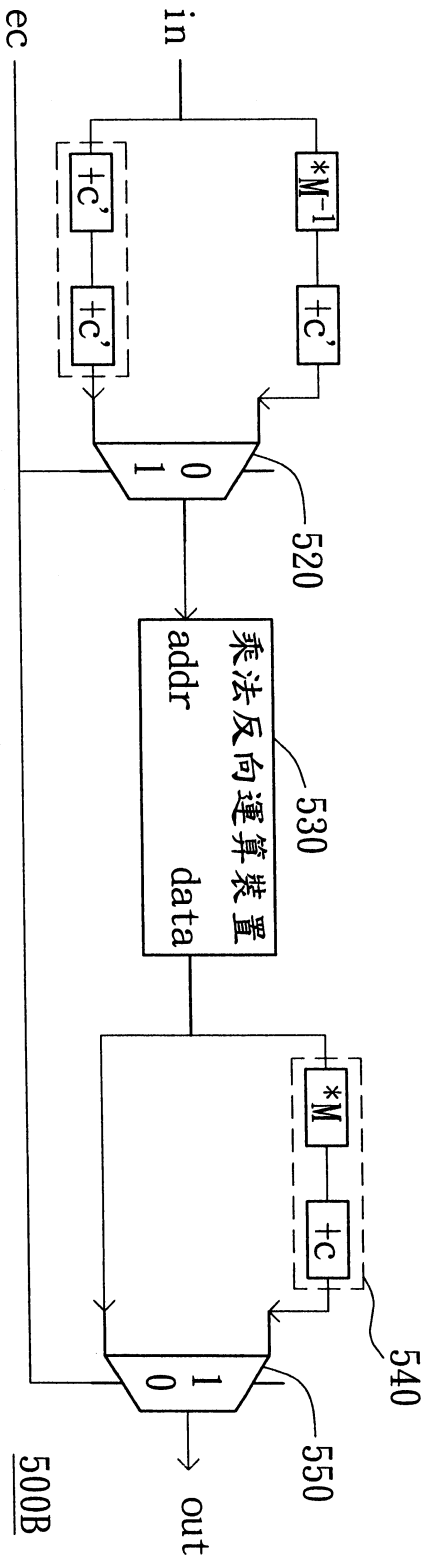
第 3 圖(習知技藝)


次位元表
y=Table_A(x)
逆次位元表
x=Table_B(y)
in0 in4 in8 in12
in1 in5 in9 in13
in2 in6 in10 in14
in3 in7 in11 in15
out0 out4 out8 out12
out1 out5 out9 out13
out2 out6 out10 out14
out3 out7 out11 out15

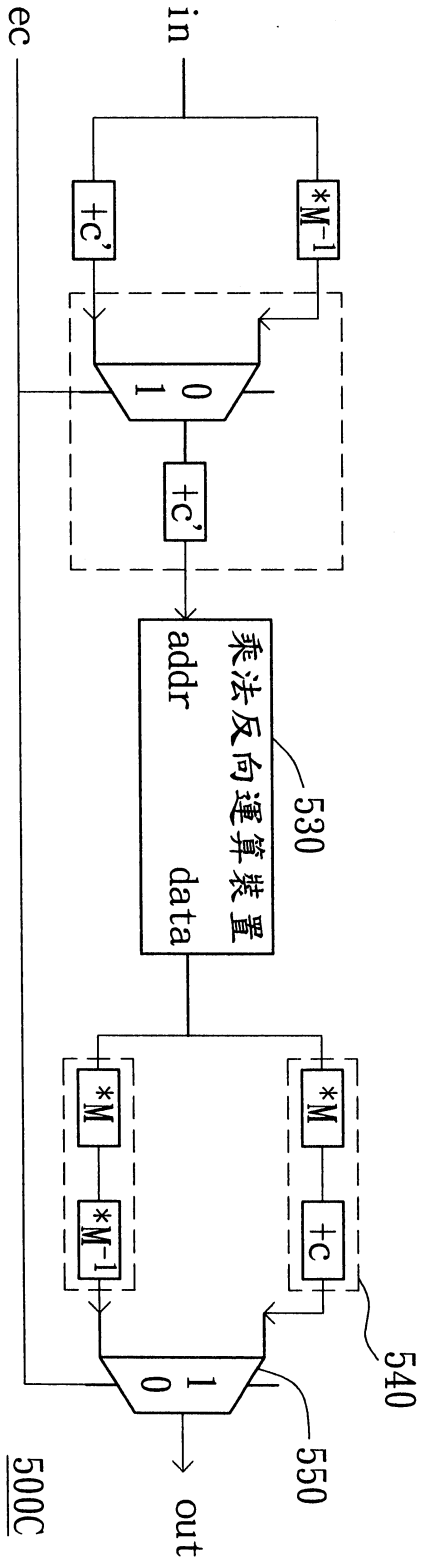
第 4 圖(習知技藝)


