



(21)申請案號：100139777

(22)申請日：中華民國 100 (2011) 年 11 月 01 日

(51)Int. Cl. : H03M13/15 (2006.01)

(71)申請人：國立交通大學(中華民國) NATIONAL CHIAO TUNG UNIVERSITY (TW)
新竹市大學路 1001 號

(72)發明人：林義閔 LIN, YI MIN (TW)；許智翔 HSU, CHIH HSIANG (TW)；張錫嘉 CHANG, HSIE CHIA (TW)；李鎮宜 LEE, CHEN YI (TW)

(74)代理人：高玉駿；楊祺雄

申請實體審查：有 申請專利範圍項數：10 項 圖式數：4 共 24 頁

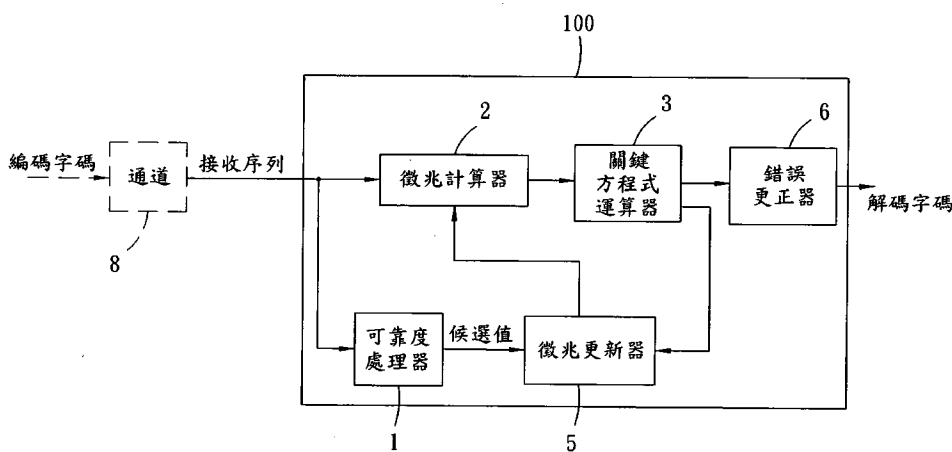
(54)名稱

循環碼解碼器及其方法

CYCLIC CODE DECODER AND METHOD THEREOF

(57)摘要

一種循環碼解碼器包含：一可靠度處理器，為接收字碼的 q 個不可靠符元設定 2^q 組候選值；一徵兆計算器，為接收字碼計算相關於其中一組候選值的 $2t$ 個徵兆值，其中 t 為接收字碼的錯誤更正能力；一關鍵方程式運算器，根據徵兆值產生一錯誤位置多項式；及一錯誤更正器。當判斷出多項式階數大於一門檻，徵兆計算器會更新該 $2t$ 個徵兆值使相關於另一組候選值，且關鍵方程式運算器會以更新後的徵兆值調整多項式；直到調整出的多項式階數不大於門檻，錯誤更正器會基於最後調整出之多項式根值來更正接收字碼。此外，一種循環碼解碼方法也被揭露。



- 1：可靠度處理器
- 2：徵兆計算器
- 3：關鍵方程式運算器
- 5：徵兆更新器
- 6：錯誤更正器
- 8：通道
- 100：循環碼解碼器

發明專利說明書

(本說明書格式、順序，請勿任意更動，※記號部分請勿填寫)

※ 申請案號： 100139777

※ 申請日： 100. 11. 01

※IPC 分類：H03M 13/15 (2006.01)

一、發明名稱：(中文/英文)

循環碼解碼器及其方法 / Cyclic code decoder and method thereof

二、中文發明摘要：

一種循環碼解碼器包含：一可靠度處理器，為接收字碼的 q 個不可靠符元設定 2^q 組候選值；一徵兆計算器，為接收字碼計算相關於其中一組候選值的 $2t$ 個徵兆值，其中 t 為接收字碼的錯誤更正能力；一關鍵方程式運算器，根據徵兆值產生一錯誤位置多項式；及一錯誤更正器。當判斷出多項式階數大於一門檻，徵兆計算器會更新該 $2t$ 個徵兆值使相關於另一組候選值，且關鍵方程式運算器會以更新後的徵兆值調整多項式；直到調整出的多項式階數不大於門檻，錯誤更正器會基於最後調整出之多項式根值來更正接收字碼。此外，一種循環碼解碼方法也被揭露。

三、英文發明摘要：

A cyclic code decoder comprises a reliability processor setting 2^q candidates for q unreliable symbols of a received codeword, a syndrome calculator obtaining $2t$ syndromes relative to one of the candidates for the received codeword wherein t stands for the error-correct ability of the received codeword, a key-equation operator generating an error position polynomial based on the syndromes, and an error

corrector. When the polynomial order is greater than a threshold, the syndrome calculator updates the $2t$ syndromes to be relative to another candidate, and the key-equation operator revises the polynomial with the updated syndromes. When the order of the revised polynomial is not greater than the threshold, the error corrector corrects the received codeword according to the roots of the last updated polynomial. Besides, a cyclic code decoding method is also disclosed.

