公告本

378294

申請日期	86 年 6 月 3 日
案號	86107636
類別	G06F 12/16

（以上各欄由本局填註）

A4
C4

378294

發明 ~~新型~~ 專利說明書

一、發明 ~~新型~~ 名稱	中文	資料處理裝置以及微電腦
	英文	
二、發明 ~~創作~~ 人	姓名	(1) 中田邦彥
	國籍	(1) 日本
	住、居所	(1) 日本國東京都小金井市前原町三－二五－二九
三、申請人	姓名（名稱）	(1) 日立製作所股份有限公司 株式会社日立製作所
	國籍	(1) 日本
	住、居所（事務所）	(1) 日本國東京都千代田區神田駿河台四丁目六番地
	代表人姓名	(1) 金井務

裝 訂 線

經濟部中央標準局員工消費合作社印製

（由本局填寫）

承辦人代碼：	
大　　類：	
ＩＰＣ分類：	

A6
B6

本案已向：

國（地區）	申請專利，申請日期：	案號：	，□有　□無主張優先權
日本	1996 年 6 月 28 日	8-169322	☑有主張優先權

有關微生物已寄存於：　　，寄存日期：　　，寄存號碼：

（請先閱讀背面之注意事項再填寫本頁各欄）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（1）

本發明係關於適用具備有剩餘演算機能的資料處理裝置而有效之技術，特別是關於適用於使用剩餘乘算演算或是冪函數剩餘演算之密碼化以及解碼化裝置而有效之技術。

公開鎖匙密碼技術係密碼技術之一。例如第14圖所示，送訊者對於訊息M藉由「$C = M^e \bmod N$」的演算法密碼化而形成密碼文C，受訊者對於密碼文C藉由「$M = C^d \bmod N$」的演算法解碼化而形成訊息M。上述密碼化的過程所採用的參數e，N係公開的，而上述解碼過程所採用的d則非公開。前述密碼化以及解碼化所使用的演算式，係冪函數剩餘演算，可以用通式「$X^Y \bmod N$」來代表記述（X，Y，N爲正的整數）。「mod」意味著剩餘演算，亦即將無法整除的部分以「餘」值來表示，冪函數剩餘演算「$X^Y \bmod N$」係將X^Y以N除之而將其「餘」之值作爲解。

於密碼化／解碼化裝置，藉由使用冪函數剩餘演算「$X^Y \bmod N$」（X，Y，N爲正整數），可以得到高度的機密性以廣爲所知，例如，池野氏與小山氏所著之「現代暗號理論」電子情報通信學會編（昭和61年出版）就有其記載。

此處，因爲前述之X，Y，N通常使用100位元～2000位元的非常大的數的緣故，所以要如何使「$X^Y \bmod N$」能夠高速的實行其演算在工程學以及數學的領域來說都是一項課題。

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（2）

解法之一是由下式所示之古典解法（算法1）。

「算法1」

input X，$Y = e_n e_{n-1} \cdots e_1$，N　　步驟1

A = X　　步驟2

B = X　　步驟3

for i = n-1 to 1 step -1 {　　步驟4

　　$A = A^2 \bmod N$　　步驟5

　　if e_i=1 then $A = A \cdot B \bmod N$　　步驟6

}　　步驟7

output A　　步驟8

於此算法中n對應於Y的位元數，$e_n e_{n-1} \cdots e_1$係Y的2進位表現法。此算法概略上係將自乘的剩餘乘算「$A^2 \bmod N$」（步驟5）以及剩餘乘算「$A \cdot B \bmod N$」（步驟6）組合而實行之，$e_n e_{n-1} \cdots e_1$之論理值爲1的個數若爲r（e），前述自乘的剩餘乘算「$A^2 \bmod N$」反覆進行n－1回，而前述剩餘乘算「$A \cdot B \bmod N$」則反覆進行r（e）回。

如此，在「算法1」，因爲係將冪函數剩餘演算「$X^Y \bmod N$」分解爲剩餘乘算「$A \cdot B \bmod N$」等而實行的緣故，所以只要使用實現「$A \cdot B \bmod N$」的演算機能的演算裝置即可。

然而，因爲A，B也是非常大的數字，所以例如資料

（請先閱讀背面之注意事項再填寫本頁）

裝　訂　線

經濟部中央標準局員工消費合作社印製

五、發明說明（3）

長度各爲５１２位元的話，中間結果ＡＢ乘起來是１０２４位元的成爲問題的龐大數字。進而Ａ・Ｂ以Ｎ除之而求其餘值作爲最終結果的緣故，必須要進行１０２４位元÷５１２位元的等級的龐大值的演算。此處的乘算係藉由將乘數與被乘數分割而可以藉由微處理器進行平行高速處理，但是除法運算要藉由平行高速處理是有困難的，即使是將其分解爲剩餘乘算「A・B mod N」而實行計算，該除法運算處理依舊是阻礙高速化運算的問題。

特別是，爲了要解決上述剩餘乘算「A・B mod N」的除法運算的問題，已知的解法還有不以Ｎ來進行除法運算，而是實行「$A \cdot B \cdot R^{-1} \bmod N$」的以下的「算法２」廣爲人知。此處，Ｒ係$2^n$（ｎ係例如Ｎ的位元數），而且係滿足Ｒ＞Ｎ的正的整數。

關於不藉由Ｎ來進行除法運算而是實行「$A \cdot B \cdot R^{-1} \bmod N$」「算法２」的詳細論證在此並不加以詳述，可以參考例如Montgomery. P. L. : Modular Multiplication without Trial Division, Mathematica of Computation, Vol. 44, No. 170, pp.519-521(1985). Dusse, S.R. and Kaliski, B. S. Jr.: A Cryptographic Library for the Motorol a DSP56000. Advances in Cryptology - EUROCRYPTO'90. Lecture Notes in Computer Science 473. Pp. 230-244, Springer-Verlag(1991)等之記載。

「算法２」

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（4）

$N' = -N^{-1} \bmod R$ 步驟 1

$M = A \cdot B \cdot N^{-1} \bmod R$ 步驟 2

$t = (A \cdot B + M \cdot N)/R$ 步驟 3

if $t \geqq N$ then return $t-N$ else return t 步驟 4

發明概要

本發明之發明人經過仔細檢討，發現前法具有如下之問題點。

第 1 個問題點

在上述「算法 2」中，其步驟 1、步驟 2 的 N', M必須要經由前處理事先計算而得之。此外也必須要新增暫時保管 M 的值的記憶裝置。

第 2 個問題點

進而，於前述「算法 2」，在其步驟 4 必須要進行 t 與 N 的大小比較，因爲必須要依照其比較結果而對於龐大的值的 t 與 N 實際進行 t − N 的減法運算的緣故，所以由於反覆進行此種減法運算處理而使得全體的演算時間增大。

第 3 個問題點

此外於上述「算法 2」的步驟 3，中間結果 A · B的資料長係 A、B 的分別的資料長（位元數）的合計之資料長

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（5）

，所以是極爲巨大的數目，必須要新設專門用來暫時存放此值得記憶裝置。進而在「算法2」，雖然說可以省略掉「算法1」之中必要的藉由N來進行的除法運算，但是在步驟3還是必須要進行龐大數字A、B、M、N之間的乘法演算，而以如何將此演算有效率地實現作爲課題的解決之道已由本發明之發明人發現。特別是藉由硬體來實現的場合，保持A、B、N的值的記憶裝置與實行乘法運算的演算器之間的資料往來必定要頻繁地進行，因此必須要將記憶裝置或是資料匯流排的邏輯電路規模抑制在最小的限度一事也是由本發明之發明人發現的。

第4個問題點

進而，還具有如何有效率地利用在「算法2」所計算的剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」，而計算冪函數剩餘演算「$X^Y \bmod N$」的課題。

其中一例爲本發明之發明人適用「算法2」而藉由將「算法1」變形而獲得以下之「算法3」。

「算法3」

input X，Y = $e_n e_{n-1} \cdots e_1$，N　　步驟1

$B=R^2 \bmod N$　　步驟2

$A=X$　　步驟3

$A=A \cdot B \cdot R^{-1} \bmod N$　　步驟4

$B=A$　　步驟5

（請先閱讀背面之注意事項再填寫本頁）

裝　訂　線

經濟部中央標準局員工消費合作社印製

五、發明說明（6）

for i = n-1 to 1 step -1 {　　步驟6

A = $A^2 \cdot R^{-1} \bmod N$　　步驟7

if e_i=1 then $A = A \cdot B \cdot N^{-1} \bmod N$　　步驟8

}　　步驟9

$A=A \cdot R^{-1} \bmod N$　　步驟10

output A　　步驟11

在此「算法3」，在步驟2「$R^2 \bmod N$」的計算係必須要作爲前處理，例如使用泛用微處理器而以軟體進行計算的場合，相當耗費計算時間，而會使「算法3」的全體的計算時間增大的事實也由本發明之發明人發現。此問題點在於「算法1」的步驟5、6也同樣會發生。

第5個問題點

在前述「算法3」的步驟7計算「$A^2 \cdot R^{-1} \bmod N$」時，因爲係使用剩餘乘算「$A = A \cdot B \cdot R^{-1} \bmod N$」來計算的緣故，所以將A的值轉送至B相當費時的事實，也是由本發明之發明人發現的。此外，在前述「算法3」的步驟10，必須將B的值設定爲B=1，也因此同樣相當費時。

第6個問題點

進而於「算法3」的步驟4的「$A = A \cdot B \cdot R^{-1} \bmod N$」，步驟7的「$A = A^2 \cdot R^{-1} \bmod N$」以及步驟8的「A

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（7）

$= A \cdot B \cdot R^{-1} \bmod N$」都必須要將演算結果保存於與輸入值相同的記憶裝置A。此時，剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」通常無法瞬間實行，而必須要花費相當多的時序而參照無數次輸入值A、B、N以便同時進行計算，所以途中的結果以及最終結果並無法與輸入值A保存於相同的記憶裝置，而有必要作爲暫存記憶值（Temp）而保存於其他的記憶裝置（例如暫時暫存器temporary registor）。因此，實質的演算式成爲「$Temp \leftarrow A \cdot B \cdot R^{-1} \bmod N$」，例如爲了實行步驟8，在最後必須要實行「$A \leftarrow Temp$」，而具有計算時間增大的問題。此問題在「算法1」的步驟5、步驟6也同樣發生。

本發明係有鑑於上述之問題點而成之發明，係以高速實現冪函數剩餘演算「$X^Y \bmod N$」爲目的。

特別是使用上述課題專用的硬體設備以及泛用的微處理器來解決，以實現高速的冪函數剩餘演算「$X^Y \bmod N$」爲目的。

此外本發明其他的目的在於實現上述的專用硬體設備時，將其邏輯電路規模限制爲最小。

本發明其他的目的還有將上述專用的硬體搭載於與IC卡用微電腦相同的半導體晶片上，以低成本實現容易使用的適用冪函數剩餘演算「$X^Y \bmod N$」的供密碼化、解碼化用的微電腦。

關於本發明之前述以及其他的目的以及其新穎之特徵，應可藉由本說明書之記載以及附圖而得以明瞭。

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（8）

以下，說明本發明所開示之發明之中較具代表性者之概要。

亦即，冪函數剩餘演算所使用的資料處理裝置（6），係將當n爲代表演算值的位元數之正整數，N爲$0 < N < 2^n$的n位元的正整數，A_{in}爲$0 \leqq A_{in} < 2^n$的n位元的正整數，A_{out}爲$0 \leqq A_{out} < 2^n$的n位元的正整數，k爲0或是正整數時，演算式爲$A_{out}=f(A_{in}) \bmod N + kN$，進行輸入值$A_{in}$與輸出值$A_{out}$的值的範圍在0以上不滿$2^n$的整數的剩餘演算之演算手段（32），以及藉由此演算手段而控制前述剩餘演算之控制手段（31）製作於1個半導體基板上而構成的。

一般式所給定的「$A_{out}=f(A_{in}) \bmod N + kN$」的剩餘演算，可以表爲$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N + kN$與$A_{out}=A_{in} \cdot B \bmod N + kN$的演算式所分別給定的剩餘演算，此時，R爲$2^n$，$R^{-1}$爲滿足$R \cdot R^{-1} \bmod N = 1$的$0 < R^{-1} \leqq N$之n位元的正整數，B係$0 \leqq B \leqq 2^n$之n位元的正整數。

前述演算式$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N + kN$所給定的剩餘演算中可以包含有$t=(A_{in} \cdot B+M \cdot N)/R$所給定的演算處理，及接著給定 if $t \geqq R$ then return t-N else return t，可以包含將t－N或者是t作爲前述A_{out}之演算處理。如第1圖之例，於t與R的比較，藉由n位元的t的溢位(overflow)而檢測出前述之$t \geqq R$。因應溢位的檢測結果而進行t－N的演算即可，與直接將t與N作爲