ec
in
+c
*M-1
510
0
1
520
乘法反向運算裝置
addr
data
530
*M
+c
540
1
0
550
out
500A

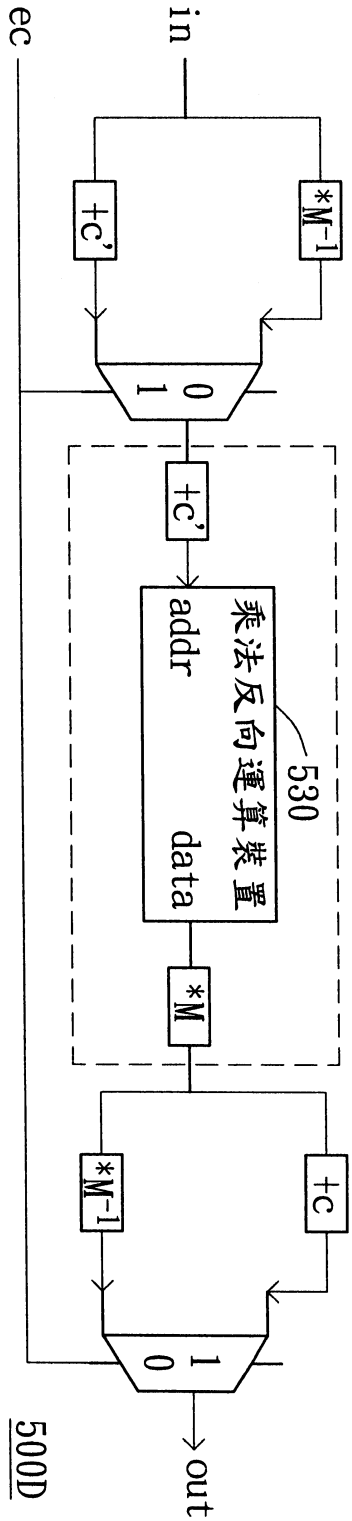
第 5A 圖


ec
in
*M-1
+c'
+c'
+c'
0
1
520
乘法反向運算裝置
addr
data
530
*M
+c
540
1
0
550