四、指定代表圖：

(一)本案指定代表圖為：圖(1)。

(二)本代表圖之元件符號簡單說明：

100 …… 循環碼解碼器

5 …… 徵兆更新器

1 …… 可靠度處理器

6 …… 錯誤更正器

2 …… 徵兆計算器

8 …… 通道

3 …… 關鍵方程式運算器

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

六、發明說明：

【發明所屬之技術領域】

本發明是有關於一種解碼器及其方法，特別是指一種循環碼(cyclic code)解碼器及其方法。

【先前技術】

一循環碼編碼字碼從傳送端發出後，會藉由通道傳遞而在接收端形成一接收序列，其中編碼字碼具有多個為硬性資訊的符元，而接收序列則具有多個為軟性資訊的元素。

最為常見的接收端解碼方式是基於 Chase 演算法，先對接收序列的多個元素進行可靠度評估以產生 R 個評估後字碼，接著解碼出 R 個對應候選字碼，最後再從這些候選字碼中選出最接近接收序列者當作解碼碼字。此可參考 D. Chase 於 1972 年提出的 "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Theory*, vol.1T-18, p.170~182。

其中，最常使用歐式距離(Euclidean distance)法則來計算哪一候選字碼最接近接收序列。但是，接收序列的各元素均為軟性資訊而呈現浮點數型式，所以這個步驟的計算十分繁瑣。

此外，習知循環碼的解碼過程中，不管是採用 Berlekamp-Massey(BM)的迭代計算，或是採用 Euclid 的輾轉相除法來求取錯誤位置多項式，實現電路都相當複雜。且 Chase 演算法又必須搭配 R 套解碼電路來解碼該 R 個候

選字碼，因此電路成本頗高。

【發明內容】

因此，本發明之目的，即在提供一種能降低電路成本的循環碼解碼器及其方法。

於是，本發明循環碼解碼方法，包含以下步驟：(A)藉由一可靠度處理器處理一個具有 n 個符元的接收字碼，並為其中 q 個不可靠符元設定 $R=2^q$ 組候選值， $n>q>1$ ；(B)藉由一徵兆計算器為該接收字碼計算出相關於其中一組候選值的 $2t$ 個徵兆值，其中 t 為該接收字碼所對應之一循環碼編碼字碼的錯誤更正能力；(C)藉由一關鍵方程式運算器根據該等徵兆值產生一個階數最多為 t 的錯誤位置多項式；(D)當該錯誤位置多項式的階數大於一門檻，藉由該徵兆計算器更新該 $2t$ 個徵兆值使相關於另一組候選值，其中該門檻 $<t$ ；(E)重複步驟(C)~(D)，直到該錯誤位置多項式的階數不大於該門檻，則藉由一錯誤更正器基於最後調整出之錯誤位置多項式的根值來對該接收字碼進行錯誤更正。

而本發明循環碼解碼器，包含：一可靠度處理器，用以處理一個具有 n 個符元的接收字碼，並為其中 q 個不可靠符元設定 $R=2^q$ 組候選值， $n>q>1$ ；一徵兆計算器，為該接收字碼計算出相關於其中一組候選值的 $2t$ 個徵兆值，其中 t 為該接收字碼所對應之一循環碼編碼字碼的錯誤更正能力；一關鍵方程式運算器，根據該等徵兆值產生一個階數最多為 t 的錯誤位置多項式；及一錯誤更正器，用以對該接收字碼進行錯誤更正；當該關鍵方程式運算器判斷出該錯

誤位置多項式的階數大於一門檻，該徵兆計算器會更新該 $2t$ 個徵兆值使相關於另一組候選值，且該關鍵方程式運算器會以更新後的該等徵兆值調整該錯誤位置多項式，其中該門檻 $< t$ ；直到調整出的錯誤位置多項式階數不大於該門檻，該錯誤更正器會基於最後調整出之錯誤位置多項式的根值來對該接收字碼進行錯誤更正。

【實施方式】

有關本發明之前述及其他技術內容、特點與功效，在以下配合參考圖式之二個較佳實施例的詳細說明中，將可清楚的呈現。

在本發明被詳細描述之前，要注意的是，在以下的說明內容中，類似的元件是以相同的編號來表示。

第一較佳實施例

參閱圖 1，本發明循環碼解碼器 100 之第一較佳實施例適用於經由一通道 8 接收一個具有 n 個元素的接收序列，以找出接收序列中哪幾個元素受到通道 8 干擾並進行錯誤更正，而得到一個逼近於接收序列的解碼字碼。

由於接收序列是取自通道 8，而解碼字碼是經過數位解碼產生，因此本例特別定義接收序列具有的 n 個元素為軟性資訊而呈現類比形式(如浮點數 0.78 或 1.26)，解碼字碼則對應地具有 n 個呈數位形式的符元。

圖 1 的循環碼解碼器 100 包含一可靠度處理器 1、一徵兆(syndrome)計算器 2、一關鍵方程式運算器 3、一徵兆更新器 5 及一錯誤更正器 6。

可靠度處理器 1 評估接收序列的 n 個類比形式元素來分別對應得到一第一評估字碼的 n 個數位形式符元，且從中選出最不可靠的 q 個符元，並為該 q 個符元設定 $R=2^q$ 組候選值， $1 < q < n$ 。徵兆計算器 2 為該接收序列計算出相關於其中一組候選值的 $2t$ 個徵兆值，關鍵方程式運算器 3 則根據該等徵兆值產生一個階數最多為 t 的錯誤位置多項式，其中 t 為接收序列所對應之循環碼編碼字碼的錯誤更正能力。

當錯誤位置多項式的階數不大於一門檻，錯誤更正器 6 利用該錯誤位置多項式找出接收序列的哪幾個元素受到通道 8 干擾，並進行錯誤更正。而當錯誤位置多項式的階數大於該門檻，徵兆更新器 5 則選用另一組候選值來產生 $2t$ 個分別對應該等徵兆值的調整因子，以供徵兆計算器 2 更新對應的徵兆值來送往關鍵方程式運算器 3。