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（9）

比較對象的場合相比t－N的減法處理的頻率較少，對於剩餘演算速度的高速化有所助益。

如第11圖以及第12圖所示，具備有直列式接續的積和演算器（33、34），實行前述演算處理$t=(A_{in}\cdot B+M\cdot N)/R$可得部分積之和。藉此，不需要部分積之一次記憶用記憶體手段，可以實現演算手段的電路規模的縮小。

前述控制手段使用前述直列是接續的積和演算器可以選擇實行乘數與被乘數爲多倍長的多倍長乘法運算。這可以適用於供冪函數剩餘演算用的「$R^2 \bmod N$」的處理所必要的多倍長乘法演算處理。

前述演算手段，如第5圖所示，可以具備有將前述演算式$A_{out}=A_{in}\cdot B\cdot R^{-1} \bmod N + kN$之中之B值，選擇指定爲$A_{in}$或者是1的手段（16、17）。藉此，「$A_{in}\cdot B\cdot R^{-1} \bmod N + kN$」、「$A_{in}^2\cdot R^{-1} \bmod N + kN$」或者是「$A_{in}\cdot R^{-1} \bmod N + kN$」可以同樣地演算，不需要對於保有值B用的暫存器轉送值A，或者是將保有值B用的暫存器設定爲1的處理，對於剩餘演算處理的高速化有所助益。

微電腦（MCU），如第3圖所示在1片半導體基板上包含有前述資料處理裝置（6），及將此資料處理裝置作爲共同處理器(coprocessor)而利用的微處理器（5）。前述微處理器，係於前述資料處理裝置設定前述演算$A_{out}=A_{in}\cdot B\cdot R^{-1} \bmod N + kN$所必須要有的輸入值，指

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（10）

示該演算處理，而利用其演算結果。藉由此微電腦，剩餘演算式子「$A_{out}=A_{in}\cdot B\cdot R^{-1} \bmod N + kN$」、「$A_{out}=A_{in}^2\cdot R^{-1} \bmod N + kN$」或者是「$A_{out}=A_{in}\cdot R^{-1} \bmod N + kN$」所給定的演算，可以藉由內藏積和演算器的共同處理器來實行，進而於該共同處理器還具備高速實行前處理「$R^2 \bmod N$」用的乘算機能，使用這些共同處理器的演算機能，可以用微電腦高速實行冪函數剩餘演算「$X^Y \bmod N$」。

可以將收容供在前述微處理器上實行冪函數剩餘演算用的動作程式的ＲＯＭ（８）搭載於微電腦的晶片上。此時，前述演算$A_{out}=A_{in}\cdot B\cdot R^{-1} \bmod N + kN$成爲包含於前述冪函數剩餘演算的演算。藉此，可以實現對於高度的隱私性的達成具有助益的微電腦。

進而可以將作爲前述冪函數剩餘演算對象的資料可電氣改寫地保持著的非易失性記憶體（１０）搭載於微電腦晶片。藉此，可以將應該要保持機密的資料的操作最適化地作爲ＩＣ卡用的微電腦之利用。

將演算結果A_{out}作爲下一個演算的輸入值A_{in}，實行演算式$A_{out}=f(A_{in})$所給定之演算的資料處理裝置，如第７圖所示，可以由分別記憶A_{in}與A_{out}的內容之記憶手段（２３、２４），以及對於該記憶手段指示A_{in}與A_{out}的記憶場所的相互交換的旗標手段（２７），以及於前述計算式的演算實行結束後藉由將前述旗標手段的值反轉而取代使A_{in}與A_{out}的值的物理上的交換，而使

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（11）

A_{in}與A_{out}的記憶場所在邏輯上交換的控制手段（28、31）等而構成。藉此，值A_{in}與A_{out}的記憶場所的交換可以在表面上瞬間實行，對於演算時間的縮短有所助益。

利用前述直列配置的積和演算器之處理也可以適用於以下的資料處理裝置。亦即，係將當n爲代表演算值的位元數之正整數，N爲$0 < N < 2^n$的n位元的正整數，A_{in}爲$0 \leqq A_{in} < 2^n$的n位元的正整數，A_{out}爲$0 \leqq A_{out} < 2^n$的n位元的正整數時，演算式爲 $A_{out}=f(A_{in}) \bmod N$，進行輸入值$A_{in}$與輸出值$A_{out}$的值的範圍在0以上不滿$2^n$的整數的剩餘演算之演算手段，以及藉由此演算手段而控制前述剩餘演算之控制手段製作於1個半導體基板上而構成的資料處理裝置。前述演算式爲 $A_{out}=f(A_{in}) \bmod N$之剩餘演算包含了演算式$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N$所給定的剩餘演算，此時，R爲$2^n$，$R^{-1}$爲滿足$R \cdot R^{-1} \bmod N = 1$的$0 < R^{-1} \leqq N$之n位元的正整數，B係$0 \leqq B < 2^n$之n位元的正整數，前述演算手段，備有供實行演算式$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N$所給定的剩餘演算的直列接續之積和演算器。作爲相對於此之具體的演算樣態，前述演算式$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N$所給定的剩餘演算，包含有$t=(A_{in} \cdot B+M \cdot N)/R$所給定的演算處理，及接著給定 if $t \geqq R$ then return $t-N$ else return t，可以包含將$t-N$或者是t作爲前述A_{out}之演算處理，前述演算手段，藉由直列接續的前述積和演算器，實行前

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（12）

述演算處理 $t=(A_{in} \cdot B+M \cdot N)/R$ 作爲部分積之和。藉此也同樣，不需要部分積之一次記憶用記憶體手段，所以可以縮小演算手段的電路規模。

較佳之實施形態

（1） 第1實施形態

在上述「算法2」，步驟4必須每次進行t與N的大小比較，特別是在t與N的值爲相當龐大的值時，因爲實際上必須要實行t－N的減算的緣故，所以具有全體計算時間增大之前述第2個問題點。於第1實施形態之中，爲了解決此問題點，而將「算法2」的步驟4採用t與R的大小比較而不是採用t與N的大小比較。其算法如下。

「算法4」

$N' = -N^{-1} \bmod R$ 步驟1

$M=A \cdot B \cdot N' \bmod R$ 步驟2

$t=(A \cdot B+M \cdot N)/R$ 步驟3

if $t \geqq R$ then return t-N else return t 步驟4

於上述「算法4」，A、B、N的位元數爲n位元，$R=2^n$（第n位元爲止的全位元的邏輯值爲0，第n+1位元的邏輯值爲1的2進位數）。此算法4，係實現後述之演算式「$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N+kN$」之演算法，可以將其處理視爲冪函數剩餘演算用的一個副程式。

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部中央標準局員工消費合作社印製

五、發明說明（13）

於算法4，如果 t≧R，則將 t－N 當作 A out，如果 t＜R，則將 t 作爲 A out。

上述演算法4的步驟4的大小比較以及減法運算，係將步驟3的「t=(A・B+M・N)/R」演算結果 t 是否在指定的值 2^n（=R）以上，藉由 t 的溢位而檢測出，檢測出溢位時，將步驟3的 t 補正爲 t－N。因而可以稱此特徵爲「溢位補正」。實現此特徵之電路之一例，係第1圖所示之溢位補正電路。於第2圖顯示作爲溢位補正電路的比較例而實現前述算法2的步驟4的電路的方塊圖。

首先，於第2圖，t－N的減法運算係在減算器3A被實行，其演算結果係以是否發生借位（borrow）而以選擇器4A選擇 t 或者是 t－N 作爲減法運算的輸出結果。亦即在此構成 t－N 的減法運算的實行是絕對必要的。

相對於此，在第1圖的構成中，t 的值是否超過指定的數 n 所給定的位元數，係以溢位位元 OV 是0或是1而來判斷。此相當於上述的 t 與 R 的大小比較。亦即，因爲如上述 R= 2n 的緣故，如果前述溢位位元 OV 爲1的話則 t≧R，如果溢位位元 OV 爲0的話則爲 t＜R。

此處僅在 OV= 1 而發生溢位時，以減算器3實行 t－N 的減算，以選擇器3選擇其減法運算的結果而作爲輸出結果。在 OV= 0 而沒有發生溢位時，減算器3不進行減法運算，將 t 直接以選擇器4選擇而作爲輸出結果。溢位位元 OV，也對控制演算的控制電路31供給。控制電路31因應 OV= 1 或是 OV= 0，而控制例如以減算器

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（14）

3進行減法運算與否。進行減法運算的場合，在結束該減法運算之後將選擇器4的輸出控制掛鉤(latch)於後段。

亦即，在第1圖的構成與第2圖的構成相比，可以降低實行減法運算的頻率，可以縮短平均演算時間。

此處，前述「算法4」所給定的最終演算結果亦有t較N爲大的場合。亦即，因爲R（$=2^n$）＞N，所以經由步驟3的處理以前述溢位補正所得到的t的值較R爲小而且較N爲大的話，該t的值成爲以「算法4」所給定的演算結果t。因爲在前述「算法2」的場合是直接與N進行比較的緣故，所以最終所得到的演算結果t必定較N爲小。因此，「算法4」全體的演算式以「$A \cdot B \cdot R^{-1} \bmod N + kN$」表示，有必要藉由補正項kN來解決前述誤差。此處k爲0或者是正整數。

前述剩餘演算式「$A \cdot B \cdot R^{-1} \bmod N + kN$」，若將其值A區分爲輸入值$A_{in}$與輸出值$A_{out}$而表示的話，演算式成爲「$A_{out} = A_{in} \cdot B \cdot R^{-1} \bmod N + kN$」。將此簡化爲一般式的話，演算式可以寫成「$A_{out} = f(A_{in}) \bmod N + kN$」，輸入值$A_{in}$與輸出值$A_{out}$的值的範圍在N以上不滿$2^n$，n係代表演算值的位元數的正整數，N係$0 < N < 2^n$的n位元的正整數，$A_{in}$係$0 \leq A_{in} < 2^n$的n位元的正整數，$A_{out}$係$0 \leq A_{out} < 2^n$的n位元的正整數，k係0或者是正整數。前述R爲$2^n$，$R^{-1}$係滿足$R \cdot R^{-1} \bmod N = 1$的$0 < R^{-1} \leq N$之n位元的正整數，B係$0 \leq B \leq 2^n$之n位元的正整數。

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（15）

使用前述「算法4」的場合，伴隨著其補正項而實行冪函數剩餘演算「$X^Y \bmod N$」而修正「算法3」的話，可得以下所示之「算法5」

「算法5」

input X，$Y = e_n e_{n-1} \cdots e_1$，N，R　　步驟1

$B = R^2 \bmod N$　　步驟2

$A = X$　　步驟3

$A = A \cdot B \cdot R^{-1} \bmod N + kN$　　步驟4

$B = A$　　步驟5

for i = n-1 to 1 step -1 {　　步驟6

　　$A = A^2 \cdot R^{-1} \bmod N + kN$　　步驟7

　　if ei=1 then $A = A \cdot B \cdot R^{-1} \bmod N + kN$　　步驟8

}　　步驟9

$A=A \cdot R^{-1} \bmod N + kN$　　步驟10

$A=A \bmod N$　　步驟11

output A　　步驟12

前述「算法3」到前述「算法5」的變更點在於分別變更了步驟4、7、8、10的剩餘演算的演算式之外，還追加了步驟11。步驟11係爲了要使演算結果較N爲小而追加的。前述「算法5」的演算也可以藉由泛用微處理器而實行。但在本實施例中，採用專用的硬體設備之共同處理器。

（請先閱讀背面之注意事項再填寫本頁）

裝　訂　線

經濟部中央標準局員工消費合作社印製

五、發明說明（16）

於第3圖，5係微處理器，6係共同處理器，7係時序發生器，8係容納著微處理器5應該要實行之程式或是固定資料的ＲＯＭ（讀取專用記憶體，Read Only Memory），9係提供微處理器5的作業領域之ＲＡＭ（可隨機讀寫之記憶體，Random Access Memory），１０係適用前述冪函數剩餘演算「ＸＹmod Ｎ」將被符號化的資訊等的保存所利用到的可以用電氣訊號改寫之ＥＥＰＲＯＭ或是快閃記憶體等的非易失性記憶體（簡稱ＥＥＰＲＯＭ），１１係輸出入埠（略稱爲Ｉ／Ｏ埠）。微處理器5、共同處理器6、ＲＡＭ8、ＲＯＭ9、ＥＥＰＲＯＭ１０以及Ｉ／Ｏ埠１１共通接續於代表之匯流排１２、１３。匯流排１２係位址匯流排與控制匯流排的總稱。１３意味著資料匯流排。時序發生器7係基於由時序端子ＣＬＫ而來所供給的時序訊號產生內部的動作基準時序訊號而供應至微處理器5以及共同處理器6。Ｉ／Ｏ埠１１係結合於資料輸出入外部端子Ｉ／Ｏ。Ｖcc，Ｖss係微電腦ＭＣＵ的電源用外部端子，ＲＥＳ係微電腦ＭＣＵ的再開機用外部端子。