out
500B

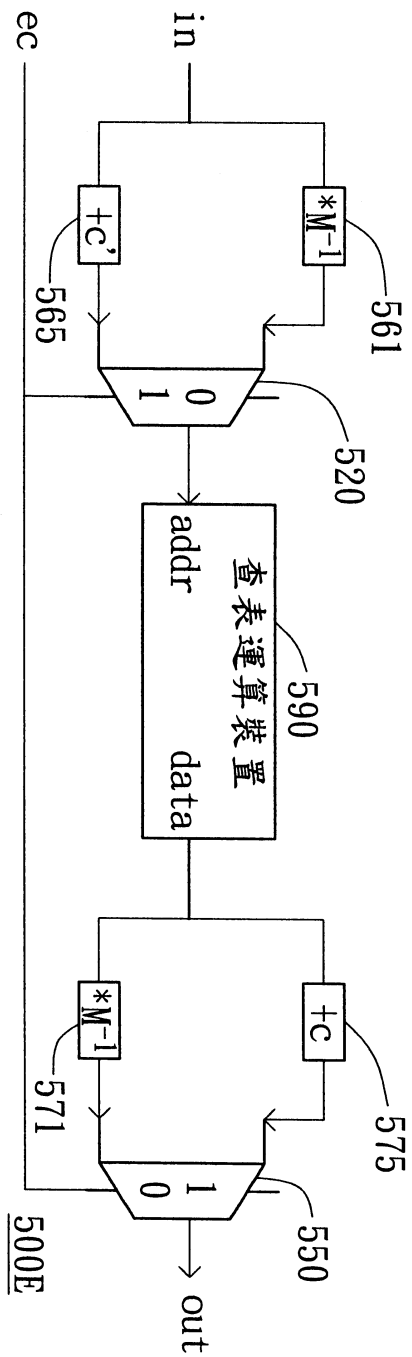
第 5B 圖


in
ec
*M-1
+c'
0
1
+c'
乘法反向運算裝置
addr
data
530
*M
+c
*M
*M-1
1
0
540
550
out
500C

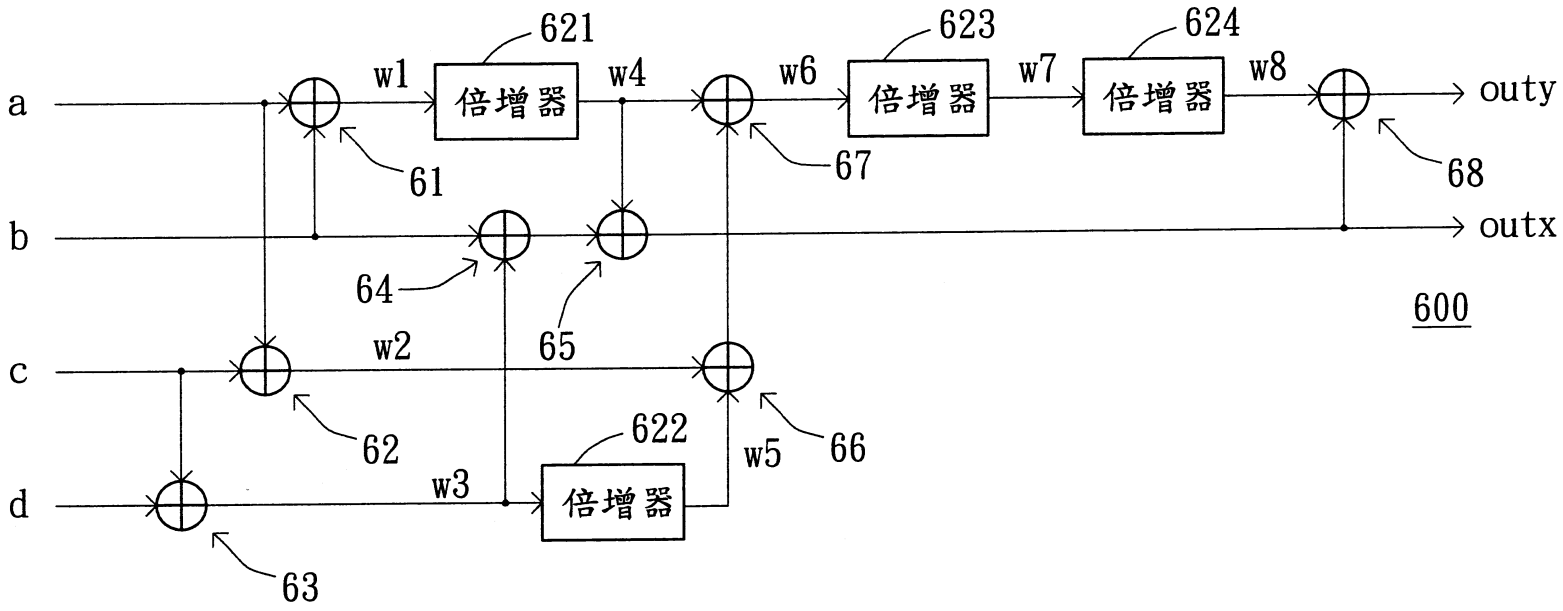
第 5C 圖


in
ec