為便於說明循環碼解碼器 100 所執行之本發明循環碼解碼方法的較佳實施例，以下假設一編碼字碼經由通道 8 傳遞而在循環碼解碼器 100 處形成該接收序列，這個編碼字碼基於 BCH(Bose-Chaudhuri Hocquenghem)編碼方式而具有 $n=15$ 個數位形式的符元，且如熟悉本技藝者所熟知：BCH 字碼的各符元僅含有 $m=1$ 個位元，所以符元值非 1 即 0。這意味著接收序列的各元素只有 $m=1$ 個浮點數信號，且評估字碼和解碼字碼的各符元只有 $m=1$ 個位元。

解碼方法包含圖 2 的以下步驟：

步驟 71：可靠度處理器 1 評估接收序列的 n 個類比形式元素來分別對應得到第一評估字碼的 n 個數位形式符元

，且從中選出最不可靠的 q 個符元。

較佳地，本例採用 LLR(對數似然比，log-likelihood ratio)來決定各類比形式元素較可能被判斷成為 1 值的符元或較可能為 0 值的符元。倘若元素為 1 與為 0 的機率相當，則稱對應符元不可靠。當然在其他應用中判斷方式不限於此。

步驟 72：可靠度處理器 1 為選出的 q 個符元設定 $R=2^q$ 組候選值。

例如：第一評估字碼中，位於第二、第五和第八位置的符元被選為最不可靠，那這三個符元會有 $R=2^3$ 組候選值，即 $\{0,0,0\}$ 、 $\{0,0,1\}$ 、 $\{0,1,0\}$ 、 $\{0,1,1\}$ 、 $\{1,0,0\}$ 、 $\{1,0,1\}$ 、 $\{1,1,0\}$ 、 $\{1,1,1\}$ 。

步驟 73：徵兆計算器 2 為該接收序列計算出相關於其中一組候選值的 $2t$ 個徵兆值。

具體來說，徵兆計算器 2 會以類似可靠度處理器 1 的評估方式來分析接收序列的 n 個類比形式元素而得到一第二評估字碼的 n 個數位形式符元，然後據以計算 $2t$ 個徵兆值。請注意，將第二評估字碼中「對應於第一評估字碼之 q 個最不可靠符元」的 q 個符元集合起來，即為步驟 73 所稱的相關候選值。

步驟 74：關鍵方程式運算器 3 根據 $2t$ 個徵兆值產生一個階數最多為 t 的錯誤位置多項式。

步驟 75：關鍵方程式運算器 3 判斷錯誤位置多項式的階數是否不大於一門檻。當判斷出階數不大於該門檻，流

程跑到步驟 77；而當判斷出階數大於該門檻，流程跑到步驟 76。

較佳地，門檻 $=t-1$ 。這是因為錯誤位置多項式的階數通常代表著該接收序列受到通道 8 干擾的元素數目，而階數達最高值 t 時，更意味著受干擾元素數目至少為 t 。假設受干擾元素數目 $=t$ ，那麼這個錯誤位置多項式的 t 個根值可被求出；但是若受干擾元素數目 $>t$ ，則根值無解。因此為了避免無解的窘境，本例不在階數 $=t$ 時進行多項式求解。當然，在其他應用中，也可以依據通道 8 狀況設定較小的門檻值，如： $(t-2)$ 、 $(t-3)$...

步驟 76：徵兆更新器 5 選用該 q 個不可靠符元的另一組候選值來求取 $2t$ 個分別對應該等徵兆值的調整因子，以供徵兆計算器 2 更新對應的徵兆值，然後跳回步驟 74 以更新後的徵兆值調整錯誤位置多項式。

在此舉例說明如何更新徵兆值，假設以接收多項式 $P(x)=r_0+r_1 \cdot x+r_2 \cdot x^2+r_3 \cdot x^3+\dots+r_{14} \cdot x^{14}$ 來表示第二評估字碼的 $n=15$ 個符元 $r_0, r_1, r_2, r_3, \dots, r_{14}$ 。徵兆計算器 2 先在步驟 73 將 $2t$ 個非零元素分別代入接收多項式來算出 $2t$ 個對應的徵兆值。然後，徵兆更新器 5 於步驟 76 以另一組候選值來取代第二評估字碼的其中 q 個符元，且計算這個取代程序會對各徵兆值產生多大變動，以做為更新對應徵兆值的調整因子。

計算方式為：若以 $\{s_1(i), s_1(j), s_1(k)\}$ 表示前一次選用的候選值，以 $\{s_2(i), s_2(j), s_2(k)\}$ 表示目前選用的候選值，當非零元素為 a 且符元 r_i, r_j, r_k 不可靠，調整因子 =

$\{s_2(i) \cdot \alpha^i + s_2(j) \cdot \alpha^j + s_2(k) \cdot \alpha^k\} - \{s_1(i) \cdot \alpha^i + s_1(j) \cdot \alpha^j + s_1(k) \cdot \alpha^k\}$ 。例如： $\{r_2, r_5, r_8\}$ 從候選值 $\{0,1,0\}$ 換成 $\{0,1,1\}$ ，那麼調整因子 $= \{0 \cdot \alpha^2 + 1 \cdot \alpha^5 + 1 \cdot \alpha^8\} - \{0 \cdot \alpha^2 + 1 \cdot \alpha^5 + 0 \cdot \alpha^8\}$ 。