微電腦ＭＣＵ，並沒有特別的限制，可以如第3圖所示全部的機能方塊被形成於例如單結晶矽基板般的一個半導體基板。

前述共同處理器6係作爲實行前述「算法5」的步驟4、步驟7、步驟8、步驟１０所示之「$A = A \cdot B \cdot R^{-1} \bmod N + kN$」等的「剩餘演算」之專用硬體，含有演算電

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（17）

路３２與控制電路３１。前述第１圖的溢位補正電路係包含於共同處理器６內。剩餘乘算的輸入值Ａ、Ｂ、Ｒ、Ｎ以及輸出值Ａ係保持於共同處理器６之中的暫存器或者是ＲＡＭ等記憶裝置。或者是也可以將其保持於前述ＲＡＭ９亦可。

微處理器５，係可以實行周知之演算命令或者是條件分歧命令等的泛用微處理器，使用共同處理器６實現演算機能的同時，藉由「算法５」而實現冪函數剩餘演算「$X^Y \bmod N$」。

於第４圖概略顯示藉由「算法５」而實行冪函數剩餘演算「$X^Y \bmod N$」時藉由微處理器５與共同處理器６而進行處理之流程圖。於第４圖Ｔ１以及Ｔ５係於「算法５」中微處理器５應該要負擔的冪函數演算處理，Ｃ２係「算法５」之中共同處理器６應該要負擔的冪函數演算處理。Ｃ２所進行的演算處理雖未特別加以限制，但爲「算法５」的步驟４、７、８、１０的處理。微處理器５在當共同處理器６實行演算時，將其所必須要用到的值（Ａ、Ｂ、Ｎ等）設定於例如共同處理器６的內部的暫存器（Ｔ２）。其後微處理器５對於共同處理器６藉由指令等指示開始進行演算（Ｔ３）。共同處理器６使用藉由Ｃ１而設定的輸入值，藉由指令等進行被指示之處理（Ｃ２）。其間，微處理器５等待共同處理器６的演算結束，或者是可以進行其他的處理（Ｔ４）。共同處理器６在Ｃ２結束演算時，通知微處理器（Ｃ３）。微處理器５使用共同處理器

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（18）

6所計算之結果而進行演算法的處理（T5）。進而還剩下共同處理器6應該要負擔的冪函數處理的場合，經由T6反覆進行上述處理。

第2實施形態

於上述「算法5」，步驟7所示之剩餘乘算「A2・R^{-1} mod N + kN」可以藉由將步驟8所示之剩餘演算「A・B・R^{-1} mod N + kN」而將B = A即可實現。然而，實際上勢將分配於A的暫存器的值向分配給B的暫存器傳送的緣故，所以會產生演算的實行時間增大的問題。此外步驟8 for的迴圈之中反覆執行的場合，因爲可以使用同一個B值反覆使用的緣故，在步驟7將B的值以A的值來改寫可以實行較有效率的演算實行。進而，在步驟10所示的剩餘乘算「A・R^{-1} mod N + kN」也是同樣可以藉由以B = 1來實現，但是將1設於B會產生使演算的實行時間增大的問題。這些是對應前述第5個問題點的內容。

爲了解決此問題點，於實現剩餘乘算「A・B・R^{-1} mod N + kN」的電路上，採用將B的值代換爲可以選擇是A的值或者是“1”的構成。

於第5圖顯示解決該課題之演算手段的方塊圖。並沒有特別的限制，但是第5圖所示之演算手段，可以作爲實現第3圖的共同處理器6的一個電路構成，而包含在例如單結晶矽基板一般的一個半導體晶片上所形成的微電腦MCU內。

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（19）

於第５圖中，１５係演算器，１６係選擇器，１７係控制暫存器，１８～２０係演算用暫存器。選擇器１６依控制暫存器１７的控制位元ＣＢ０、ＣＢ１的值而選擇將暫存器１８、１９或是固定值“１”供給至演算器１５。控制暫存器１７的２個位元ＣＢ０、ＣＢ１因應其邏輯值的組合而指定演算式，亦即，因應其設定值，而將剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N + kN$」之中的Ｂ值，介由選擇器１６而選擇暫存器１８的值Ａ、暫存器１９的值Ｂ或者是固定值“１”之任一而對演算器１５供給。因應控制暫存器１７的位元ＣＢ０、ＣＢ１而進行選擇器的選擇動作，演算器１５可以將「$A \cdot B \cdot R^{-1} \bmod N + kN$」、「$A^2 \cdot R^{-1} \bmod N + kN$」或者是「$A \cdot R^{-1} \bmod N + kN$」可以同樣地演算。亦即，不需要對於保有值Ｂ用的暫存器轉送值Ａ，或者是將保有值Ｂ用的暫存器設定爲１的處理。對於控制暫存器１５的值的設定，係由例如第３圖所視之微處理器５而經由匯流排１２、１３而進行。

於第６圖顯示與第５圖之比較例之演算手段的方塊圖。在第６圖的場合，實行剩餘乘算「$A^2 \cdot R^{-1} \bmod N + kN$」的場合，必須要經由匯流排而進行將Ａ的值轉送至Ｂ，或者是要將Ｂ的值設定爲１的處理。

（３）第３實施形態

於上述「算法５」，在步驟８所代表的剩餘演算「$A(A_{out}) = A(A_{in}) \cdot B \cdot R^{-1} \bmod N + kN$」，必須要

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部中央標準局員工消費合作社印製

五、發明說明（20）

將演算結果（Aout）保存於與輸入值A（Ain）同樣的記憶裝置或者是暫存器。剩餘乘算通常無法瞬間實行，而必須耗費很多的時序(clock)（演算步驟數）而且要多次參照輸入值A、B、N而進行計算的緣故，無法將計算的中間結果以及最終結果保存於與輸入值A相同的記憶裝置，而產生保存於一次記憶值（Temp）用的其他的一次記憶裝置的必要性。因此，演算式「$A = A \cdot B \cdot R^{-1} \bmod N + kN$」成爲「$Temp \leftarrow A \cdot B \cdot R^{-1} \bmod N + kN$」，爲了實行步驟8，在最後必須要實行「A←Temp」，而具有計算時間增大的問題已由本發明之發明人發現這對應著前述第6個問題點。此問題在「算法5」的步驟4、7、10也同樣發生。

於第7圖顯示著眼於解決上述問題點之演算手段之方塊圖。雖然沒有特別的限制，第7圖所示之演算手段可以當成是實現第3圖的共同處理器6的一個電路構成，包含在形成於例如單結晶矽基板般的一個半導體晶片上而形成的微電腦MCU之內。

於第7圖，22係演算器，23與24係容納前述值A的暫存器與一次記憶體Temp用的暫時暫存器交互切換而被利用的暫存器，25係供容納B值所被利用的暫存器，26係容納N值所被利用的暫存器。暫存器23～26與演算器22係共通接續於共同演算器內部的匯流排40。

於第7圖S1係暫存器23的選擇訊號，S2係暫存器24的選擇訊號。Rtmp係對於一次記憶體Temp的

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（21）

操作訊號，Ra係對於A值的操作訊號。操作訊號Rtmp，Ra係微處理器因應對於共同處理器供給的指令而由控制電路31輸出的演算控制訊號的一種。

29、30係選擇器。選擇器29、30依照旗標27的值而選擇操作訊號Ra或者是Rtmp，將選擇的訊號作爲暫存器選擇訊號S1、S2而輸出。旗標27爲「1」時，選擇器29選擇操作訊號Ra作爲暫存器選擇訊號S1，選擇器30選擇操作訊號Rtmp作爲暫存器選擇訊號S2。旗標27爲「0」時，選擇器29、30所作的選擇狀態相反。

前述旗標27的設定係藉由旗標反轉控制邏輯28而進行的。此旗標反轉控制邏輯28，係例如包含於前述控制電路31，控制演算的程序的編續器在顯示剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」的結束時間時，與其同步將旗標27的狀態控制爲反轉。

如此，前述暫存器23與24每次進行剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」時交互被分配爲收容值A或者是一次記憶值Temp的用途。例如，暫存器23在被使用爲容納值A時，暫存器24被使用於一次記憶體Temp的收容。在剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」結束實行時，控制其爲可以切換對應於操作訊號Ra，Rtmp之選擇訊號S1、S2的對應。此暫存器機能的交換，如圖所示，係藉由將1位元的旗標27在剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」結束執行時將其反轉而被控制。藉此，值A、Temp的內容

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（22）

的交換可以在表面上瞬間被執行，對於演算時間的縮短有所幫助。

第8圖係顯示對於第7圖的演算手段的比較例。於第8圖所示之演算手段，剩餘乘算「A· B· R^{-1} mod N」的實行成果，首先被收容於被固定分配用來供暫時記憶值Temp的暫時暫存器，該暫時暫存器的值Temp，被轉送至值A專用的暫存器，而值A的暫存器保持著最終結果。如此的資料轉送會花上很多時間。

於第9圖顯示將第7圖的暫存器23～26以RAM32實現的場合的共同處理器的例。

雖未有特別的限制，在第9圖所示之共同處理器，可以是適用於作爲第3圖的微電腦MCU的共同處理器6的一個電路構成，與前述相同，包含於例如單結晶矽基板般的一個半導體晶片上所形成的微電腦MCU之內。

於第9圖50係RAM，在該圖中顯示代表的記憶領域23M～26M。記憶領域23M、24M被利用於交互切換容納前述值A的領域與容納演算中途結果之一次記憶值Temp的領域。記憶值25M被利用於容納值B用，記憶體26M被利用爲容納值N用。各記憶領域23M～26M介由演算電路內部的匯流排40，接續於被代表顯示的演算器22。對於RAM50的操作控制係介由匯流排52、53來進行的。位址訊號係藉由Amsb～A0的指定位元的位址訊號而進行RAM50的定位址（addressing）。

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（23）

於控制暫存器51設定了由微處理器5所供給的指令，由控制電路31將其解讀，而對演算器22等產生控制訊號。

對於前述記憶領域23M～26M的選擇資訊，雖未加以特別的限制，係指定爲位址訊號的最上位側的2個位元A_{msb}、A_{msb-1}。此處，A_{msb}= 1時，因應旗標27的邏輯值而爲了使A_{msb-1}的邏輯值強制反轉的緣故而設有和閘(AND gate)54以及排他邏輯和閘（NOR gate）55，該閘55的輸出A'_{msb-1}取代A_{msb-1}而被供給至RAM50。A_{msb}=1、A'_{msb-1}=0時選擇記憶領域23M，A_{msb}=1、A'_{msb-1}=1時選擇記憶領域24M，A_{msb}=0、A'_{msb-1}=0時選擇記憶領域25M，A_{msb}=0、A'_{msb-1}=1時選擇記憶領域26M。在共同處理器的動作程式上，A_{msb}=1、A_{msb-1}=0指的是數值A的收容領域，A_{msb}=1、A_{msb-1}=1指的是temp的收容領域。

前述旗標27的設定係藉由旗標反轉控制邏輯電路28而進行的。此旗標反轉邏輯電路28係例如在況至演算的程序的控制電路31內部的編序器在顯示剩餘乘算「$A \cdot B \cdot R^{-1} \bmod N$」的結束時間時，與其同步反轉控制旗標27的狀態。

藉此，前述記憶領域23M、24M分別排他的容納值A以及一次記憶值Temp。例如，旗標27爲0時，記憶領域23M使用於容納值A，記憶領域24M使用於容納Temp。在剩餘乘算實行結束時旗標27的值被反

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（24）

轉爲１時，於接著的演算中，記憶領域２３Ｍ用來容納Ｔｅｍｐ，而記憶領域２４Ｍ被使用來容納值Ａ。藉此，在取代暫存器而採用ＲＡＭ５０的場合也是可以在表面上可以瞬間實現值Ａ與Ｔｅｍｐ的內容的互換。

（４）第４實施形態

於上述的「算法５」，爲了實行步驟８所代表的實行剩餘乘算「$A = A \cdot B \cdot R^{-1} \bmod N + kN$」用的演算法，係以前述「算法４」所記載的方式計算，但是此演算法有必要考慮以下數點。

前述「算法５」的步驟１、步驟２必須要事先準備Ｎ'、Ｍ。此外採用極大數值的Ｍ必須持續保持到演算的時刻。此外，前述「算法５」的步驟３中，中間結果ＡＢ因爲成爲值Ａ、Ｂ的資料長度的合計之資料長度的極爲龐大的數字的緣故，所以有必要具有因應這些計算之暫時保存各值之記憶裝置。這些對應於前述之第１個問題點。

此外，在前述「算法５」的步驟３中有必要進行Ａ、Ｂ、Ｍ、Ｎ之間的龐大數目之間的乘算，而具有如何有效率的實現的課題。特別是藉由硬體設備來實現的場合，必須要頻繁地進行保持Ａ、Ｂ、Ｎ的值的記憶裝置與實行乘法運算的演算器之間進行資料的交換，有必要考慮記憶裝置或是資料匯流排得邏輯電路規模被抑制於最小限度。此係對應於前述第３個問題點。