*M-1
+c'
0
1
+c'
乘法反向運算裝置
addr
data
530
*M
*M-1
+c
1
0
out
500D

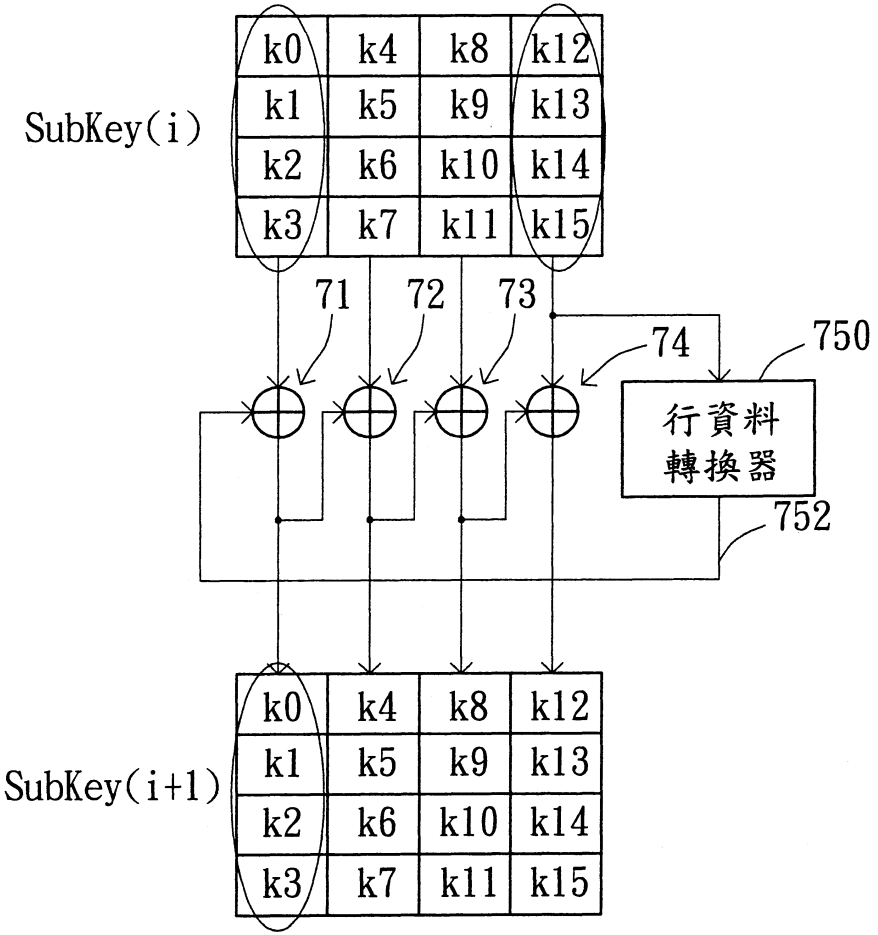
第 5D 圖


in
ec
*M-1
+c'
561
565
0
1
520
查表運算裝置
addr
data
590
*M-1
+c
571
575
1
0
550
500E
out