步驟 77：錯誤更正器 6 解析錯誤位置多項式，以找出該接收序列受到通道 8 干擾的元素位置並進行錯誤更正來得到該解碼字碼。

詳細來說，錯誤更正器 6 會以該 q 個不可靠元素最後選用的候選值來更新第二評估字碼，並且求出錯誤位置多項式的根值來得知需進行錯誤更正的元素位置，接著修改第二評估字碼中和那些位置相呼應的符元。以 BCH 編碼來說，符元值非 1 即 0，所以修改動作是將該符元反相即可。

值得注意的是，本例循環碼解碼器 100 不侷限使用於 BCH 編碼應用，也可適用於其他循環碼，如：RS(Reed Solomon, 李德所羅門)編碼。本發明領域具有通常知識者可類推前述解碼方法而找出 RS 接收序列受到通道干擾的元素位置，故在此不多贅述。不過，RS 符元具有 $m(m>1)$ 個位元，因此符元的可能值有 2^m 個，無法僅以簡單的反相動作進行修改。所以，錯誤更正器 6 還會找出匹配於錯誤位置多項式的錯誤值多項式，藉以求出錯誤更正後的適當符元值。

又，應用於 RS 解碼時，接收序列之各元素所具有的 m 個信號為類比形式，且評估字碼之各符元所具有的 m 個位元為數位形式。可靠度處理器 1 會逐一評估這些元素的各個信號是較偏向 1 值的位元或較偏向 0 值的位元。當其中

一信號為 1 值與為 0 值的機率相當，該可靠度處理器 1 判斷對應位元不可靠。

此外，步驟 73 不限於在步驟 71~72 後執行，也可以同時執行，或早於步驟 71~72。並且，徵兆更新器 5 於步驟 76 選用另一組候選值的選擇順序可以是基於二進位碼(Binary Code)或是格雷碼(Gray Code)。例如：當第一個選用的候選值是 $\{0,1,0\}$ ，可依據二進位方式依序選取 $\{0,1,1\}$ 、 $\{1,0,0\}$ 、 $\{1,0,1\}$...，其中各候選值的集合值會遞增。或者，可依據格雷碼方式依序選取 $\{1,1,0\}$ 、 $\{1,1,1\}$ 、 $\{1,0,1\}$...，其中任意兩個相鄰候選值只有一個位元改變。

又，雖然步驟 71 是希望選出「最不可靠」的 q 個符元，但是如果有多個符元不可靠度相當，也可以選取其中 q 個「較不可靠」的符元即可。

由前述流程可以明顯觀察到：本例會一一檢視該等候選值所對應的錯誤位置多項式，直到檢驗出錯誤位置多項式的階數不大於該門檻，才停止。所以求取錯誤位置多項式的次數 \leq 候選值 $R=2^q$ ，且只要針對最後調整出的錯誤位置多項式進行求解即可。反觀習知 Chase 演算法，必須搭配 R 套電路來算出 R 組候選值的錯誤位置多項式並分別求出根值，運算次數和電路成本都遠高於本例。

此外，本例以更新徵兆值的方式來得到下一個候選值所對應的錯誤位置多項式，而不是像習知技術般為每一個候選值都獨立算出對應的徵兆值，所以省去了許多重複的計算電路。

參閱圖 3，其顯示了本例相對於 Chase 演算法的字碼錯誤率模擬。這個模擬使用了一個 RS(255,239)編碼字碼，具有以 $k=239$ 個資料符元編碼而得到的 $n=255$ 個編碼符元。從圖示可觀察到：選取出的不可靠元素數目 q 越多，字碼錯誤率越低。特別說明的是，本例於 $q=5$ 的字碼錯誤率和 Chase 演算法於 $q=3$ 時的表現相當，但是從表一可得知本例所需進行的運算次數明顯少於 Chase 演算法。

表一

	本例($q=5$)			Chase 演算法($q=3$)		
	6.0	6.5	7.0	6.0	6.5	7.0
訊雜比	6.0	6.5	7.0	6.0	6.5	7.0
徵兆值 求取次數	5.22	1.30	1.07	8	8	8
位置多項式 求取次數	5.22	1.30	1.07	8	8	8
錯誤位置 搜尋次數	5.22	1.30	1.07	8	8	8
錯誤值 更正次數	5.22	1.30	1.07	8	8	8

表二更比較了本例和 Chase 演算法的硬體實現成本，不難察覺本例的邏輯閘數(gate count)少於 Chase，尤其是記憶體的使用。其中，Chase 相關數據取自 Zhang 提出的 "High-speed VLSI Architecture for Low-complexity Chase Soft-

decision Reed-Solomon Decoding”。

表二

	常 數 乘 法 器	變 數 乘 法 器	加 法 器	多 工 器	唯 讀 記 憶 體	隨 機 存 取 記 憶 體	暫 存 器	潛 時
本例	32	68	85	984	520	256×3	834	259
Chase	8	66	108	1109	1024	68+ 256×8	1606	528

第二較佳實施例

參閱圖 4，本發明循環碼解碼器 200 之第二較佳實施例不直接接收載有軟性資訊的接收序列，而是透過一前級電路 9 來接收相關信號。明確來說，該前級電路 9 會評估接收序列而送出具有 n 個數位形式符元的接收字碼，且會送出一個不可靠信號來提示接收字碼中哪 q 個符元不可靠。

因此，可靠度處理器 1' 不用進行評估與不可靠位置選取，可直接將接收字碼視為第一評估字碼，且根據不可靠信號為其中 q 個符元設定 $R=2^q$ 組候選值。也就是說，解碼方法省略了步驟 71。且步驟 73 中，徵兆計算器 2' 也不用進行評估，可直接拿接收字碼當作第二評估字碼來計算徵兆值。