此處，作爲解決這些問題的手段，首先，說明關於如

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（25）

何有效率地進行前述A、B、M、N的極爲龐大的值之間的乘法運算的原理。

於第10圖顯示值B與M的區塊分割之例。於第10圖中，顯示將n位元長的B、M分別分割爲L位元長的區塊的狀態。藉此，B、M係分別被分割爲n／L個區塊。例如n係512位元，L係32位元。

使用此區塊分割，於「算法4」的步驟3所記載的演算式「(A・B+M・N)/R」，施以分解爲部分積的變形的話，成爲如下形態。下列式子所顯示的數式變形結果省略了部分積的項的形式在可以把握的範圍的記載。

$$(A \cdot B+M \cdot N)/R = (A \cdot (2^{n-L} \cdot B_{n/L-1}+\cdots+2^{L} \cdot B_1+B_0)$$
$$+(2^{n-L} \cdot M_{n/L-1}+\cdots+2^{L} \cdot M_1+M_0) \cdot N)/R$$
$$=(\cdots((A \cdot B_0+M_0 \cdot N)/2^{L}+(A \cdot B_1+M_1 \cdot N)/2^{L}+\cdots+$$
$$(A \cdot B_{n/L-1}+M_{n/L-1} \cdot N))/2^{L}$$

由此便形式可知，演算「$(A \cdot B_i+M_i \cdot N)/2^{L}$」反覆進行n／L次的話可以實現「(A・B+M・N)/R」。

於第11圖顯示著眼於藉由前述部分積實行演算「$(A \cdot B_i+M_i \cdot N)/2^{L}$」用的演算手段之共同處理器之其他例。雖未加以特別的限制，第11圖所示之共同處理器，可以作爲適用於第3圖的共同處理器6的一個電路構成，與前述相同，被包含於例如單結晶矽基板般的一個半導體晶片上而被形成的微電腦MCU之內。

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（26）

於第１１圖，３３係第１積和演算器，３４係第２積和演算器，３５係保持一次記憶值Ｔｅｍｐ用的暫時暫存器，３６係保持值Ａ所利用的暫存器，３７係保持收容值Ｂ用的暫存器，３８係收容值Ｎ所利用的暫存器，３９係Ｍｉ產生邏輯電路，４０係供保存以Ｍｉ產生邏輯電路所產生的值Ｍｉ用的掛鉤（latch），４１係進行「÷２^{L}」用的移位電路。

第１１圖所示之電路係基於第１０圖所示之樣態之區塊分割而實行「$(A \cdot B_i+M_i \cdot N)/2^{L}$」演算的電路。首先，第１積和演算器３３將暫存器３５的值Ｔｅｍｐ，暫存器３６的值Ａ，暫存器３７的值ＢＩ作爲輸入值，實行積和演算「$Temp+A \cdot B_i$」。其演算結果值作爲Temp2而被送至下一段的第２積和演算器３４。值Temp2係ｎ＋Ｌ位元長的整數。另一方面，Ｍｉ產生邏輯電路３９，將Ｌ位元長的數A_o, B_i, N_o作爲輸入值而產生Ｌ位元的整數Ｍｉ，此正數Ｍｉ被暫時保存於暫存器４０。第２積和演算器３４將前述Temp2, N, Mi作爲輸入值，實行積和演算「$Temp2+M_i \cdot N$」。ｎ＋Ｌ位元長的演算結果的下位Ｌ位元全部是０，將此藉由移位暫存器而將其消除（亦即以２^{L}除之），ｎ位元長的結果作爲值Temp而被送往暫存器３５保存。

反覆實行以上的動作ｎ／Ｌ次的話，可以實現演算「$(A \cdot B_i+M_i \cdot N)/2^{L}$」。藉此，不必預先將ｎ位元的整數Ｍ預先計算而保持之，僅將Ｌ位元長的Ｍｉ於積和演算器３３的計算中求之並保存於暫存器４０即可。可以消除Ｍ值的

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（27）

計算時間以及縮小保存M值得記憶手段的規模。進而，藉由將積和演算器３３與積和演算器３４直列連接而連續動作，可以不需要特別設置用來暫時保存ｎ＋Ｌ位元長的中間結果Temp2的記憶手段。

於第１２圖顯示著眼於藉由前述部分積而實行「（A・B_i+M_i・N）/2^L」演算用的演算部之共同處理器之其他例。雖未加以特別的限制，在第１２圖所示之共同處理器，可以當成是實現第３圖的共同處理器６的一個電路構成，包含在形成於例如單結晶矽基板般的一個半導體晶片上而形成的微電腦ＭＣＵ之內。

於第１２圖之例，對於第１１圖的共同處理器而言，在暫存器３５～３８的積和演算器３３、３４上以匯流排４０接續是不同點。亦即可以藉由ＲＡＭ４２構成前述暫存器３５～３８。藉此，半導體晶片上的暫存器的面積可以降低。此外，於此構成，因爲特別是藉由匯流排４０轉送的資料量相當多，所以具有匯流排的寬度要變大而且半導體晶片的面積又不能夠增大的必要性。如第１１圖之例所示般的藉由將積和演算器３３與積和演算器３４直列連接，可以不需要將中間結果Temp2藉由匯流排轉送，所以可以降低藉由匯流排得資料轉送量。

（５）第５實施形態

於第１１圖或是第１２圖之例，藉由在第１積和演算器３３使Temp=0，在第２積和演算器３４使M_i・N＝０，

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（28）

進而藉由選擇器４１而進行「÷２ᴸ」的動作，可以將該圖所顯示的演算手段作爲例如「A・Bi」般的多倍長度乘法演算（較小的數Bi與相當於其多倍長度的較大的數A之相乘運算）的電路來使用。這是解決前述第４個問題點用的一個手段。如「A・ Bi」般的多倍長度乘法演算，藉由使用例如「算法５」的步驟２的演算「R² mod N」以微處理器實行而適用，可以使該演算高速化。

亦即，於第１３圖顯示「R² mod N」的計算的概念圖。於第１３圖，R＝２ⁿ，n＝５１２，N爲５１２位元，R²僅有最上位的位元爲１，下位側的１０２４位元全部爲０。以微處理器進行「R² mod N」的演算時，將極大的數R²以同樣大的數N直接進行除法運算，並非有效率的方法，所以將被除數由最上位取６４位元單位的區塊，此外，將除數由最上位側取３２位元單位的區塊，依序將上位側的區塊相除進行運算，藉此所得之值取作商的概略數。於第１３圖顯示例如以Q（＝Da÷Na）取作商的概略數。概略而言，對於R２的上位側進行「Q・Na」的減算。對於「Q・Na」減算的結果進行同樣的處理。進而藉由對於其結果反覆進行同樣處理之手法，可以得到「R² mod N」的結果。實際上在其途中，爲了將剩餘位元除去而中介著有減法運算處理。此時，根據第１３圖之例，前述「Q・Nb」演算的處理，在第１次係３２位元與４８０位元的龐大數目的乘法運算處理。而且如此龐大數目的乘法運算處理要反覆很多次。此時，藉由第１１

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（29）

圖或是第12圖所示之共同處理器而利用可以運算的如前述「A·Bi」的多倍長度乘法運算，換言之，使共同處理器負擔如此多倍長度演算的化，在使用微處理器實行「算法5」之步驟2的演算「R^2 mod N」時，可以提高演算處理的速度。

以上說明了由本發明之發明人所完成之發明，但本發明並不以上述範圍爲限，在不超過其要旨的範圍內當然可以進行種種的變更。基於第1、5、7、9、11、12圖所說明的內容，係爲了使容易理解各課題所分別對應之解決手段，而說明共同處理器的個別的電路構成，但是也當然可以將這些構成的全部或者是選擇其中部份構成於一個共同處理器內而實現之。

此外，值A、B、N的位元數也不限定於512位元，當然也可以用位元數更多的數。此外，爲了要實現冪函數剩餘演算而用之硬體設備之構成並不限定於上述各實施形態所列舉的而是可以加以適當的變更。

此外，上述說明之演算技術，雖未特別加以圖示，但是在不採用前述算法2的地方依照算法1而進行冪函數剩餘演算的共同處理器或是微電腦等資料處理裝置也可以適用。

進而，在上述的說明係以使用補正項kN的場合爲一範例，但是利用前述直列配置的積和演算器33、34的處理，基於前述算法3而進行冪函數剩餘演算的以下的資料處理裝置也可以適用。亦即，n係代表演算值的位元數的

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（30）

正整數，N係0＜N＜2^n的n位元的正整數，A_{in}係0≦A_{in}＜2^n的n位元的正整數，A_{out}係0≦A_{out}＜2^n的n位元的正整數時，演算式是以$A_{out}=f(A_{in})$ mod N給定的，進行輸入值A_{in}與輸出值A_{out}的值的範圍在N以上不滿2^n的整數的剩餘演算之演算手段，以及藉由此演算手段而控制前述剩餘演算之控制手段製作於1個半導體基板上而構成的。前述$A_{out}=f(A_{in})$ mod N所給定的剩餘演算，包含$A_{out}=A_{in}\cdot B\cdot R^{-1}$ mod N之類的演算式所給定的剩餘演算，前述R爲2^n，R^{-1}係滿足$R\cdot R^{-1}$ mod N =1的0＜R^{-1}≦N之n位元的正整數，B係0≦B≦2^n之n位元的正整數。前述演算手段，爲了實行$A_{out}=A_{in}\cdot B\cdot R^{-1}$ mod N所給定的剩餘演算，而具備有直列接續的積和演算器33、34。對此具體的演算樣態，前述演算式$A_{out}=A_{in}\cdot B\cdot R^{-1}$ mod N所給定的剩餘演算，包含有$t=(A_{in}\cdot B+M\cdot N)/R$所給定的演算處理，及接著給定 if t≧R then return t- N else returnt，包含將t－N或者是t作爲前述A_{out}之演算處理。前述演算手段，藉由被直列接續的前述積和演算器33、34，而作爲部份積的和，實行前述演算處理$t=(A_{in}\cdot B+M\cdot N)/R$。藉此也同樣的可以不需要部份積的一次記憶用記憶體手段，所以可以縮小演算手段的電路規模。

本發明除了共同處理器或是微電腦之外，還可以適用於具有共同處理器或是微電腦之ＩＣ卡、ＩＣ卡、密碼化

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（31）

·解密化裝置等廣泛用途。

藉由本發明所開示之發明之中具有代表性的發明所得到的效果，簡單說明如下。

亦即，可以高速實現冪函數剩餘演算「$X^Y \bmod N$」。

此外，於實現冪函數剩餘演算用的上述專用硬體時，可以使其邏輯電路的規模縮小。

進而，將上述專用硬體搭載應用於與ＩＣ卡用的微電腦相同的半導體晶片，可以容易實現低成本的適用於冪函數剩餘演算「$X^Y \bmod N$」的密碼化·解碼化的微電腦。

而，於搭載微處理器的微電腦，剩餘演算以所給定的 $A = A \cdot B \cdot R^{-1} \bmod N + kN$，$A = A^2 \cdot R^{-1} \bmod N + kN$ 以及 $A = A \cdot B \cdot R^{-1} \bmod N + kN$ 演算，以內藏積和演算器的共同處理器來實行，進而於同一共同處理器具備有高速實行前處理 $R^2 \bmod N$ 用的乘法運算機能，使用這些共同處理器的演算機能，可以微處理器高速實行冪函數剩餘演算 $X^Y \bmod N$。

圖面之簡單說明

第１圖係溢位補正電路之一例之方塊圖。

第２圖係相對於溢位補正電路之比較例之方塊圖。

第３圖係可以實行之進行冪函數剩餘演算「$X^Y \bmod N$」的算法５的微電腦之一例之方塊圖。

第４圖係第３圖之微電腦中依照算法５進行冪函數剩

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

五、發明說明（32）

餘演算「$X^{Y} \bmod N$」時的顯示微處理器與共同處理器的概略的處理流程的流程圖。

第5圖係顯示可以選擇實行複數種類的剩餘演算「$A \cdot B \cdot R^{-1} \bmod N + kN$」、「$A^2 \cdot R^{-1} \bmod N + kN$」或者是「$A \cdot R^{-1} \bmod N + kN$」的共同處理器的演算手段之方塊圖。

第6圖係顯示對於第5圖的演算手段之比較例之方塊圖。

第7圖係顯示實現於$A = A \cdot B \cdot R^{-1} \bmod N + kN$等的演算時輸入暫存器與暫時暫存器之暫存器機能交換的演算手段之一例之方塊圖。

