International Engineering Journal For Research & Development

E-ISSN No:2349-0721

Volume 2: Issue 2

REVIEW ON DIFFERENT ERROR CORRECTION CODES

Ms.R.S.Khond[1], Prof.P.R.Indurkar[2], Prof.P.R Lakhe[3]

[1]Department of Electronics Engineering S.D. College Of Engg, [2]Department of ENTC B.D. College Of Engg.,[3]Department of Electronics Engineering S.D. College of Engineering Wardha, Maharashtra, India

[1]r.khond21@gmail.com

ABSTRACT—

Data transfer between two or multiple nodes is a common but critical operation in many applications of communication network. When the data transfer through the communication network some error bit and noise is added with data. To obtain the original data at the receiver side Error Correction Codes are used. Communication performance is improved by enabling the transmitted signal to better withstand the effects of channel disabilities such as noise, interference and fading which occur during transmission as well as during comparison. In short the Error Correction is the detection of errors and reconstruction of the original, error free data. There are errors correcting codes to detect as well as correct errors. In this paper the detail review of error correction code techniques is done. The main aim of this paper is to study the different error correcting codes techniques available for the error correction during the transformation of the data from the communication channel in details.

Keywords—Error Correction Code, data comparison, Communication channel.

INTRODUCTION :

In the communication system there are errors during the transmission of the information. Communication performance is improved by enabling the transmitted signal to better withstand the effects of channel disabilities such as noise, interference and fading which occur during transmission as well as during comparison. In short the Error Correction is the detection of errors and reconstruction of the original, error free data. There are errors correcting codes to detect as well as correct errors. These error correcting codes have vast field of application.

In a computing system, it is necessary to compare the received data with the stored by the method of data comparison. During the comparison if it is observed that the received data cannot match with the stored data when we go through the data comparison process that the mismatched data is called as an error. So for the proper and quality communication it is necessary to reduce those errors. Hence by using data comparison technique the errors can be detected and corrected. Data comparison is widely used in computing systems to perform many operations such as the tag matching in a cache memory and the virtual-to-physical address translation in a translation look a side buffer (TLB). The circuit, therefore, must be designed to have as low latency as possible, or the components will be disqualified from serving as accelerators and the overall performance of the whole system would be severely deteriorated. As recent computers employ error-correcting codes (ECCs) to protect data and improve reliability [1]–[5], complicated decoding procedure, which must precede the data comparison, elongates the critical path and exacerbates the complexity overhead. Thus, it becomes much harder to meet the above design constraints. Despite the need for sophisticated designs as described, the works that cope with the problem are not widely known in the

literature since it has been usually treated within industries for their products. Recently, however, [6] triggered the attraction of more and more attentions from the academic field. The most recent solution for the matching problem is the direct compare method [6], which encodes the incoming data and then compares it with the retrieved data that has been encoded as well. Therefore, the method eliminates the complex decoding from the critical path. In performing the comparison, the method does not examine whether the retrieved data is exactly the same as the incoming data. Instead, it checks if the retrieved data resides in the error correctable range of the codeword corresponding to the incoming data. As the checking necessitates an additional circuit to compute the Hamming distance, i.e., the number of different bits between the two code words, the saturate adder (SA) was presented in [6] as a basic building block for calculating the Hamming distance.

However, [6] did not consider an important fact that may improve the effectiveness further, a practical ECC codeword is usually represented in a systematic form in which the data and parity parts are completely separated from each other [7]. In addition, as the SA always forces its output not to be greater than the number of detectable errors by more than one, it contributes to the increase of the entire circuit complexity.

DATA COMPARISON

Data comparison is a logic that has many applications and it is widely used in computing system. For example tag matching in a cache memory means to check whether a piece of information is in cache, the address of the information in the memory is compared to all cache, the tags in the same set that might contain that address. Another place that uses data comparison logic is the virtual-to-physical address translation in a translation look aside buffer (TLB). As recent computers employ error-correcting codes (ECCs) to protect data and improve reliability [8]–[12], complicated decoding procedure, which must precede the data comparison, elongates the critical path and increases the complexity overhead. Thus, it becomes much harder to meet the above design constraints. Data comparison circuit is usually in the critical path of a pipeline stage because the result of the comparison determines the flow of the succeeding operations. When the memory array is protected by ECC, it exacerbates the criticality because of the added latency due to ECC logic. In proposed work, we consider the characteristics of systematic codes in designing the proposed architecture and propose a low-complexity processing element that computes the Hamming distance faster. Therefore, the latency and the hardware complexity are decreased considerably even compared with architecture.