綜上所述，前述較佳實施例中，循環碼解碼器 100、

200 會為該 R 個候選值求出對應的錯誤位置多項式，且直到錯誤位置多項式階數不大於門檻，才進行錯誤位置多項式求解。所以前述實施例不需像習知技術般準備 R 套電路，也不需要以歐式距離法則選擇最接近接收序列的候選字碼，能有效降低電路成本，故確實能達成本發明之目的。

惟以上所述者，僅為本發明之較佳實施例而已，當不能以此限定本發明實施之範圍，即大凡依本發明申請專利範圍及發明說明內容所作之簡單的等效變化與修飾，皆仍屬本發明專利涵蓋之範圍內。

【圖式簡單說明】

圖 1 是一方塊圖，說明本發明循環碼解碼器之第一較佳實施例；

圖 2 是一流程圖，說明循環碼解碼方法；

圖 3 是一模擬示意圖，說明本例相對於 Chase 演算法的字碼錯誤率；及

圖 4 是一電路圖，說明本發明循環碼解碼器之第二較佳實施例。

【主要元件符號說明】

100 ……	循環碼解碼器	5 ……	徵兆更新器
200 ……	循環碼解碼器	6 ……	錯誤更正器
1、1' ……	可靠度處理器	71~77 ……	步驟
2、2' ……	徵兆計算器	8 ……	通道
3 ……	關鍵方程式運算器	9 ……	前級電路

七、申請專利範圍：

1. 一種循環碼解碼方法，包含以下步驟：

(A)藉由一可靠度處理器處理一個具有 n 個符元的接收字碼，並為其中 q 個不可靠符元設定 $R=2^q$ 組候選值， $n>q>1$ ；

(B)藉由一徵兆計算器為該接收字碼計算出相關於其中一組候選值的 $2t$ 個徵兆值，其中 t 為該接收字碼所對應之一循環碼編碼字碼的錯誤更正能力；

(C)藉由一關鍵方程式運算器根據該等徵兆值產生一個階數最多為 t 的錯誤位置多項式；

(D)當該錯誤位置多項式的階數大於一門檻，藉由該徵兆計算器更新該 $2t$ 個徵兆值使相關於另一組候選值，其中該門檻 $<t$ ；

(E)重複步驟(C)至(D)，直到該錯誤位置多項式的階數不大於該門檻，則藉由一錯誤更正器基於最後調整出之錯誤位置多項式的根值來對該接收字碼進行錯誤更正。

2. 依據申請專利範圍第 1 項所述之循環碼解碼方法，其中，步驟(D)中，會藉由一徵兆更新器使用該另一組候選值來產生 $2t$ 個分別對應該等徵兆值的調整因子，以做為該徵兆計算器的更新基礎。

3. 依據申請專利範圍第 2 項所述之循環碼解碼方法，其中，該接收字碼的 n 個符元 $r_0, r_1, r_2, \dots, r_{(n-1)}$ 對應一接收多項式 $P(x) = r_0 + r_1 \cdot x + r_2 \cdot x^2 + r_3 \cdot x^3 + \dots + r_{(n-1)} \cdot x^{(n-1)}$ ，且該 $2t$ 個徵兆值分別

對應 $2t$ 個非零元素；

當其中一非零元素為 α 且符元 r_i 、 r_j 、 r_k 不可靠，該徵兆更新器為該非零元素所產生的調整因子為 $\{s_2(i) \cdot \alpha^i + s_2(j) \cdot \alpha^j + s_2(k) \cdot \alpha^k\} - \{s_1(i) \cdot \alpha^i + s_1(j) \cdot \alpha^j + s_1(k) \cdot \alpha^k\}$ ；

其中 i 、 j 、 k 為介於 $0 \sim (n-1)$ 的整數， $\{s_1(i), s_1(j), s_1(k)\}$ 是前一次選用的候選值， $\{s_2(i), s_2(j), s_2(k)\}$ 是目前選用的候選值。

4. 依據申請專利範圍第 1 項所述之循環碼解碼方法，其中，每次執行步驟(D)該徵兆計算器會選用特定的候選值，且選用順序是基於二進位碼(Binary Code)或是格雷碼(Gray Code)。
5. 依據申請專利範圍第 1 項所述之循環碼解碼方法，更包含於步驟(A)之前的一步驟：

藉由該可靠度處理器接收一個具有 n 個元素的接收序列，而評估得到具有 n 個符元的該接收字碼，其中每一元素所具有的 m 個信號為類比形式，且每一符元所具有的 m 個位元為數位形式， $m \geq 1$ ；

該可靠度處理器使用對數似然比(log-likelihood ratio, LLR)來決定每一類比形式元素的信號較可能被判斷成為 1 值的位元或較可能為 0 值的位元；且

當其中一信號為 1 值與為 0 值的機率相當，該可靠度處理器判斷對應位元不可靠。

6. 一種循環碼解碼器，包含：

一可靠度處理器，用以處理一個具有 n 個符元的接

收字碼，並為其中 q 個不可靠符元設定 $R=2^q$ 組候選值， $n>q>1$ ；

一徵兆計算器，為該接收字碼計算出相關於其中一組候選值的 $2t$ 個徵兆值，其中 t 為該接收字碼所對應之一循環碼編碼字碼的錯誤更正能力；

一關鍵方程式運算器，根據該等徵兆值產生一個階數最多為 t 的錯誤位置多項式；及

一錯誤更正器，用以對該接收字碼進行錯誤更正；

當該關鍵方程式運算器判斷出該錯誤位置多項式的階數大於一門檻，該徵兆計算器會更新該 $2t$ 個徵兆值使相關於另一組候選值，且該關鍵方程式運算器會以更新後的該等徵兆值調整該錯誤位置多項式，其中該門檻 $<t$ ；