第8圖係顯示對於第7圖之演算手段之比較例之方塊圖。

第9圖係將第7圖的暫存器以RAM置換而實現暫存器機能交換的演算手段之一例之方塊圖。

第10圖係顯示將供部份積用之值B、M區塊分割之例之說明圖。

第11圖係含有供藉由部分積而實行$(A \cdot B_i + M_i \cdot N)/2^L$用的演算手段的共同處理器的方塊圖。

第12圖係含有供藉由部分積而實行$(A \cdot B_i + M_i \cdot N)/2^L$用的其他的演算手段的共同處理器的方塊圖。

第13圖係顯示$R^2 \bmod N$的計算概念的說明圖。

第14圖係適用冪函數剩餘演算之密碼化、解碼化裝置的概略說明圖。

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部中央標準局員工消費合作社印製

五、發明說明（33）

圖號說明

3 演算器

4 選擇器

MCU 微電腦

5 微處理器

6 共同處理器

15 演算器

16 選擇器

17 控制暫存器

CB、CB2 控制位元

18～20 暫存器

22 演算器

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部中央標準局員工消費合作社印製

四、中文發明摘要（發明之名稱：　　）

資料處理裝置以及微電腦

於搭載微處理器之微電腦，具有將剩餘乘算數式 $A = A \cdot B \cdot R^{-1} \bmod N + kN$、$A = A^2 \cdot R^{-1} \bmod N + kN$、$A = A \cdot R^{-1} \bmod N + kN$所給定的演算，以內藏積和演算器的共同處理器來實行的機能，進而於同一共同處理器具有供高速實行前處理 $R^2 \bmod N$ 用的乘算機能，使用這些共同處理器的演算機能，以微處理器高速實行冪函數剩餘演算 $X^Y \bmod N$。

英文發明摘要（發明之名稱：　　）

（請先閱讀背面之注意事項再填寫本頁各欄）

裝 訂 線

經濟部中央標準局員工消費合作社印製

六、申請專利範圍

1．一種資料處理裝置，係形成於1個半導體基板上的資料處理裝置，係包含有：

將當n爲代表演算值的位元數之正整數，N爲$0 < N < 2^n$的n位元的正整數，A_{in}爲$0 \leqq A_{in} < 2^n$的n位元的正整數，A_{out}爲$0 \leqq A_{out} < 2^n$的n位元的正整數，k爲0或是正整數時，演算式爲$A_{out}=f(A_{in}) \bmod N + kN$，進行輸入值$A_{in}$與輸出值$A_{out}$的值的範圍在0以上，不滿$2^n$的整數的剩餘演算之演算手段，及

藉由此演算手段而控制前述剩餘演算之控制手段而構成的。

2．如申請專利範圍第1項之資料處理裝置，其中，前述$A_{out}=f(A_{in}) \bmod N + kN$所給定的剩餘演算，包含$A_{out}= A_{in} \cdot B \cdot R^{-1} \bmod N + kN$以及$A_{out}= A_{in} \cdot B \bmod N + kN$之演算式所給定的剩餘演算，前述R爲$2^n$，$R^{-1}$係滿足$R \cdot R^{-1} \bmod N =1$的$0 < R^{-1} \leqq N$之n位元的正整數，B係$0 \leqq B < 2^n$之n位元的正整數。

3．如申請專利範圍第2項之資料處理裝置，其中，$A_{out}= A_{in} \cdot B \cdot R^{-1} \bmod N + kN$所給定的剩餘演算，包含有$t=(A_{in} \cdot B+M \cdot N)/R$所給定的演算處理，及接著給定 if t≧R then return t-N else return t，將t－N或者是t作爲前述Aout之演算處理；前述演算手段係藉由n位元的t的溢位(overflow)而檢測出前述之t≧R。

4．如申請專利範圍第2項之資料處理裝置，其中，

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部中央標準局員工消費合作社印製

六、申請專利範圍

前述演算手段係具有實行演算式 $A_{out} = A_{in} \cdot B \cdot R^{-1} \mod N + kN$ 所給定的剩餘演算用的，直列接續的複數的積和演算器。

5．如申請專利範圍第2項之資料處理裝置，其中，$A_{out} = A_{in} \cdot B \cdot R^{-1} \mod N + kN$ 所給定的剩餘演算，包含有 $t=(A_{in} \cdot B+M \cdot N)/R$ 所給定的演算處理，及接著給定 if $t \geqq R$ then return t-N else return t，將 t－N 或者是 t 作爲前述 A_{out} 之演算處理；前述演算手段具有直列接續的複數的積和演算器，作爲部份積之和而實行前述演算處理 $t=(A_{in} \cdot B+M \cdot N)/R$。

6．如申請專利然第5項之資料處理裝置，其中，前述控制手段係使用前述直列接續之積和演算器而選擇地實行乘數與被乘數係多倍長度之乘法運算。

7．如申請專利然第4項之資料處理裝置，其中，前述演算手段，具有將前述演算式 $A_{out} = A_{in} \cdot B \cdot R^{-1} \mod N + kN$ 之中的 B 的值，選擇性的指定爲 A_{in} 或是1的手段。

8．如申請專利範圍第2項之資料處理裝置，其中，$A_{out} = A_{in} \cdot B \cdot R^{-1} \mod N + kN$ 所給定的剩餘演算，包含有 $t=(A_{in} \cdot B+M \cdot N)/R$ 所給定的演算處理，及接著給定 if $t \geqq R$ then return t-N else return t，將 t－N 或者是 t 作爲前述 A_{out} 之演算處理；前述演算手段係藉由 n 位元的 t 的溢位(overflow)而檢測出前述之 $t \geqq R$，此外，具備有直列接續的複數的積和演算器而作爲部份

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

六、申請專利範圍

積之和而實行前述演算處理t=(A_{in} · B+M · N)/R，此外，具有將前述演算式$A_{out}= A_{in} \cdot B \cdot R^{-1} \bmod N + kN$之中的B的值，選擇性的指定爲A in或是1的手段。

9．如申請專利然第8項之資料處理裝置，其中，前述控制手段，係使用前述直列接續之積和演算器而選擇地實行乘數與被乘數係多倍長度之乘法運算。

10．一種微電腦，係將申請專利範圍第9項之資料處理裝置，即將此資料處理裝置作爲共同處理器利用之微處理器包含於1個半導體積板，前述微處理器，係對於前述資料處理裝置設定前述演算式$A_{out}= A_{in} \cdot B \cdot R^{-1} \bmod N + kN$所必要的輸入值指示對應的演算處理，而利用該演算結果。

11．如申請專利範圍第10項之微電腦，其中，於前述微處理器具有保存供實行冪函數剩餘演算用的動作程式之ROM，前述演算$A_{out}= A_{in} \cdot B \cdot R^{-1} \bmod N + kN$係包含於前述冪函數剩餘演算。

12．如申請專利範圍第11項之微電腦，其中，進而具備有將冪函數剩餘演算的對象之資料可以電氣改寫地保存之非易失性記憶體。

13．一種資料處理裝置，係將演算結果A_{out}作爲下一個演算的輸入值A_{in}，實行演算式$A_{out}=f(A_{in})$所給定之演算的資料處理裝置，係包含有：

可以分別記憶A_{in}與A_{out}的內容之記憶手段，及

對於該記憶手段指示A_{in}與A_{out}的記憶場所的相互

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製

六、申請專利範圍

交換的旗標手段，及

於前述計算式的演算實行結束後藉由將前述旗標手段的值反轉而取代使A_{in}與A_{out}的值的物理上的交換，而使A_{in}與A_{out}的記憶場所在邏輯上交換的控制手段等而構成。

１４．一種資料處理裝置，係將當n爲代表演算值的位元數之正整數，N爲$0 < N < 2^n$的n位元的正整數，A_{in}爲$0 \leq A_{in} < 2^n$的n位元的正整數，A_{out}爲$0 \leq A_{out} < 2^n$的n位元的正整數，k爲0或是正整數時，演算式爲$A_{out}=f(A_{in}) \bmod N + kN$，進行輸入值$A_{in}$與輸出值$A_{out}$的值的範圍在N以上不滿$2^n$的剩餘演算之演算手段，以及藉由此演算手段而控制前述剩餘演算之控制手段製作於１個半導體基板上；

前述「$A_{out}=f(A_{in}) \bmod N + kN$」的剩餘演算，包含$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N + kN$的演算式所給定的剩餘演算，其中，R爲$2^n$，$R^{-1}$爲滿足$R \cdot R^{-1} \bmod N = 1$的$0 < R^{-1} \leq N$之n位元的正整數，B係$0 \leq B < 2^n$之n位元的正整數；

前述演算手段，係具備有供實行演算式$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N + kN$所給定的剩餘演算用的，直列式接續的積和演算器。

１５．如申請專利範圍第１４或１５項之資料處理裝置，其中，前述演算式$A_{out}=A_{in} \cdot B \cdot R^{-1} \bmod N + kN$所給定的剩餘演算中，包含$t=(A_{in} \cdot B+M \cdot N)/R$所給定的演

（請先閱讀背面之注意事項再填寫本頁）

訂

線

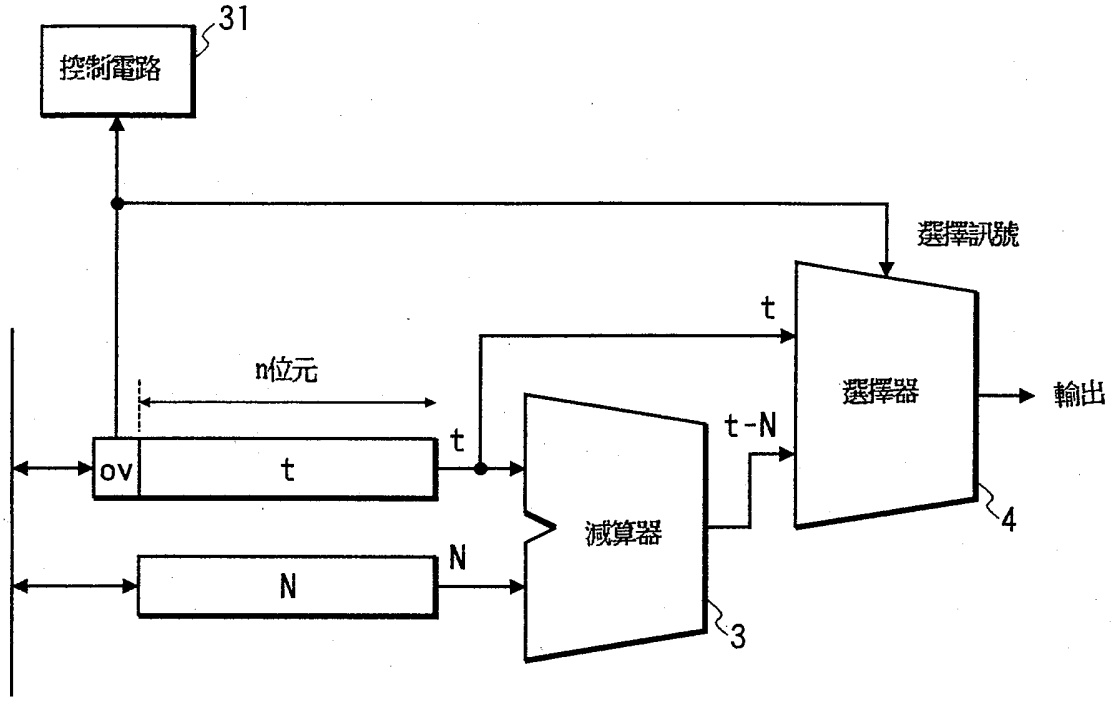
經濟部中央標準局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

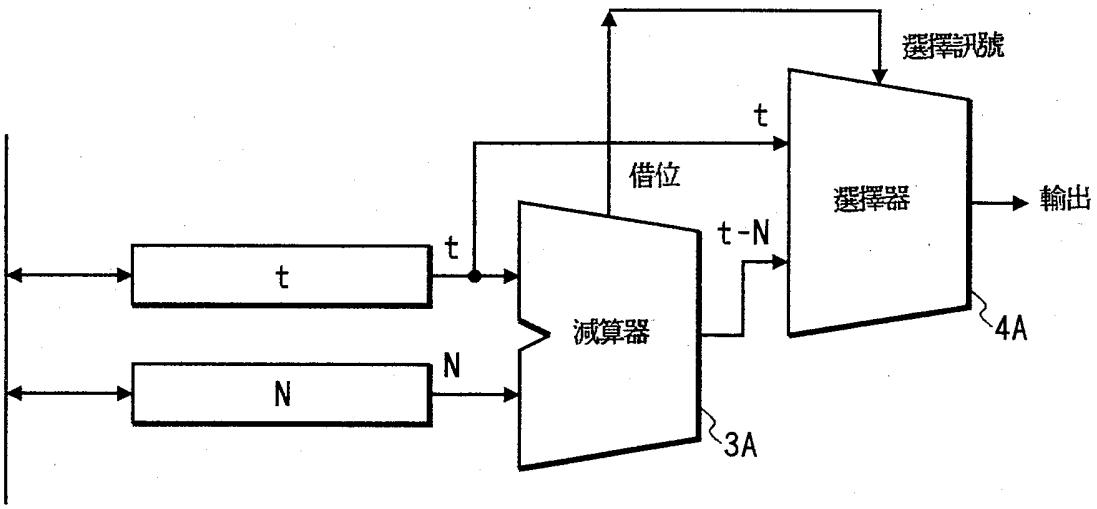
六、申請專利範圍

算處理，及接著給定 if t≧R then return t-N else return t，及將 t－N 或者是 t 作爲前述 A_{out} 之演算處理；前述演算手段，藉由直列接續的前述積和演算器，實行前述演算處理 $t=(A_{in}\cdot B+M\cdot N)/R$ 作爲部分積之和。