DIFFERENT ERROR CORRECTION CODE TECHNIQUE

A. Double Error Correcting Long Code

Double Error Correction Long Code technique is a novel binary, long double error correcting, systematic code (8 2 5) that can detect and correct errors up to two bits in the received vector using simple concept of syndrome decoding. In an error correcting code the central concept is the notion of minimum distance. If a code can be constructed with the minimum distance of 2t+1 between two code words, then any number of errors per codeword which does not exceed t can be corrected. A linear block code C is generally specified as (n, k, d) code, where n= length of the code word, k= length of information bits and d is the minimum Hamming distance between any two

code words [13][14]. Shannon showed that at any rate of information transmission up to the channel capacity, it should be possible to transfer information at error rates that can be reduced to any desired level. In [15] author shows that the practical codes such as, long turbo codes with code rates <½ can approach the theoretical limits on code (near Shannon limit) performance [16].

B. Polar Codes

Polar codes [17] are a significant breakthrough in coding theory, since it is proved that polar codes can achieve the channel capacity of binary-input symmetric memory-less channels in [17] and any discrete or continuous channel in [18]. Polar codes can be efficiently decoded by the low-complexity successive cancelation (SC) decoding algorithm [17] with complexity of O(N logN), where N is the block length.

The generation matrix of a polar code is an N×N matrix $G = B_N F^{\otimes n}$, where N $=2^n$, BN is the bit reversal permutation matrix, and $F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Here $\otimes n$ denotes the nth Kronecker power and $F^{\otimes n} = F \otimes F^{\otimes(n-1)}$. Let $u_0^{N-1} = (u_0, u_1, \dots, u_{N-1})$denote the data bit sequence and $x_0^{N-1} = (x_0, x_1, \dots, x_{N-1})$the corresponding encoded bit sequence, then $x_0^{N-1} = u_0^{N-1} G$. The indices of the encoding bit sequence u_0^{N-1} are divided into two sets: the information bits set A contains K indices and the frozen bits set A^ccontains N − K indices. uA are the information bits whose indices all come from A. uAc are the frozen bits whose indices from A^c. The encoding graph of a polar code with N = 8 is shown in Fig. 1.A polar code of length N = 2n can also be represented by a full binary tree of depth n [19], where each node of the tree is associated with a constituent code. The binary tree representation of an (8, 3) polar code is shown in Fig. 2, where the black and white leaf nodes correspond to information and frozen bits, respectively. In order to show the connection between the tree representation and the direct encoding graph in Fig. 1, the constituent code associated with

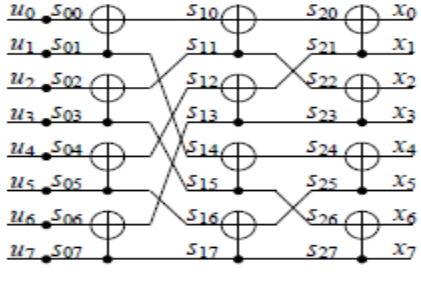

Fig. 1. Polar encoder with N = 8

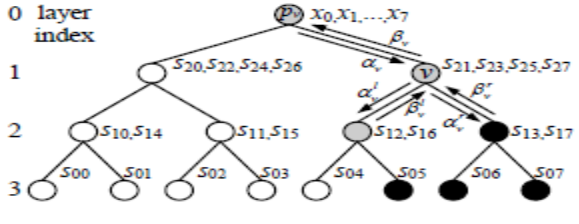

Fig. 2. Binary tree representation of a (8, 3) polar code

each tree node is also shown in Fig. 2. There are three types of nodes in a binary tree representation of a polar code: rate-0 , rate-1 and arbitrary rate nodes. The leaf nodes of a rate-0 and rate-1 nodes are associated with only frozen and information bits, respectively. The leaf nodes of an arbitrary rate node are associated with both information and frozen bits. For example, the rate-0, rate- 1 and arbitrary rate nodes in Fig. 2 are represented by circles in white, black and gray, respectively.