直到調整出的錯誤位置多項式階數不大於該門檻，該錯誤更正器會基於最後調整出之錯誤位置多項式的根值來對該接收字碼進行錯誤更正。

7. 依據申請專利範圍第 6 項所述之循環碼解碼器，更包含：

一徵兆更新器，使用該另一組候選值來產生 $2t$ 個分別對應該等徵兆值的調整因子，以做為該徵兆計算器的更新基礎。

8. 依據申請專利範圍第 7 項所述之循環碼解碼器，其中，該接收字碼的 n 個符元 $r_0, r_1, r_2, \dots, r_{(n-1)}$ 對應一接收多項式 $P(x) = r_0 + r_1 \cdot x + r_2 \cdot x^2 + r_3 \cdot x^3 + \dots + r_{(n-1)} \cdot x^{(n-1)}$ ，且該 $2t$ 個徵兆值分別

對應 $2t$ 個非零元素；

當其中一非零元素為 α 且符元 r_i 、 r_j 、 r_k 不可靠，該徵兆更新器為該非零元素所產生的調整因子為 $\{s_2(i) \cdot \alpha^i + s_2(j) \cdot \alpha^j + s_2(k) \cdot \alpha^k\} - \{s_1(i) \cdot \alpha^i + s_1(j) \cdot \alpha^j + s_1(k) \cdot \alpha^k\}$ ；

其中 i 、 j 、 k 為介於 $0 \sim (n-1)$ 的整數， $\{s_1(i), s_1(j), s_1(k)\}$ 是前一次選用的候選值， $\{s_2(i), s_2(j), s_2(k)\}$ 是目前選用的候選值。

9. 依據申請專利範圍第 6 項所述之循環碼解碼器，其中，該徵兆計算器選用該等候選值的順序是基於二進位碼 (Binary Code) 或是格雷碼 (Gray Code)。

10. 依據申請專利範圍第 6 項所述之循環碼解碼器，其中，

該可靠度處理器接收一個具有 n 個元素的接收序列，而評估得到具有 n 個符元的該接收字碼，其中每一元素所具有的 m 個信號為類比形式，且每一符元所具有的 m 個位元為數位形式， $m \geq 1$ ；

該可靠度處理器使用對數似然比 (log-likelihood ratio, LLR) 來決定每一類比形式元素的信號較可能被判斷成為 1 值的位元或較可能為 0 值的位元；且