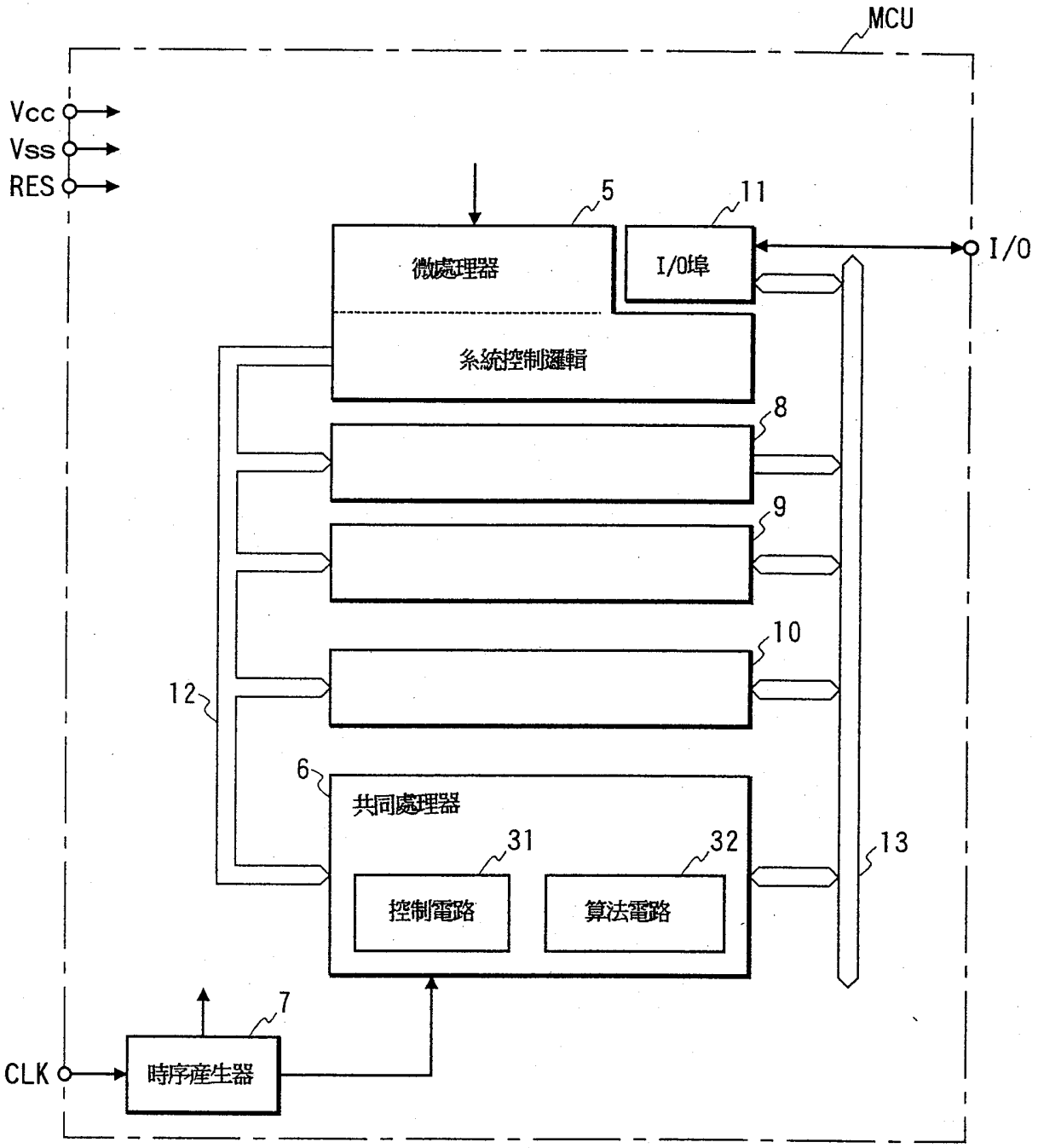
（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部中央標準局員工消費合作社印製
本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）


31
控制電路
選擇訊號
t
n位元
選擇器
輸出
t-N
ov
t
減算器
4
N
N
3

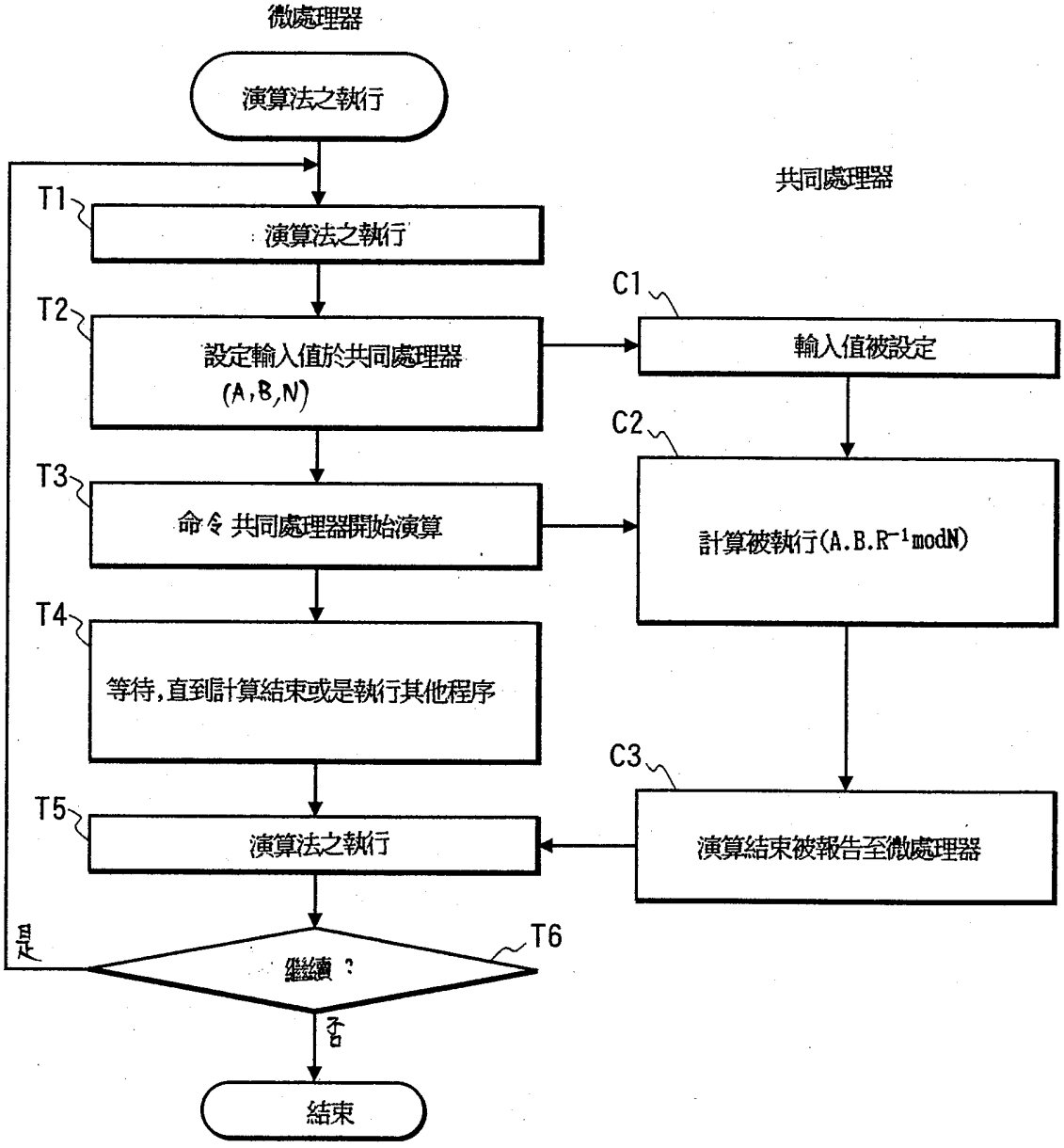
第 1 圖


選擇訊號
t
借位
選擇器
輸出
t-N
t
t
減算器
4A
N
N
3A

第 2 圖

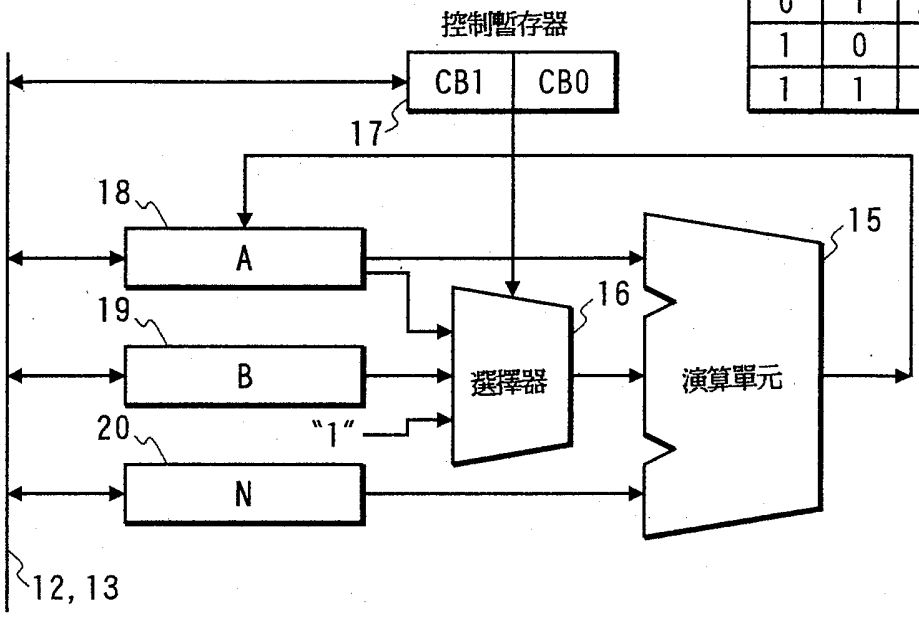

MCU
Vcc
Vss
RES
5
11
微處理器
I/O埠
I/O
系統控制邏輯
8
9
10
12
6
共同處理器
31
32
13
控制電路
算法電路
7
CLK
時序產生器

第 3 圖

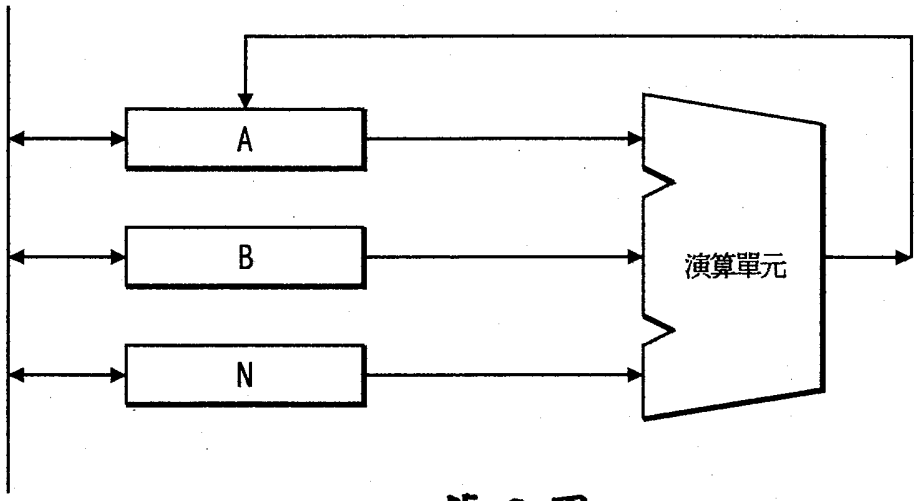

微處理器
演算法之執行
T1
演算法之執行
T2
設定輸入值於共同處理器
(A,B,N)
T3
命令 共同處理器開始演算
T4
等待,直到計算結束或是執行其他程序
T5
演算法之執行
T6
繼續？
是
否
結束
共同處理器
C1
輸入值被設定
C2
計算被執行(A.B.R^{-1}modN)
C3
演算結束被報告至微處理器