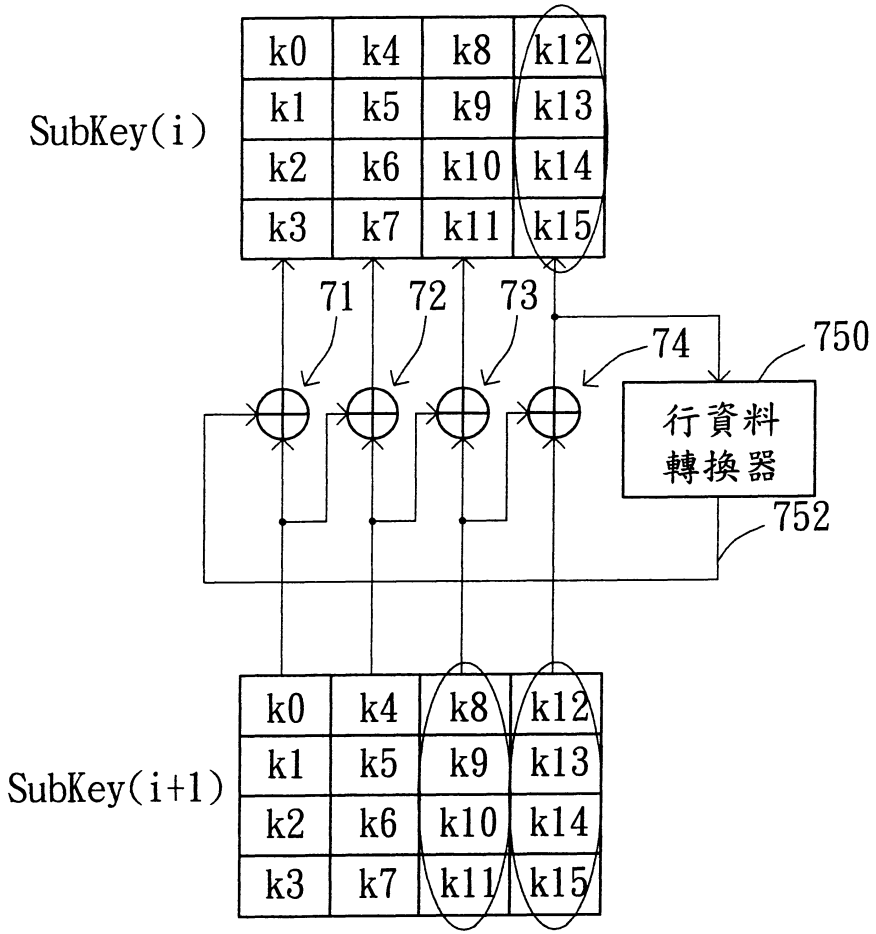
第 5E 圖


621
623
624
w1
w4
w6
w7
w8
倍增器
倍增器
倍增器
a
b
c
d
outy
outx
61
67
68
64
65
600
w2
62
66
622
w5
w3
倍增器
63

