

用嵌入式系統實現歐基里德演算法

黃偉凱、胡大湘

E-mail: 9607573@mail.dyu.edu.tw

摘要

資訊傳輸時基於資訊安全與機密性考量，常以資料編碼作為基本防護，而雜訊干擾以及傳輸符元之遺失都是傳輸過程中難以避免的，對於錯誤符元及消失符元之更正編碼法則的首要目標。本論文係以簡化程序之歐基里德演算法則為基礎於Reed-Solomon碼中對錯誤及消失符元做改正，再利用嵌入式系統將歐基里德演算法具體實現。而驗證上，係將設計完成之歐基里德演算法VHDL程式下載至嵌入式發展系統中，再以Borland C++程式做一人機介面，其主要作為將輸入影像之數值擷取，待編碼後，以加入隨機亂數值之方式模擬當資料傳輸時受雜訊干擾及符元消失時之影響，而後再將受影響之數值透過RS 232傳送至嵌入式系統中，待嵌入式系統中之歐基里德演算法解碼程式將輸入之數值進行錯誤及消失符元之更正與解碼，最後將正確影像數值傳回電腦存取，透過BCB介面將影像顯示出，並且與原始影像做比對，檢視其更正結果，由實驗驗證出本系統可以完整的對於傳輸時的干擾做修正，有效解決傳輸過程中所受之影響。

關鍵詞：RS碼；歐基里德演算法；改錯碼；嵌入式系統

目錄

封面內頁 簽名頁 授權書	iii	中文摘要	iii
. . . iv 英文摘要	v	誌謝	vi
.	vii	圖目錄	x
.	x	表目錄	x
.	xii	第一章 緒論 1.1 前言	1
.	2	1.2 研究動機	1
.	2	1.3 全文架構	3
.	3	第二章 Reed-Solomon碼原理	
2.1 有限場	4	2.1.1 群與場	4
.	4	2.1.2 有限場的基本運算	7
.	7	2.2 里德所羅門碼(Reed-Solomon)定義介紹	7
.	9	2.3 里德所羅門碼(Reed-Solomon)編碼演算法則	9
.	9	2.3.1 生成多項式	9
.	9	2.3.2 Parity Check Message	10
.	12	2.4 里德所羅門碼(Reed-Solomon)解碼演算法則	12
.	14	2.4.1 特徵值(Syndrome)	12
.	14	2.4.2 解碼演算法則	15
.	19	2.4.2.1 Peterson-Gorenstein-Zierler演算法	15
.	23	2.4.2.2 Berlekamp-Massey演算法	19
.	23	2.4.2.3 Euclidean演算法	23
.	29	第三章 簡化程序之歐基里德演算法 3.1 前言	29
.	29	3.2 文獻探討	29
.	30	3.3 解碼過程比較與分析	30
.	30	3.3.1 解碼過程比較	30
.	30	3.3.2 簡化程序演算過程分析	32
.	32	3.4 解碼演算範例	37
.	37	第四章 硬體實現與系統實驗設計 4.1	
Reed-Solomon碼嵌入式控制實現	43	4.2 Reed-Solomon編解碼硬體電路實現與模擬	46
.	43	4.2.1	46
Reed-Solomon編碼電路硬體架構	47	4.2.2 Reed-Solomon解碼電路硬體架構	49
.	47	4.2.2.1 Syndrome電路硬體架構	50
.	50	4.2.2.2 ECU(Euclidean)電路硬體架構	52
.	50	4.2.2.3 Chien Search電路硬體架構	56
.	56	4.2.2.4 Reed-Solomon解碼電路集合模擬	58
.	56	4.3 實驗流程設計與驗證	61
.	61	4.3.1 Borland C++	62
.	61	4.3.2 驗證結果	64
.	61	第五章	
結論及未來展望	69	參考文獻	71

參考文獻

- [1] Y. SUGIYAMA, M. KASAHARA, S. HIRASAWA, and T. NAMEKAWA, " A method for solving key equation for decoding Goppa codes ", Inf and Contr, 1975, 27, pp.87-99
- [2] G.D. FORNEY, " On decoding BCH codes ", IEEE Trans On Information Theory, 1965, IT-11, pp.549-557
- [3] R.E. BLAHUT, " Theory and practice of error control codes ", Addison- Wesley Publishing Co, 1984, pp.258
- [4] I.S. REED, T.K. TRUONG, and R.L. MILLER, " Decoding of B.C.H. and RS codes with errors and erasures using continued fractions ", Electron. Lett, 1979, 15, (17), pp.542-544
- [5] E.P. BERLEKAMP, " Algebraic coding theory ", McGraw-Hill, 1968
- [6] E.P. BERLEKAMP, " Bit-serial Reed-Solomon encoders ", IEEE Transactions Information Theory, 1982, IT-28, (6), pp.869-874
- [7] I.S. HSU, I.S. REED, T.K. TRUONG, and W.L. EASTMAN, " Simplified procedure for correcting both errors and erasures of Reed-Solomon code using Euclidean algorithm ", Iet Jnl, pp.318-324
- [8] S. Lin and D.J. Costello, Jr, " Error Control Coding ", 2nd edition Prentice Hall, 2004
- [9] 謝坤宏, " Design and Implementation of Small-Area Reed-Solomon Decoder ", 交通大學 電信工程學系碩士班碩士論文, 2003
- [10] 謝瑋霖, " Reed-Solomon Decoder Hardware

- Implementation for Digital Video Broadcasting Standard for Terrestrial Transmission (DVB-T) Channel Coding”, 中央大學 通訊工程研究所碩士論文, 2005 [11] G.L. Feng and K.K. Tzeng, “ A generalized Euclidean algorithm for multisequence shift-register synthesis ”, IEEE Transactions on Information Theory, Vol. 35, Issue 3, pp.584-594, May 1989.
- [12] M. Srinivasan and D.V. Sarwate, “ Malfunction in the Peterson-Gorenstein-Zierler decoder ”, IEEE Transactions on Information Theory, Vol. 40, Issue 5, pp.1649-1653, Sept. 1994.
- [13] D.J. Jr. Costello, J. Hagenauer, H. Imai and S.B. Wicker, “ Applications of error-control coding ”, IEEE Transactions on Information Theory, Vol. 44, No. 6, pp.2531-2560, Oct. 1998.
- [14] 張志璋, “ Reed-Solomon Decoder for Digital Communication Systems ”, 中正大學 電機工程研究所碩士論文, 2003 [15] I. S. Reed, M. T. Shih, and T. K. Truong, “ VLSI Design of Inverse-free Berlekamp-Massey Algorithm ”, IEEE Proceedings-E, vol. 138, no. 5, pp. 295-298, Sept. 1991.
- [16] W. Wilhelm, “ A New Scalable VLSI Architecture for Reed-Solomon Decoder ”, IEEE Journal of Solid-State Circuits, vol. 34, pp. 388-396, Mar. 1999.
- [17] R. P. Brent and H. T. Kung, “ Systolic VLSI Arrays for Polynomial GCD Computation ”, IEEE Transactions on Computers, vol. c-33, no. 8, pp. 731-736, August 1984.
- [18] 劉紹漢、林灶生、劉新民, VHDL晶片設計, 全華科技圖書, 台北, 第7-33頁, 2004。