第 4 圖

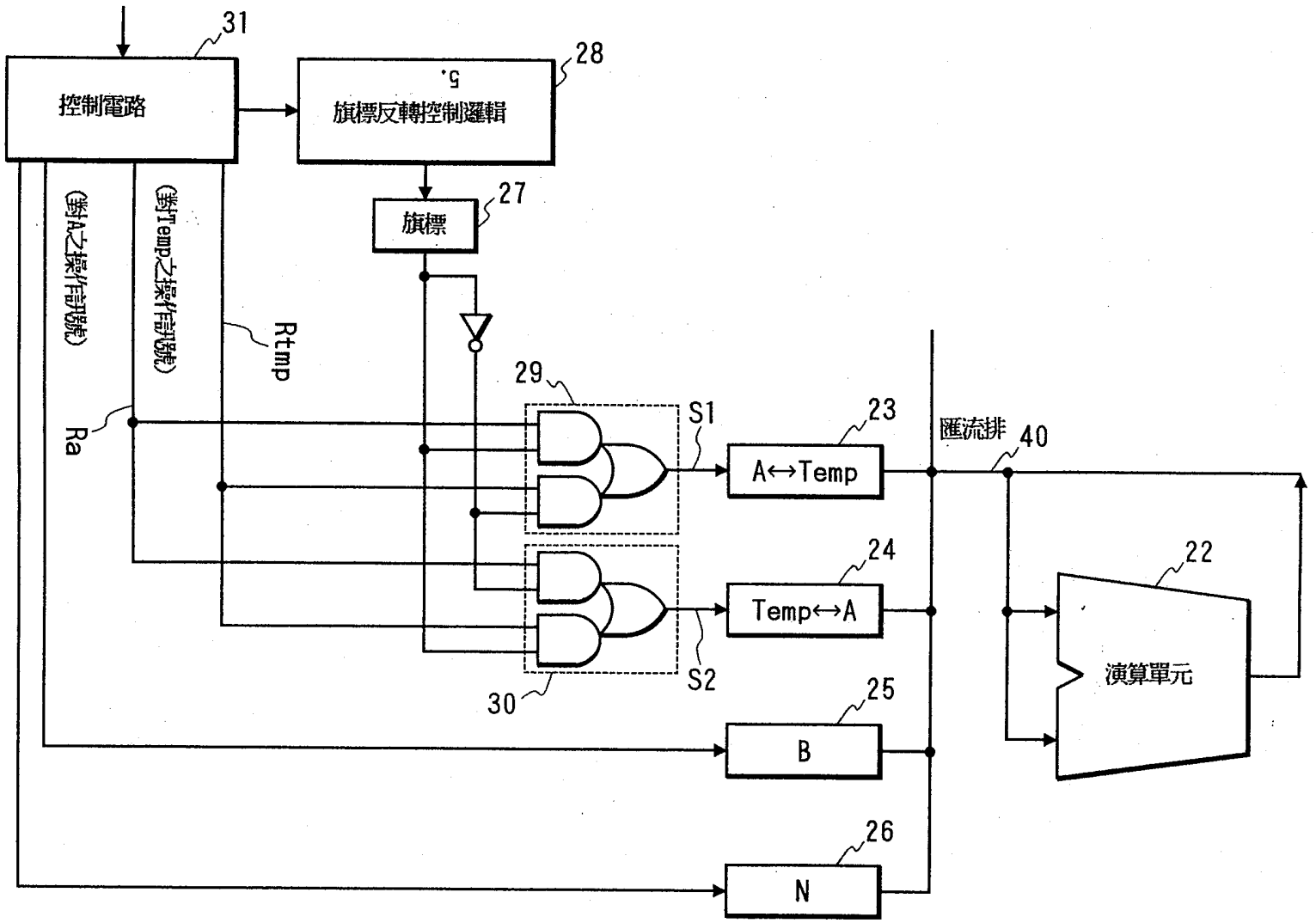
CB1	CB2	操作表示
0	0	$ABR^{-1} \bmod N + kN$
0	1	$A^2R^{-1} \bmod N + kN$
1	0	$AR^{-1} \bmod N + kN$
1	1	



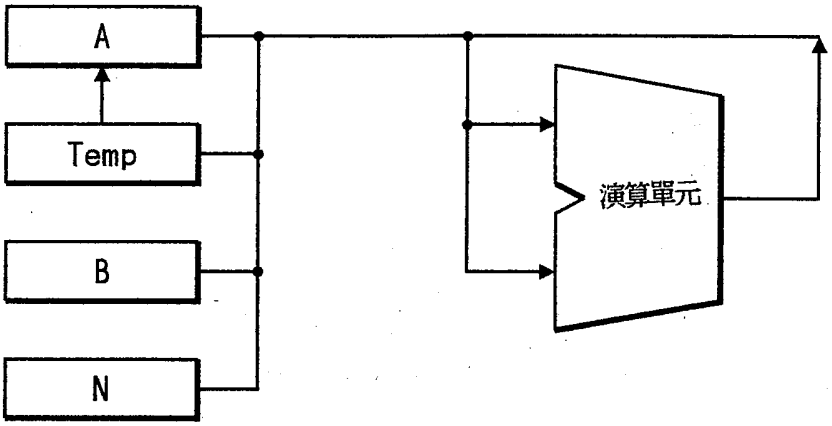
第 5 圖



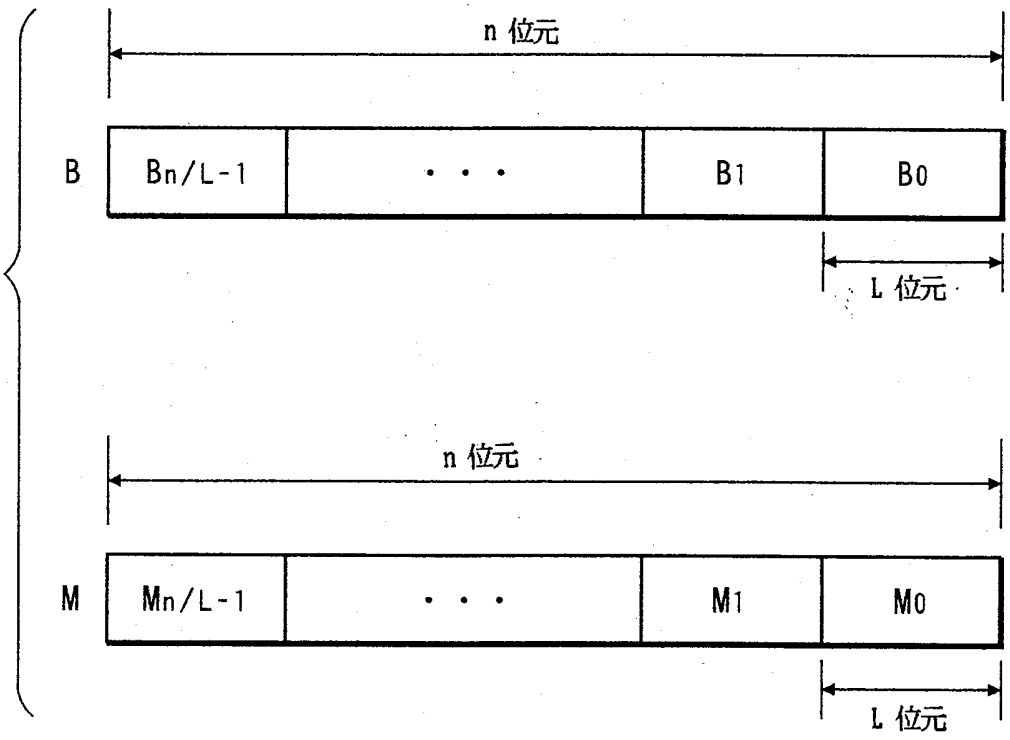
第 6 圖


31
控制電路
28
旗標反轉控制邏輯
27
旗標
(對A之操作訊號)
(對Temp之操作訊號)
Rtmp
Ra
29
S1
23
A↔Temp
匯流排
40
24
Temp↔A
S2
30
22
演算單元
25
B
26
N