當其中一信號為 1 值與為 0 值的機率相當，該可靠度處理器判斷對應位元不可靠。

八、圖式：

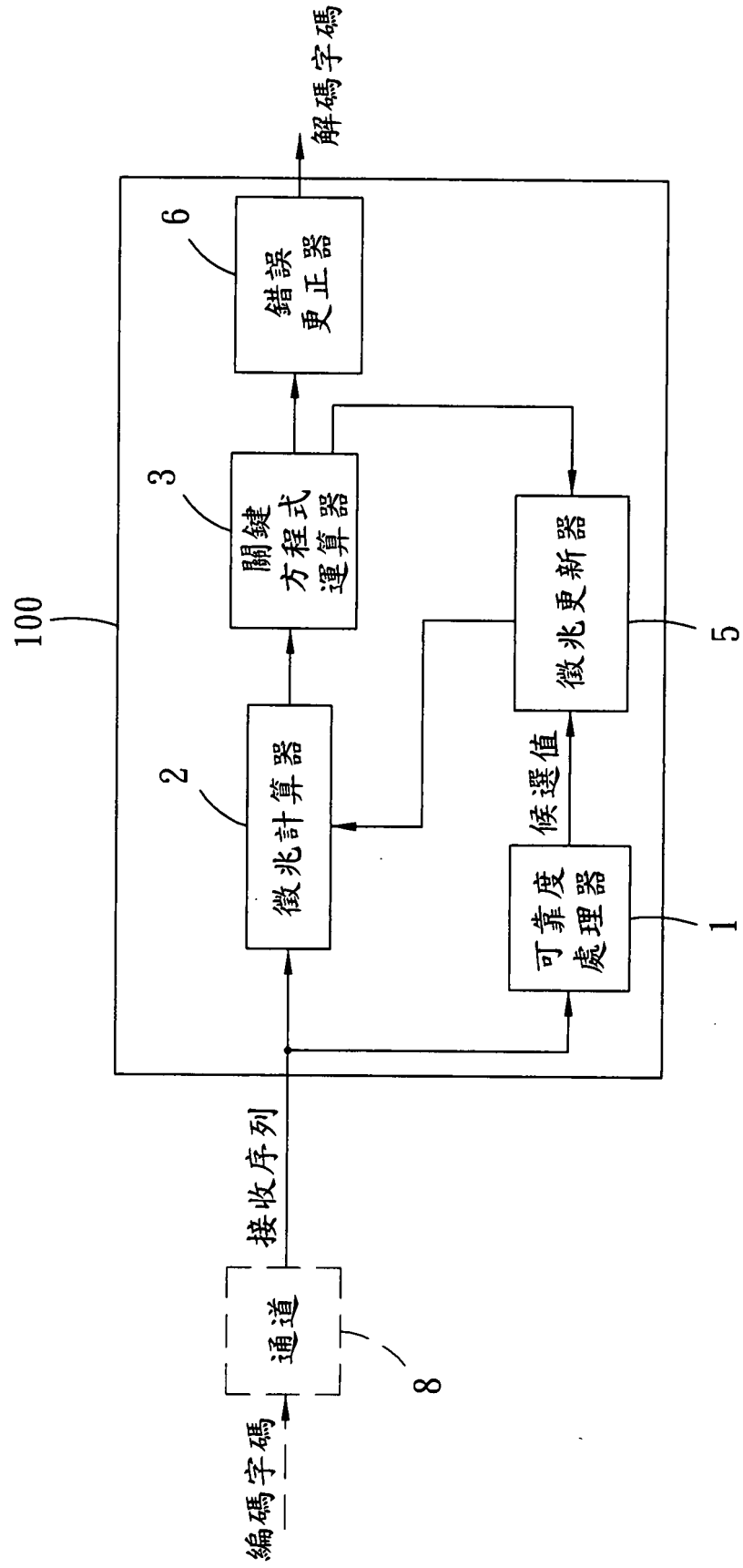


圖1

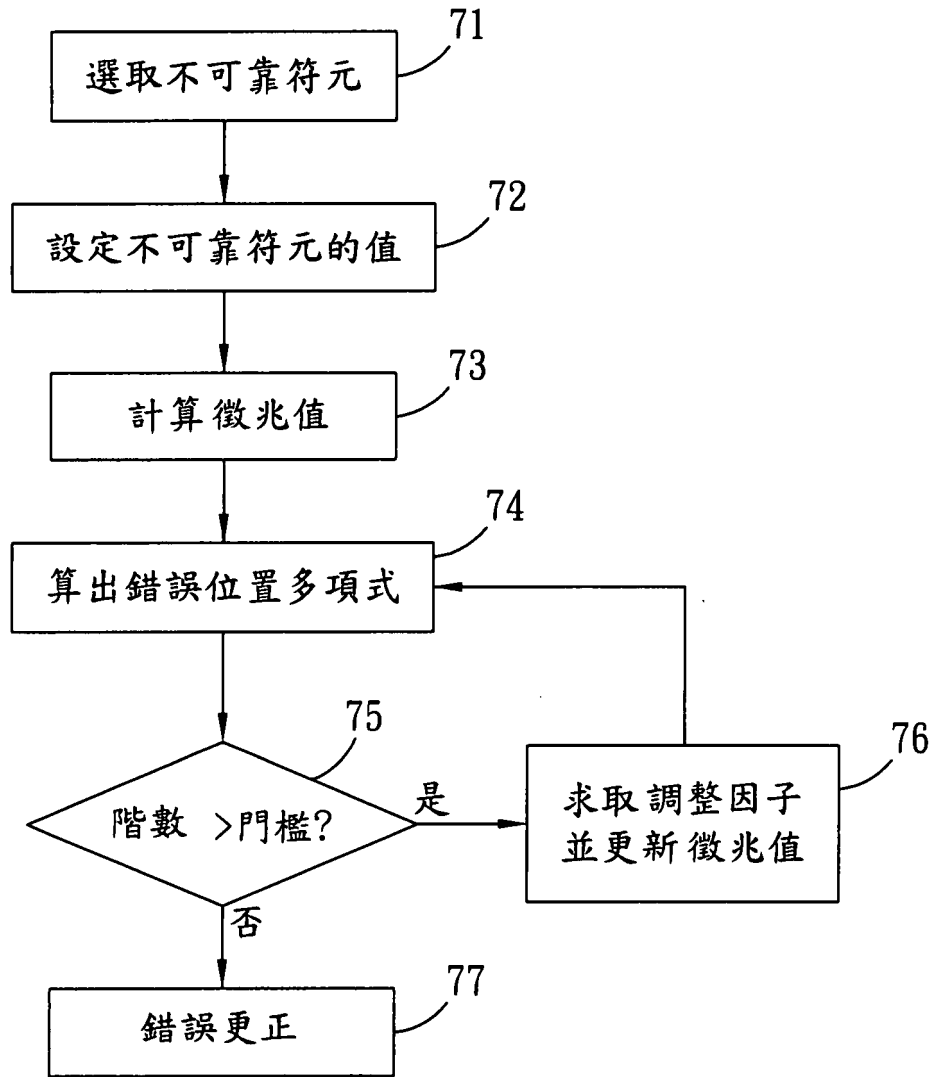