C. BCH Codes

BCH codes are t-error-correcting codes with symbol field from GF(q) and error location field from GF(q^m). In general, m_1 for BCH codes, if m = 1, we give it another name --Reed-Solomon codes. Let p and m be given and β be any element of GF(q^m) of order n (i.e. β^n. = 1). Then for any positive t and any integer j, the corresponding BCH code is the cyclic code of block length n with the generator polynomial [20] [2].

$$g(x) = \text{LCM}\ [f_{j_0}(x), f_{j_0+1}(x), \quad , f_{j_0+2t-1}(x)]$$

where fj(x) is the minimal polynomial of βj. The above equation is a so-called generalized BCH generator polynomial. Usually j0 = 1 and β is the primitive element of GF(q^m) α. Therefore, g(x) will have $\alpha^1, \alpha^2, \alpha^3$, ..., α^{2t} as its roots. If the symbol field and the error-locator field are different, i.e., the error locator field is an extension field of the symbol field, then g(x) will have more than 2t degree. The message length is k < n-2t. In the case of Reed-Solomon codes, the message word length is exactly n-2t. We say this is more efficient because the code rate k/n is larger for the same n.

D. Reed-Solomon Codes

Reed-solomon codes are random single- or multiple-symbol error correcting codes. Operating on symbols which are elements of a finite field. The coefficients of the data polynomial and the check symbols are elements of the field, and all encoding, decoding , and correction computations are performed in the field. Reed-Solomon are symbol oriented and the circuits implementing them are typically colocked once per data symbol, although bit-serial techniques are also employed. Reed-solomon codes are used as part of the forward-error-correction strategy of digital transmission of signals in the four proposed all digital HDTV systems. The following table summarizes the length of a code block in bytes1, the length of parity-check bytes, and the error-correcting capability of these digital HDTV systems [20] [4].

Table 1: Reed-Solomon codes used in HDTV systems.

HDTV Systems	$Code\ Structure^2$	Check Bytes	T^3
ADTV	(147, 127)	20	10
CCDC	(158, 148) (32 QAM) (116, 106) (16 QAM)	10 10	5 5

Digi-Cipher	(155, 145) (32 QAM) (116, 106) (16 QAM)	10 10	5 5
DSC	(167, 147)	20	10

The length of the parity-check symbols are twice the error-correcting capability. This is a unique property of Reed-Solomon codes and is a result of having a generator polynomial of degree 2t.

Table 2: Comparison of different ECC Encoder Results

Technique	Data Width	Latency (ns)	Area (m^2)
Ham. SEC	16	0.4	296
	32	0.5	598
	64	0.65	1302
Hsiao SEC-DED64	16	o.4	291
	32	0.5	605
	64	0.7	1168
DEC	16	0.5	496
	32	0.6	1250
	64	0.7	2335
DEC-TED	16	0.9	786
	32	1.1	1424
	64	1.3	2546

Table 3: Comparison of different ECC Decoder Results

Technique	Data Width	Latency (ns)	Area (m^2)
Ham. SEC	16	0.4	296
	32	0.5	598
	64	0.65	1302
Hsiao SEC-DED64	16	0.4	291
	32	0.5	605
	64	0.7	1168
DEC	16	0.5	496
	32	0.6	1250
	64	0.7	2335
DEC-TED	16	0.9	786
	32	1.1	1424
	64	1.3	2546

Table 1 shows the comparison of different Error Correcting Code encoder result with respect to latency and Area. Table 2 shows the comparison of different Error Correcting Code decoder result with respect to latency and Area. Form the table is it is clear that Ham. SEC uses BCH code ECC technique and has the minimum latency for the different data size of data transfer through the communication network.

CONCLUSION :

In this paper the detail review of error correction code techniques is done. The main aim of this paper is to study the different error correcting codes techniques available for the error correction during the transformation of the data from the communication channel in details.

In future work pipelining based Error Correction Code is used for the fast and high error correction capability for the communication system.

REFERENCES :

[1] J. Chang, M. Huang, J. Shoemaker, J. Benoit, S.-L. Chen, W. Chen, S. Chiu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, "The 65-nm 16-MB shared on-die L3 cache for the dual-core Intel xeon processor 7100 series," IEEE J. Solid-State Circuits, vol. 42, no. 4, pp. 846–852, Apr. 2007.