第 6 圖

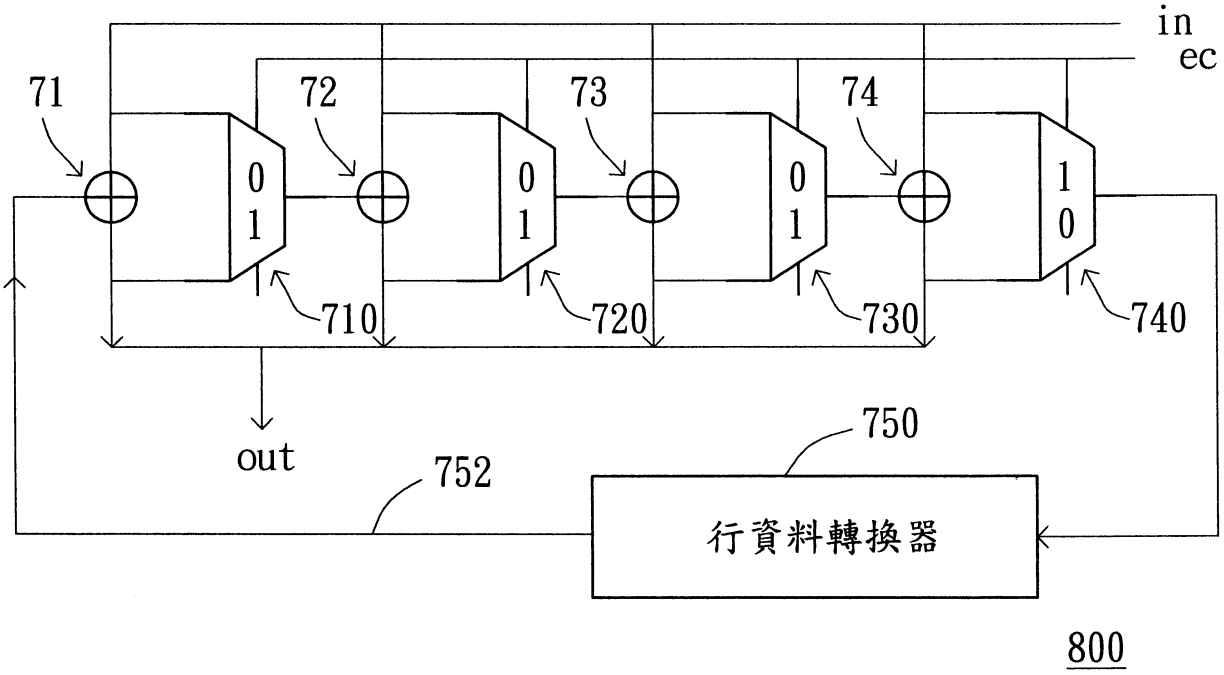


第 7A 圖

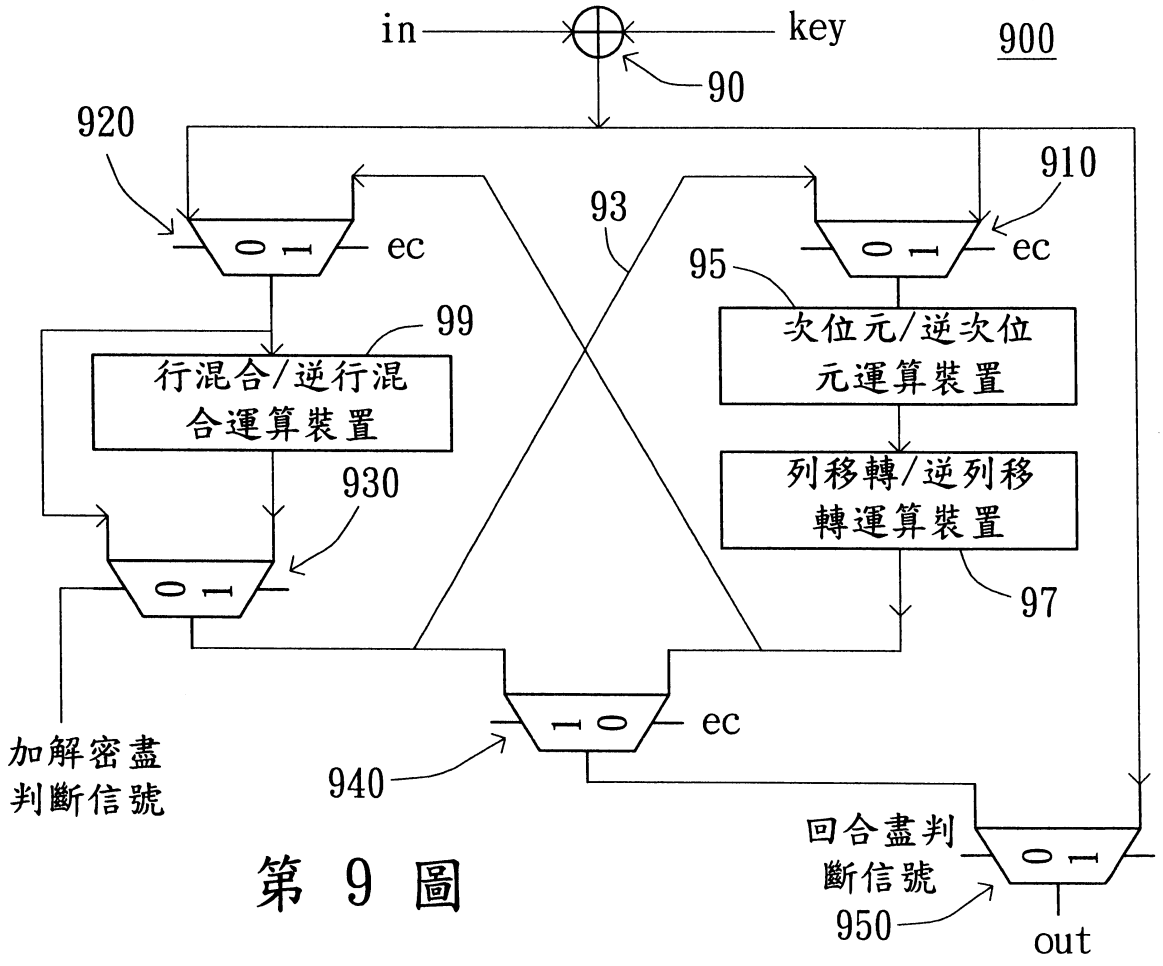


第 7B 圖

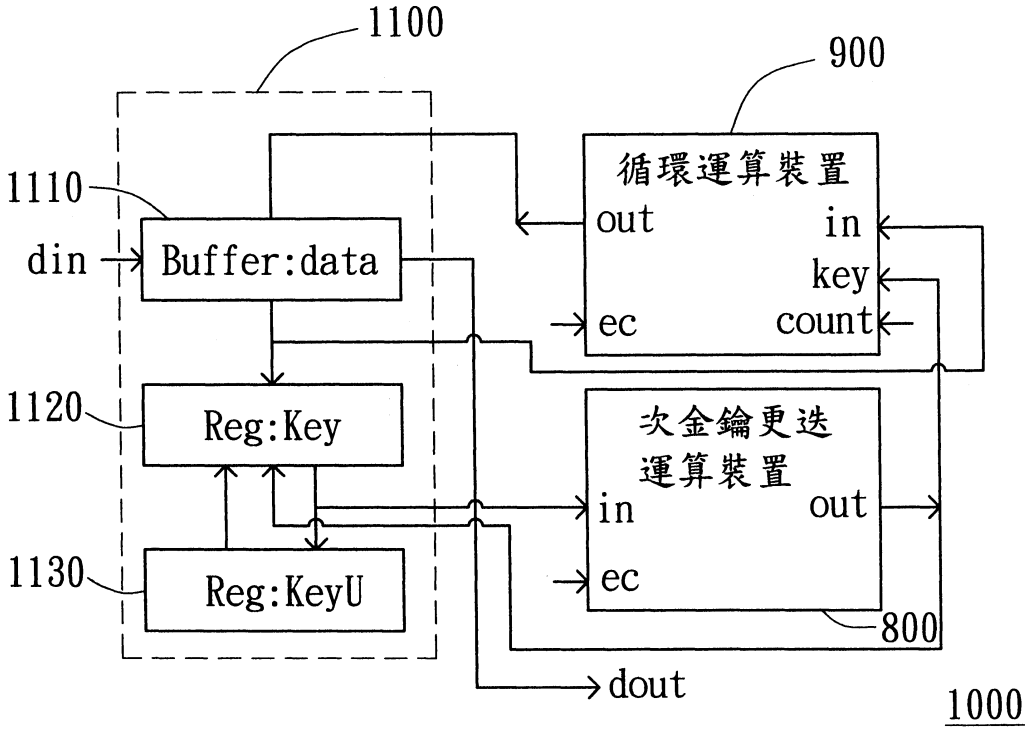
I235582


in
ec
71
72
73
74
0
1
1
0
710
720
730
740
out
750
752
行資料轉換器
800

第 8 圖


in
key
900
90
920
910
93
ec
ec
95
99
次位元/逆次位元運算裝置
行混合/逆行混合運算裝置
列移轉/逆列移轉運算裝置
930
97
ec
940
加解密盡判斷信號
回合盡判斷信號
950
out

第 9 圖


1100
900
循環運算裝置
out
in
key
ec
count
1110
din
Buffer:data
1120
Reg:Key
次金鑰更迭
運算裝置
in
out
1130
Reg:KeyU
ec
800
dout
1000

第 10 圖