圖2

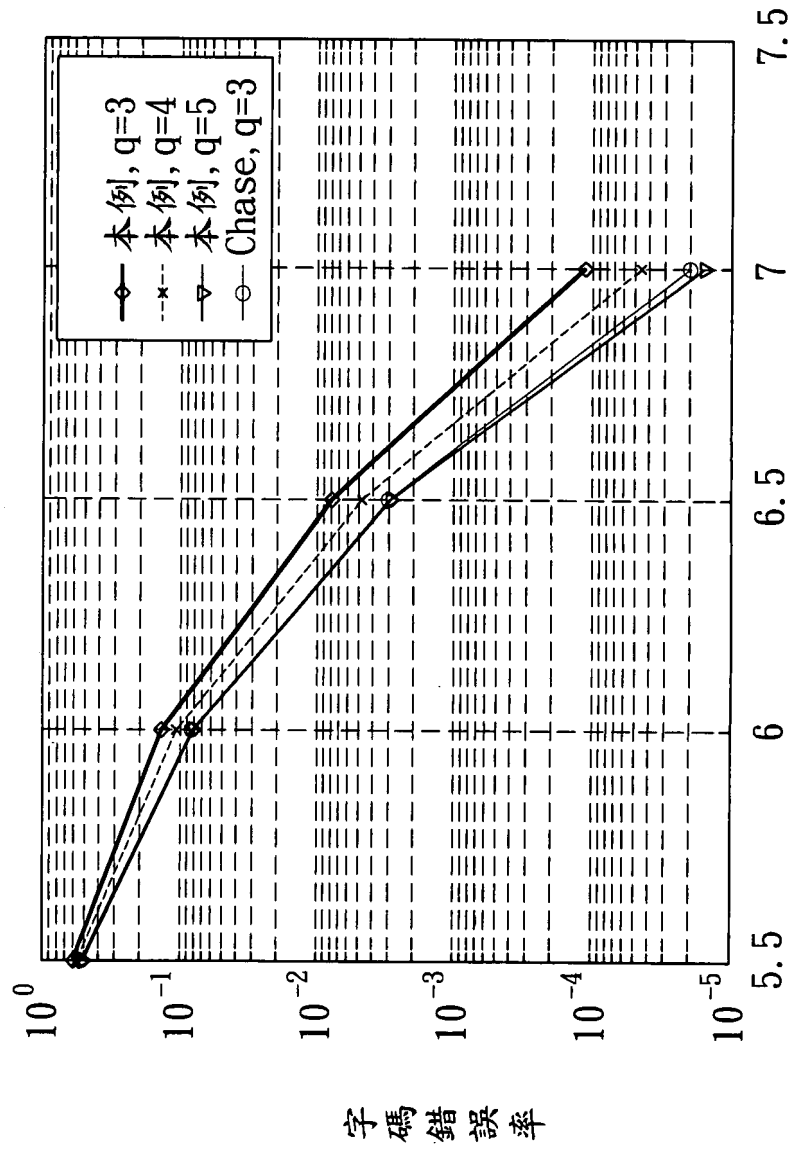


圖3

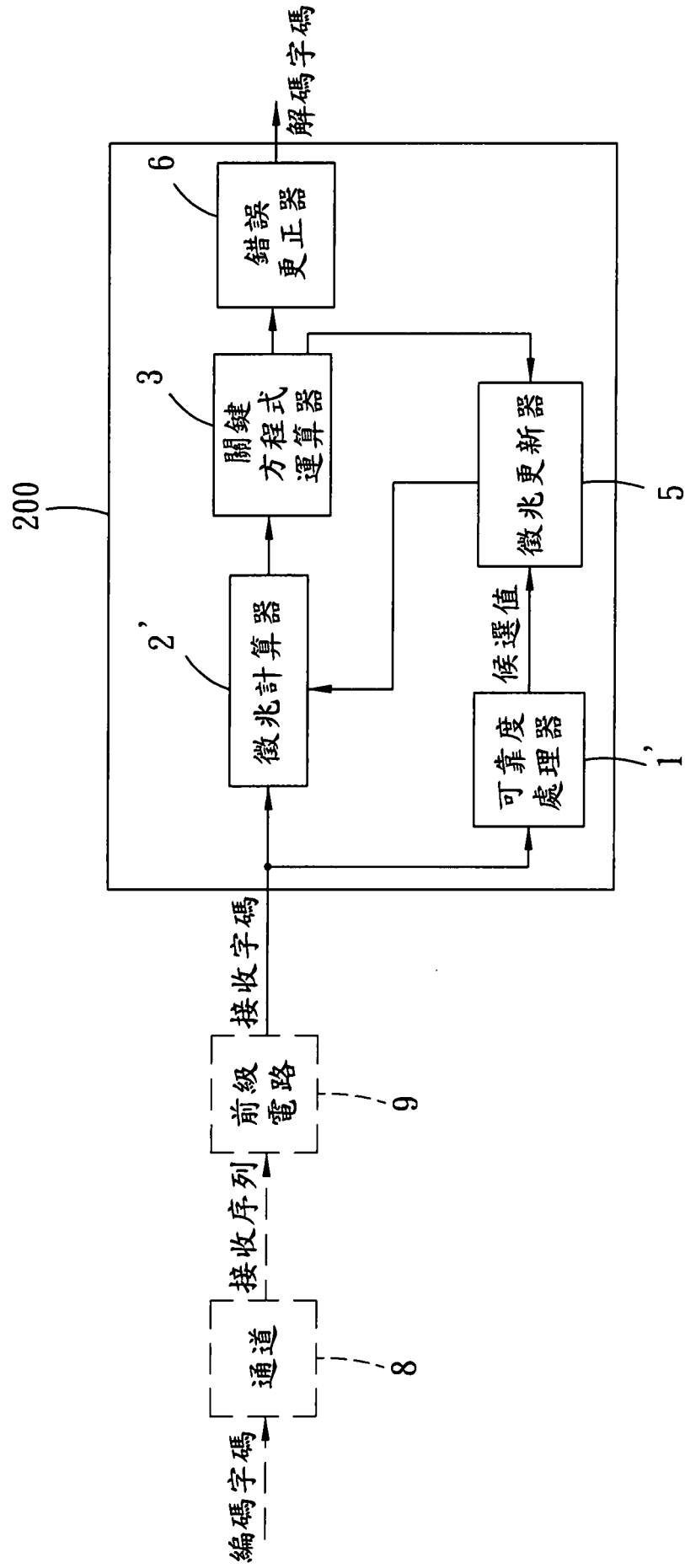


圖4