[2] J. D. Warnock, Y.-H. Chan, S. M. Carey, H. Wen, P. J. Meaney, G. Gerwig, H. H. Smith, Y. H. Chan, J. Davis, P. Bunce, A. Pelella, D. Rodko, P. Patel, T. Strach, D. Malone, F. Malgioglio, J. Neves, D. L. Rude, and W. V. Huott "Circuit and physical design implementation of the microprocessor chip for the zEnterprise system," IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 151–163, Jan. 2012.

[3] H. Ando, Y. Yoshida, A. Inoue, I. Sugiyama, T. Asakawa, K. Morita, T. Muta, and T. Motokurumada, S. Okada, H. Yamashita, and Y. Satsukawa, "A 1.3 GHz fifth generation SPARC64 microprocessor," in IEEE ISSCC. Dig. Tech. Papers, Feb. 2003, pp. 246–247.

[4] M. Tremblay and S. Chaudhry, "A third-generation 65nm 16-core 32-thread plus 32-scout-thread CMT SPARC processor," in ISSCC. Dig. Tech. Papers, Feb. 2008, pp. 82–83.

[5] AMD Inc. (2010). Family 10h AMD Opteron Processor Product Data Sheet, Sunnyvale, CA, USA [Online]. Available:http://support.amd.com/us/Processor_TechDocs/40036.pdf

[6] W. Wu, D. Somasekhar, and S.-L. Lu, "Direct compare of
information coded with error-correcting codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 11, pp. 2147–2151, Nov. 2012.

[7] S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.

[8] Byeong Yong Kong, Jihyuck Jo, Hyewon Jeong, Mina Hwang, Soyoung Cha, Bongjin Kim, and In-Cheol Park "Low-Complexity Low-Latency Architecture for Matching of Data Encoded With Hard Systematic Error-Correcting Codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, No. 7, July 2014.

[9] Wei Wu, Dinesh Somasekhar, and Shih-Lien Lu "Direct Compare of Information Coded With Error-Correcting Codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, No. 11, Nov 2012.

[10] Bo Yuan, Keshab K. Parhi "Low-Latency Successive-Cancellation Polar Decoder Architectures Using 2-Bit Decoding," IEEE Trans. On Ckt & Systems-I: Regular Papers, vol. 61, No. 4, APRIL 2014.

[11] Shashank V. Maiya, Daniel J. Costello, Jr., Thomas E. Fuja "Low Latency Coding: Convolution Codes vs. LDPC Codes," IEEE Trans. On Communication, vol. 60, No. 5, MAY 2012

[12] Wei Yu, Masoud Ardakani, Benjamin Smith, Frank Kschischang "Copmlexity-Optimized Low-Density Parity-Check Codes for Gallager Decoding Algorithm B."

[13] S. Lin and D.J. Costello, (1983), "Error Control Coding: fundamentals and applications, Prentice Hall Publishers.

[14] S B Wicker, (1994), Error Control Systems for Digital Communication and Storage, Prentice Hall Publishers.

[15] Christian Schlegel and Lance Per´ez, (1999), "On Error Bounds and Turbo-Codes", IEEE Communications Letters, Vol. 3, No. 7, pp 205-207.

[16] George Clark and J Cain, "Error correcting code for digital communications" John Willey Publishers.

[17] E. Arıkan, "Channel polariztion: a method for constructing capacityachieving codes for symmetric binary-input memoryless channels," IEEE Trans. Info. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[18] E. Sasoglu, E. Teltar and E. Arıkan, "Polariztion for arbitrary discrete memoryless channels," in Proc. IEEE Int. Symp. on Information Theory,2009, pp. 144–148.

[19] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successivecancellation decoder for polar codes," IEEE Commun. Lett., vol. 15, no. 12, pp. 1378–1380, Dec. 2011.

[20] Neal Glover and Trent Dudley, "Practical Error Correction Design for Engineers" 2nd ed., DST, 1988.

[21] Riaz Naseer and Jeff Draper," Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs" at 34th European Solid-State Circuits Conference, 2008. ESSCIRC 2008, Page(s): 222 – 225, 15-19 Sept. 2008