第7圖

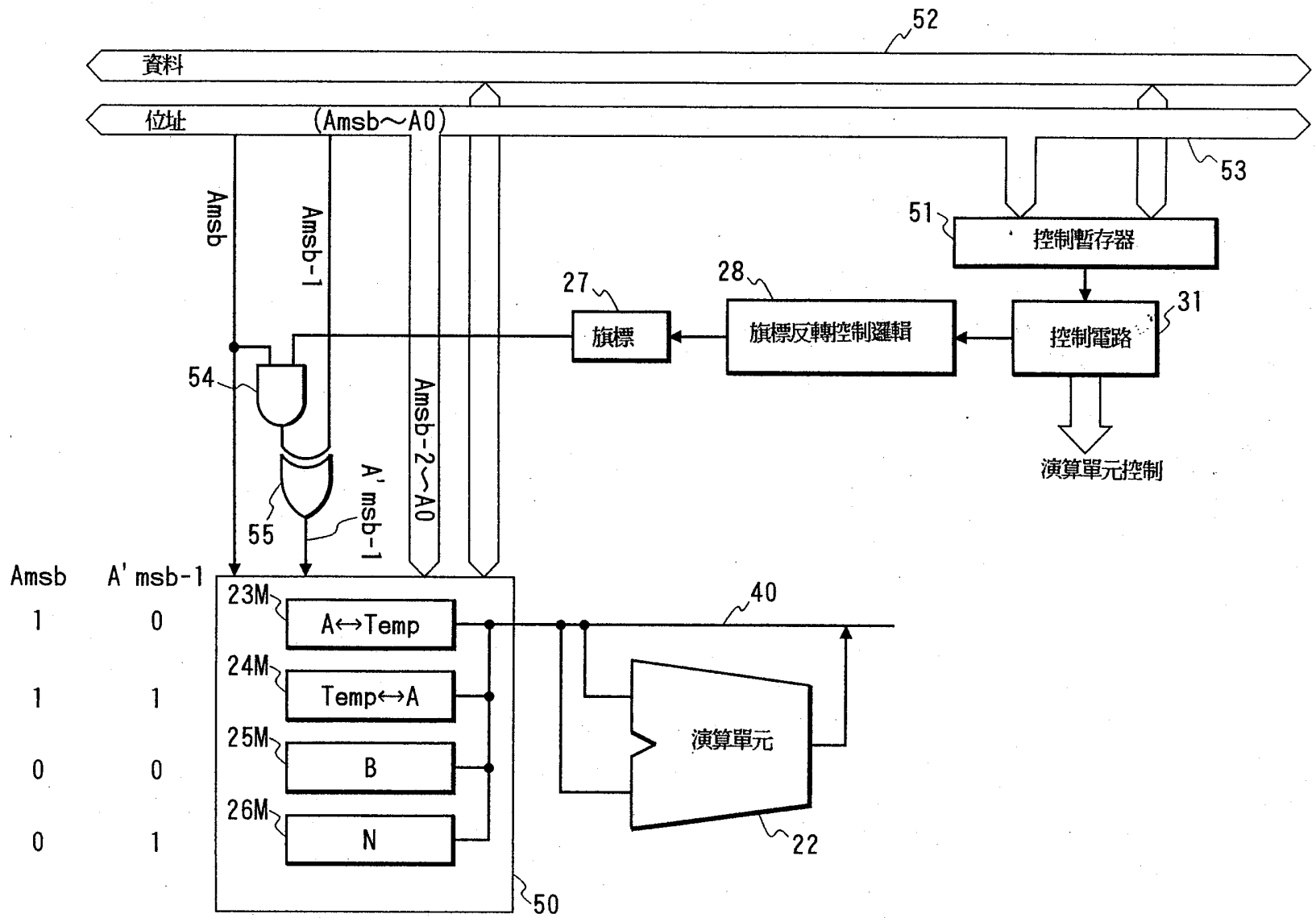


第 8 圖

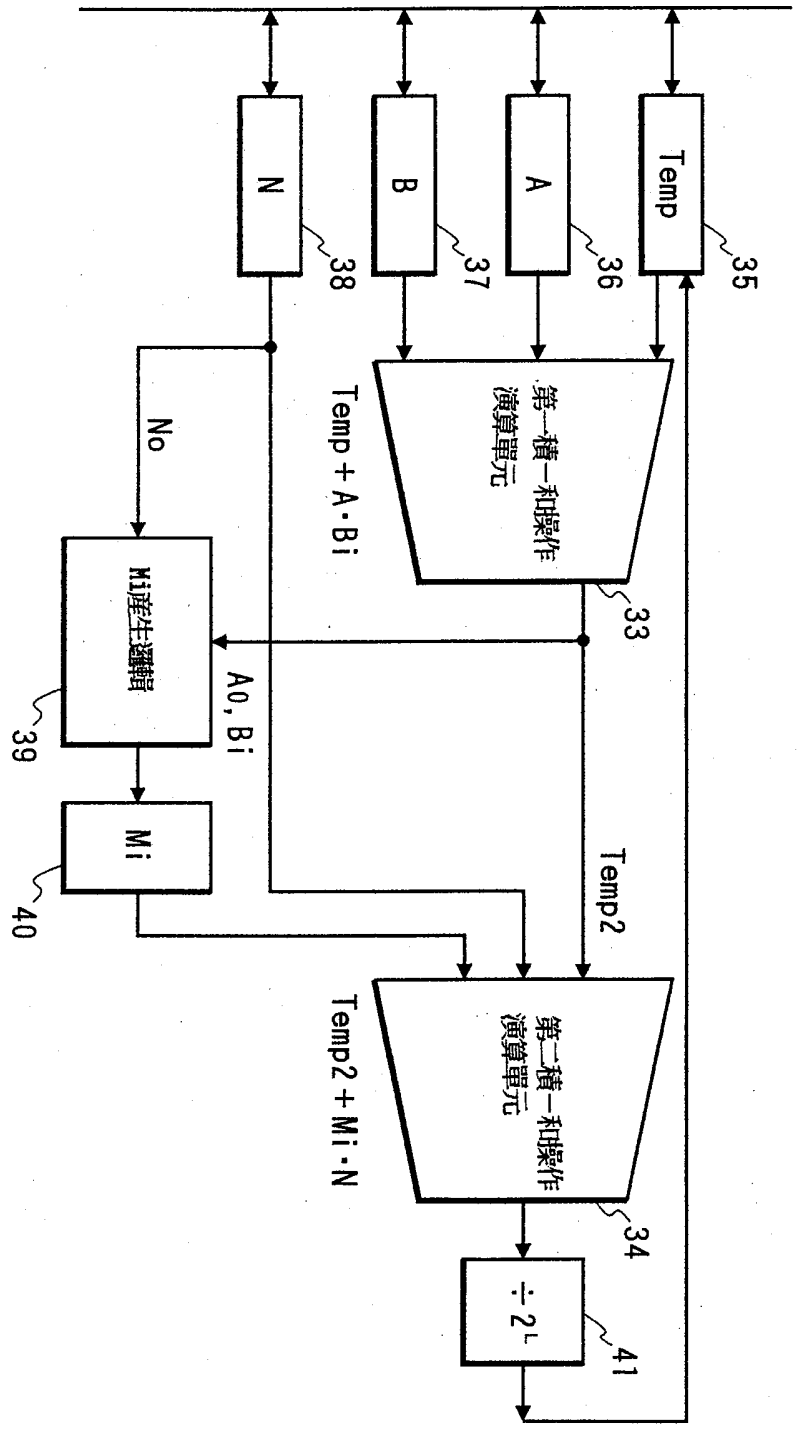


第10圖

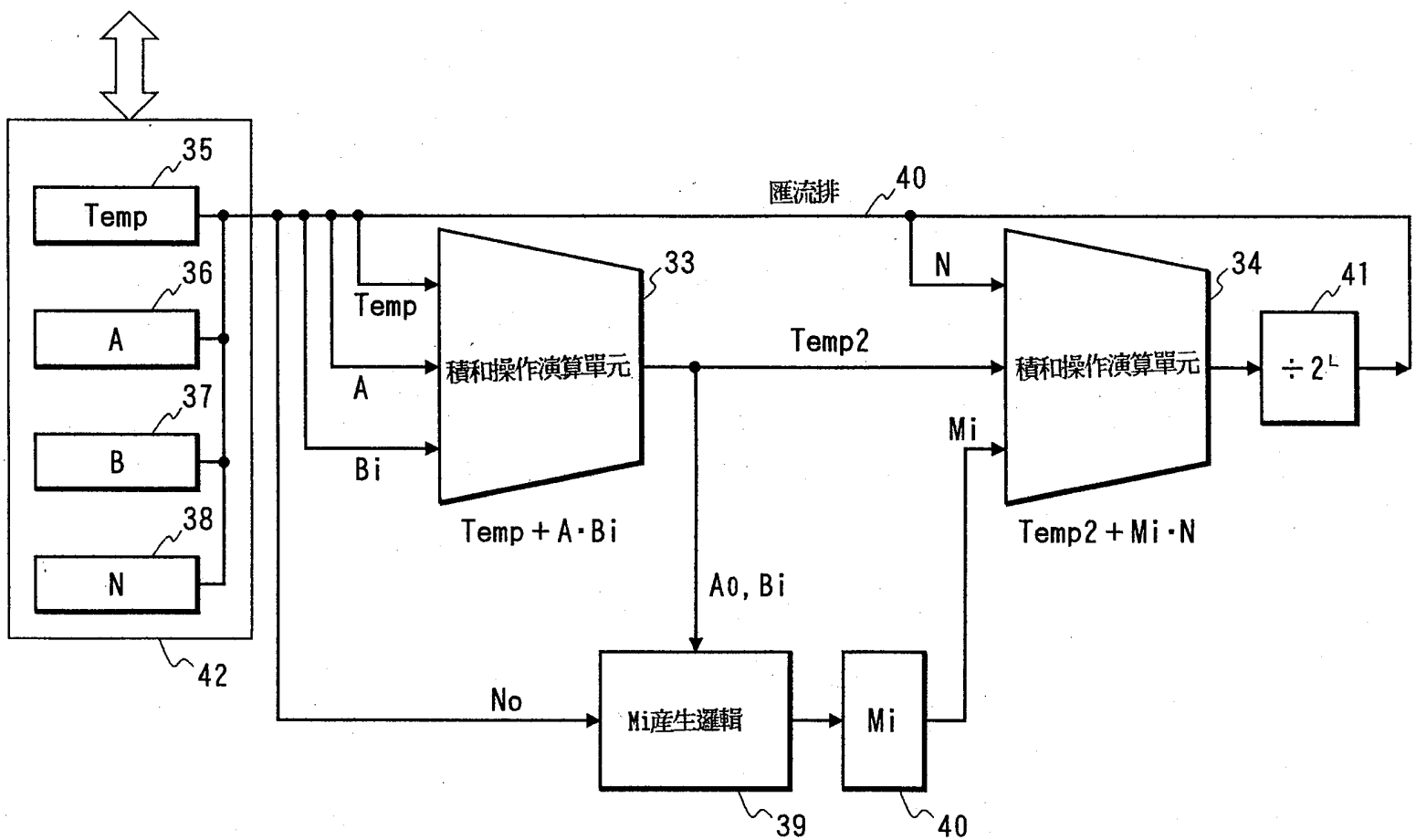
373294


52
資料
位址
(Amsb～A0)
53
Amsb
Amsb-1
Amsb-2～A0
51
控制暫存器
27
旗標
28
旗標反轉控制邏輯
31
控制電路
演算單元控制
54
55
A' msb-1
Amsb
A' msb-1
1
0
1
1
0
0
0
1
23M
A↔Temp
24M
Temp↔A
25M
B
26M
N
50
40
演算單元
22

第 9 圖


Temp
35
A
36
B
37
N
38
第一積一和操作演算單元
Temp + A · Bi
33
No
Mi產生邏輯
39
Ai, Bi
Mi
40
Temp2
第二積一和操作演算單元
Temp2 + Mi · N
34
÷ 2^L
41

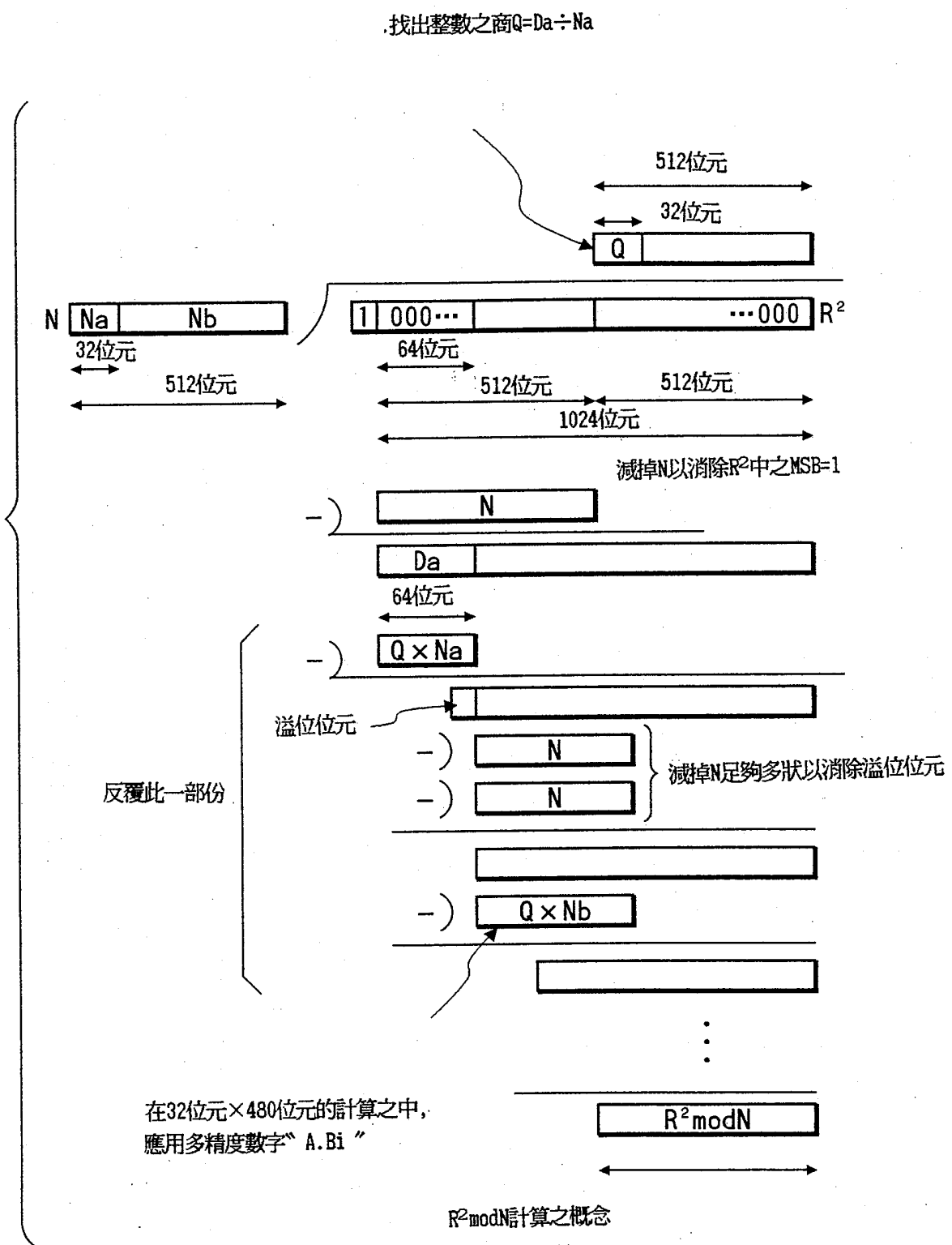
第11圖



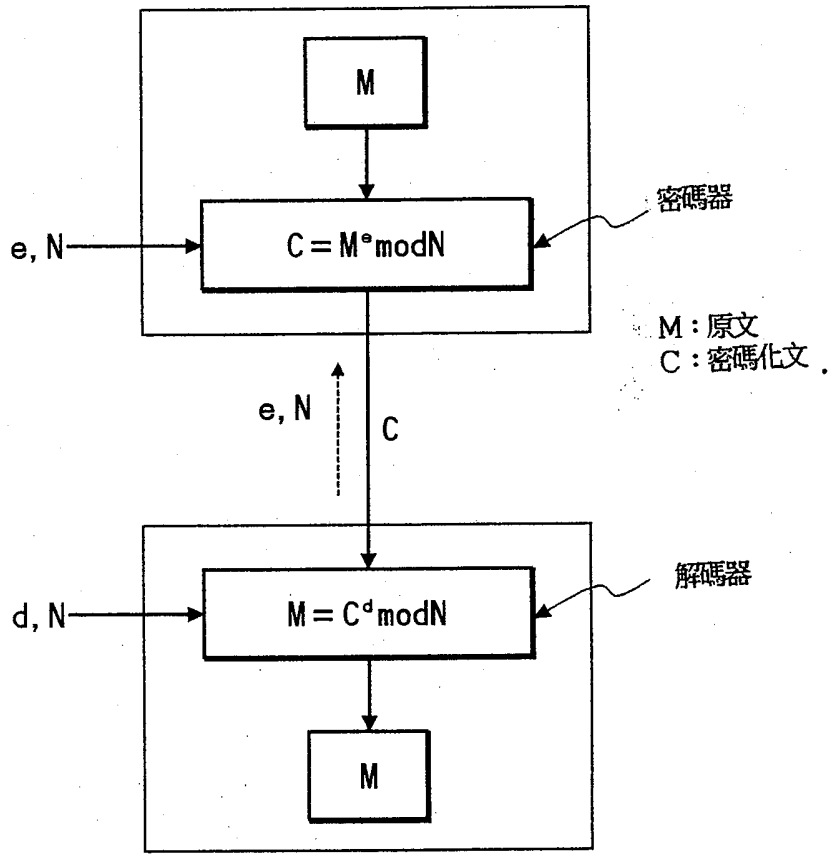
35
Temp
36
A
37
B
38
N
42
匯流排
40
Temp
A
Bi
33
積和操作演算單元
Temp + A·Bi
Temp2
N
34
積和操作演算單元
Temp2 + Mi·N
Mi
41
÷2^L
A0, Bi
No
Mi產生邏輯
39
Mi
40

第12圖

373294


找出整數之商Q=Da÷Na
512位元
32位元
Q
N
Na
Nb
32位元
512位元
1 000…
…000
R²
64位元
512位元
512位元
1024位元
減掉N以消除R²中之MSB=1
N
Da
64位元
Q×Na
溢位位元
N
N
減掉N足夠多狀以消除溢位位元
反覆此一部份
Q×Nb
在32位元×480位元的計算之中，
應用多精度數字" A.Bi "
R²modN
R²modN計算之概念

第13圖



第14圖